Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Consistent Mode-Finding for Parametric and Non-Parametric Clustering

A thesis submitted to Trinity College Dublin, the University of Dublin in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Joshua Tobin

Supervised by
Prof. Mimi Zhang

Discipline of Statistics – School of Computer Science & Statistics
Trinity College Dublin, the University of Dublin.

2023

Declaration

I declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work.

I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Joshua Tobin

October 22, 2022

Abstract

Density peaks clustering detects modes as points with high density and large distance to points of higher density. To cluster the observed samples, points are assigned to the same cluster as their nearest neighbor of higher density. This efficient and intuitive approach has, in recent years, grown in popularity in applications. Despite its widespread use, little work has been completed aiming at understanding the theoretical properties of the density peaks method, as well as its strengths and limitations when clustering. Here, we provide a detailed analysis of the density peaks clustering algorithm. We demonstrate that it recovers consistent estimates of the modes of the underlying density and correctly clusters the data with high probability. However, deficiencies of the density peaks clustering methodology are also highlighted. Noise in the density estimates can lead to errors when estimating modes and incoherent cluster assignments. Two adaptations of the density peaks clustering approach are proposed to remedy these issues. The first method seeks to detect modal sets rather than point modes in the data. This reduces the sensitivity of the clusterings to fluctuations in the density estimate. The second approach partitions the data into regions mutually separated by areas of low density, before applying the density peaks clustering algorithm. Doing so ensures that the result of the cluster assignment method meets the conceptual understanding of a correct clustering. Both approaches are analyzed theoretically and their superior performance is demonstrated on simulated and real-world datasets. Moreover, they are shown to be suitable for modern clustering applications in computer vision. Model-based clustering methods, where clusters are taken to be unimodal components in a finite mixture model, are then considered. Motivated by the

consistent estimates of the modes provided by the density peaks clustering algorithm, a novel model-based clustering method is proposed. This approach uses a set of high density points as initial mean parameters, and iteratively prunes them to return a sequence of nested clusterings. The method outperforms popular model-based clustering methods. To conclude, the contributions of the thesis are used to motivate suggestions for future research.

Acknowledgements

First of all, I must acknowledge the guidance, support and patience of my supervisor Mimi Zhang. Mimi combines the roles of mentor, teacher, advisor, and friend. Her dedication to her work is incredible and her door is always open, ready to offer guidance and help in any way she can. I could not have asked for a better supervisor.

In addition I would like to thank Clint Ho for his contributions to our collaborative work, for sharing his deep knowledge of optimization methods, and his grace and cheer in the face of setbacks.

I am particularly thankful for the statistics community in Trinity; for the passion and drive of the other students in our group, that constantly energized and inspired. Particular thanks to Daniel Dempsey and Michael Ferreira, for their support and for their good humour, without which the office would not have been the same. I am grateful to Arthur White, for all of his work organising weekly seminars, to Simon Wilson, for giving me the opportunity for the trip of a lifetime, to Myra O'Regan, Michael Stuart, John McDonagh, and James Ng for giving me the foundation to teach statistics and, while doing so, discover a new passion; and to all of the statistics faculty in Trinity, for pairing their incredible expertise with generosity and goodwill.

I also gratefully acknowledge the supports provided by the broader Trinity community, the technical support provided by Michael White, Chris, our building manager in the Lloyd Institute, and Tina and Christine for brightening up many mornings.

I am sincerely thankful for my examination committee, Jason Wyse, Nicos Pavlidis and Douglas Leith. It's hard to believe, but their kindness, humour, and engagement made the viva an experience to savour and enjoy.

Lastly, thanks to those who are closest to me. I am profoundly lucky to have a family as kind and caring as I do. I would not be the person I am without them. And to Alex, for your unending support and love throughout this journey, for which I owe an infinite gratitude. I can't wait for our adventures yet to come.

Publications

The following articles, arising from this work, are published or are under review.

Tobin, J., & Zhang, M. (2021, December). DCF: An Efficient and Robust Density-Based Clustering Method. In 2021 IEEE International Conference on Data Mining (ICDM).

Tobin, J., & Zhang, M. A Theoretical Analysis of Density Peaks Clustering and the CPF Algorithm. IEEE Transactions on Pattern Analysis and Machine Intelligence (under review).

Tobin, J., Ho, C. P., & Zhang, M. Reinforced EM Algorithm for Clustering with Gaussian Mixture Models. 2023 SIAM International Conference on Data Mining (SDM) (under review).

Contents

List of Figures

List of Tables

List of Notation

$\mathcal{X}$	The support of the density.
$\boldsymbol{X}$	The observed sample.
n	The number of instances in the sample.
p	The dimension of the data.
$\boldsymbol{x}$	A data point.
f	The data generating probability density.
k	The number of neighbors used to compute density estimates.
$r_k(\boldsymbol{x})$	The distance from a point to it's k-th nearest neighbor.
$\hat{f}_(\boldsymbol{x})$	The k-NN density estimate at a point.
h	The bandwidth.
$\hat{f}_h(\boldsymbol{x})$	The kernel density estimate at a point.
$\omega(\boldsymbol{x})$	The distance from a point to its nearest neighbor of higher density.
$\gamma(\boldsymbol{x})$	The peak-finding criterion for a point.
$L(\lambda)$	The level set of the population density.
$G(\boldsymbol{V}, E)$	An undirected graph with vertex set $\boldsymbol{V}$ and edge set E.
$G(\lambda)$	The estimated level set of the graph.
$\vec{G}(\boldsymbol{V}, \vec{\boldsymbol{E}})$	A directed graph with vertex set $\boldsymbol{V}$ and edge set $\vec{E}$.

1 Introduction

Prominent methods for clustering, the task of partitioning data into homogeneous groups termed clusters, while intuitive, often lack clear statistical motivations. Early techniques recover clusters using only the distances between observations, grouping close objects together to form groups before a notion of group similarity is used to merge groups together. A second methodology represents clusters by a central point, often the mean or median, and derives the grouping of the data based on the proximity of observations to these representatives. Both approaches, while intuitive, lack a clear notion of the clusters they seek to obtain and, thus, a measure of what constitutes a successful clustering. An attempt to navigate the ill-posedness of the clustering problem relates the notion of clusters to the underlying probability distribution assumed to generate the observed data. By doing so, the clustering problem is framed as a problem of statistical estimation, firstly of the underlying density and, subsequently, of the clusters present within it. Approaches that assume this framework are henceforth referred to as density-based clustering methods.

The two prominent methodologies for density-based clustering differ fundamentally in their conception of the distribution assumed to generate the data. Parametric, or model-based clustering methods conceive the density as a mixture of simple parametric distributions, often the multivariate Gaussian distribution for continuous data and the Poisson distribution for discrete data. The parameters of the component distributions are estimated from the observed samples using maximum likelihood techniques. Non-parametric methods impose fewer (if any) assumptions on the structure of the density.

Instead, the probability distribution assumed to generate the data is estimated with kernel density functions.

Both methodologies are united in relating the notion of a cluster to "bumps" in the density resulting from a tight mass of data. These bumps are the local maxima of the probability density and are termed the modes of the density. For parametric clustering, a cluster is a unimodal component within an appropriate finite mixture model. The clustering task is thus to estimate parameters of the mixture model such that the component distributions each possess a single mode. For non-parametric density-based clustering methods, the link between the notion of a cluster and the modes of the density is even more immediate. Taking clusters to be regions of concentrated probability mass separated from each other by regions of lower probability density, the clusters are naturally associated with the modes of the probability density distribution. Each cluster is typically understood as the domain of attraction of a mode. To obtain estimates of the clusters from the observed sample, two contrasting proposals predominate in the literature. The first extracts clusters using level sets of the density. A density-level set, at a certain level, is the set of points whose density is at least the value of the level. The clusters are taken to be connected subsets of these density-level sets. The aim is to estimate the connected subsets of the level sets such that each contains exactly one mode. A second set of estimation approaches, termed mode-seeking methods, find the mode associated with an observation as the point of convergence of an ascent of the density beginning at that point.

The ability to directly detect modes of the underlying probability distribution from a finite sample is of central importance in non-parametric density-based clustering. A rich literature exists developing estimators for the modes of a density and providing theoretical guarantees on their performance. The majority of this work concerns methods that estimate only a single mode of the density, and often present estimators that are challenging to implement in practice. The difficulty in implementation is the result of seeking mode estimates over the whole support of the density, in contrast to more direct methods that estimate the mode using statistics of the observed sample.

Theoretical guarantees for the direct sample-based approaches have been developed recently in the literature.

The same distinction, between methods that cluster the data using estimates of the modes from the entire support of the density and those that use only statistics of the observed sample, also exists in the literature for non-parametric clustering. Previous research has focused on mode-seeking methods that determine clusters by relating the observation to estimated modes lying in the support of the density. The gradient ascent procedure used to relate the observations in the sample to their respective modes thus operates over the entire support of the density. Such approaches are challenging to implement, and their performance depends heavily on the appropriate selection of parameters.

The density peaks clustering algorithm of Rodriguez and Laio (2014) is a non-parametric density-based clustering method that clusters data using only statistics of the observed sample. Estimates for the modes are selected as points with high density at a relatively large distance from any points of higher density. The algorithm then uses a sample-based analogy to the gradient ascent approach of competitor methods to relate observations to their respective modes: each observation is assigned to the same cluster as its nearest neighbor of higher local density. As such, the density peaks clustering algorithm is said to be a mode-seeking method. The density peaks method provides users with an intuitive plot, from which the best estimators of the modes are detected. The density peaks method is intuitive and can detect clusters of arbitrary shape and size. For these reasons, density peaks clustering has been widely adopted in applications. Examples of its use are found in fields as diffuse as the analysis of social networks, the profiling of cancer risk factors, and multiple applications in the field of computer vision. Furthermore, the density peaks clustering method has been extended in innumerable ways, with proposals variously editing the method used to estimate the underlying probability density, the method used to select the mode estimates, and the method used to allocate observations to the clusters of their respective modes.

Despite the popularity and prominence of the density peaks clustering method, there has been, to date, no theoretical work demonstrating its ability to achieve the population goals of non-parametric density-based clustering. This involves producing consistent estimates of the modes of the generating probability distribution and correctly assigning the instances to the clusters in accordance with the gradient of the underlying density. This dearth of theoretical analysis has inhibited the understanding of the density peaks clustering method as a well-grounded non-parametric clustering algorithm. Many of the adaptations of the density peaks clustering method are, thus, poorly justified in the context of density-based clustering.

1.1 Research Aims

Building on the above discussion outlining the context of the work contained in this thesis, we next detail the four primary aims of this research.

This first aim of this thesis is to provide a theoretical grounding for the density peaks clustering method. This requires formalizing the density peaks clustering approach and subsequently analyzing the properties of both the mode estimates returned by the algorithm, and the resulting clusterings after observations have been assigned to the clusters of their associated modes. It is in Chapter 3 that this analysis is presented. Our work demonstrates that the density peaks clustering algorithm recovers consistent estimates for each mode of the underlying probability distribution. To our knowledge, these are the first such guarantees for the consistency of the mode estimates produced by the density peaks method. Moreover, we give finite sample bounds on the quality of the recovered modes. Subsequently, it is shown that the sample-based assignment method of the density peaks clustering method correctly clusters the data with high probability.

While theoretical foundations for the density peaks clustering algorithm yield useful insights, the performance of the algorithm for datasets of the size typically seen in applications is equally important. As will be shown in Chapter 3, the algorithm can

return poor quality clusterings due to failures in either the mode estimation method, the assignment strategy, or both concurrently.

As such, the second primary aim of this thesis is to develop a strategy that improves the mode estimation method of the density peaks clustering algorithm. It is demonstrated that the quality of the mode estimates is significantly degraded in the presence of noisy estimates of the density. To account for this, a robust method for modelling high density regions in the data is required. The algorithm must be robust to noise in the density estimate and thus detect clusters at varying density levels. Furthermore, the algorithm must be competent at deciding the correct number of clusters, even when the number of clusters is very high. In Chapter 4, a novel method is introduced that directs the peak-finding technique to discover modal sets, rather than point modes present in the data. We seek to demonstrate the superiority of this approach against commonly used alternatives for non-parametric clustering.

As well as developing an improved method for estimating the high density regions in the data, we seek to improve the assignment procedure of the density peaks clustering algorithm. This forms the third primary goal of this research. The density peaks clustering method uses a sample-based analogy to gradient ascent of the density to assign points to the clusters of their associated modes. It is shown that this approach, while efficient, can lead to geometrically incoherent clusterings. Furthermore, the assignment strategy can assign instances to clusters across regions of low density in the data. This contravenes the understanding of a cluster as a region of attraction of a mode. To navigate these issues, in Chapter 5, a novel approach is proposed that initially partitions the data, such that regions separated from each other by areas of low density are kept apart. The density peaks clustering method can then be applied to each subset present in the partition individually, thus restricting the assignment strategy to produce clusters as contiguous regions of high density. The performance improvements offered by these methods will be demonstrated in Chapter 6.

The final aim of this thesis is to demonstrate the applicability of the density peaks

clustering method for model-based clustering. The model-based and non-parametric formulations of the clustering task have been developed independently with few insights from one being adapted for use in the other. This is particularly unusual considering both methodologies have a shared interest in understanding modal regions of the data-generating density. We seek to demonstrate that the mode estimates produced by the density peaks clustering method can be used to provide high-quality parameter estimates for model-based clustering. In Chapter 7, such an approach is developed. A well-justified method for selecting which mode estimates to use as mean parameters is introduced. Taking these estimates as the mean parameters in the mixture, the quality of the maximum likelihood estimates of the remaining model parameters, and the resulting clusterings, are enhanced. The results demonstrate the potential for productive integration of parametric and non-parametric clustering methods in future research.

1.2 Methodology

We next consider the research methodologies adopted in order to achieve the research aims introduced in the previous section.

Methods that motivate cluster analysis by relating the clusters to features of the underlying density, in both parametric and non-parametric formulations, have been developed since the 1960s. To understand the literature on this topic, a review of the existing literature was completed. The review, found in Chapter 2, first considers the model-based formulation. The concept of a cluster of the underlying probability density is described. For model-based clustering, the population clusters are unimodal components within an appropriate finite mixture model. Particular attention is given to the mixture of multivariate Gaussian components, as it is the focus of later work in this thesis. Subsequently, prominent estimation procedures are reviewed. The Expectation Maximization (EM) algorithm (Dempster et al., 1977) used to derive maximum likelihood estimates for the parameters, as well as popular model selection criteria for

mixture modelling are introduced. The same structure is used for reviewing work on non-parametric clustering methods. Firstly, the notion of a population cluster in the non-parametric setting, as the attraction region of a mode of the underlying density, is discussed. As with the parametric approaches, the popular estimation procedures for non-parametric clustering are then reviewed. This begins with a review of prominent level set methods. Mode-seeking methods are then introduced, providing context for the development of the density peaks clustering method. The challenges associated with model selection for non-parametric clustering are also briefly discussed. The comprehensive literature review is seen to serve two purposes: firstly, it provides the reader with enough knowledge to follow the original research that follows, and secondly, it motivates the theoretical analysis and the development of the novel methods herein.

In Chapter 3, we delve deeper into an analysis of the density peaks clustering method as it is the central focus of this thesis. We provide a formal introduction of this approach, detailing the the density estimator used, as well as the sample-based assignment procedure used to extract clusterings from the data. Next, an analysis of the theoretical performance of the density peaks clustering approach is undertaken. Works discussed in Chapter 2 provide a useful theoretical framework in which to analyze the density peaks clustering algorithm. We provide consistency guarantees for the estimates of the modes produced by the density peaks clustering algorithm, and demonstrate that the algorithm correctly clusters the data with high probability. The behaviour of the density peaks clustering method for datasets of typical size is also of interest. Illustrative datasets are used to assess the density peaks clustering algorithm in the presence of features that can hinder clustering performance. This analysis provides insights into the deficiencies of the peak-finding approach.

Three novel clustering methods are included in this thesis, two non-parametric clustering methods, one in Chapter 4 and one in Chapter 5, and a parametric clustering approach in Chapter 7. For each approach, we use broadly the same research methodology. Each method is motivated using a worked example, and then the details of its

operation are provided. Having provided algorithmic details for the method, an analysis of the approach is then undertaken. The analysis has the following components: firstly, relevant theoretical results that provide justification for and insights into the performance of the approach are developed; for the non-parametric clustering methods the computational complexity of the algorithms is then analysed; all methods are assessed on simulated data to demonstrate their performance; experiments on real-world datasets are used to further validate the performance results; and finally, we provide insights regarding the effect of the hyper-parameters for each approach.

1.3 Thesis Structure

We conclude our introduction with a brief summary of the the material contained in the chapters of this thesis.

- *Chapter 2 - Literature Review:* The fundamental aspects of density-based clustering used throughout the thesis are formally introduced. The two notions of density-based clustering, parametric and non-parametric are described. For each formulation, we outline the chosen conception of clusters, the population goal for the clustering, and review popular estimation procedures. For parametric clustering methods, this involves formalizing the definition of a mixture density, with particular focus on a mixture of multivariate Gaussian distributions. The method used to derive maximum likelihood estimates for the parameters is discussed, and the EM algorithm is introduced. Finally, we discuss prominent methods used for model selection for parametric clustering. The non-parametric notion of population clusters is then introduced, formalizing the concept of the domain of attraction of a density mode. The review of prominent estimation procedures first considers non-parametric density estimation methods, before introducing the two contrasting approaches to non-parametric clustering. Level set methods are described, along with a popular generalization of density-level set methods termed the cluster tree. The section continues with a review of mode-

seeking methods, considering methods based on gradient ascent of the kernel estimate and introducing sample-based approaches, notably the density peaks clustering method. The chapter concludes with a brief discussion of the challenges associated with model selection for non-parametric clustering.

- *Chapter 3 - Density Peaks Clustering:* The density peaks clustering method, as formulated in this thesis, is introduced. The density estimator, based on nearest neighbors is formalized and the methodology of density peaks clustering is described in detail. Subsequently, a theoretical analysis of the density peaks clustering algorithm is provided. It is shown that the density peaks clustering method recovers consistent estimates of the modes using observed samples of the data. Furthermore, the density peaks clustering method correctly assigns the remaining instances to their respective clusters with high probability. This analysis relies on the consistency properties of the density estimator as the number of observations increases. The performance of the density peaks clustering method for datasets of typical size is then investigated. This analysis concludes that the mode estimates returned by density peaks clustering method are susceptible to errors caused by noise in the density estimate, and that the allocation mechanism used can lead to incoherent and incorrect clusterings.

- *Chapter 4 - Modal Set Detection with the Peak-Finding Criterion:* In this chapter, a method aiming at improving the ability of the density peaks clustering algorithm to model high density regions of the data is introduced. Following the analysis of Chapter 3, it is seen that density peaks clustering often fails to adequately represent clusters with areas of relatively uniform density, as noise in the density estimate leads to the detection of erroneous mode estimates. To improve the clustering performance, the novel algorithm directs the peak-finding technique to discover modal sets, rather than point modes in the data. By modelling high density regions in the data using modal sets, the algorithm is robust to noise in the density estimate and thus detects clusters at varying densities, and is competent at deciding the correct number of clusters, even when the num-

ber of clusters is very high. A theoretical analysis of the approach is presented and experimental results verify that the algorithm works well in practice and executes efficiently. The chapter concludes with a demonstration of a potential application of this method for unsupervised face recognition.

- *Chapter 5 - Peak-Finding on Density-Level sets:* Following from the method introduced in Chapter 4 that provides reliable estimates of the high density regions in the data, this chapter develops a novel method aiming at improving the allocation mechanism of the density peaks clustering algorithm. An example is given in which the allocation strategy returns clusterings that contradict geometrical intuition and the notion of a cluster introduced in Chapter 2. Motivated by this example, the novel method developed in this chapter aims to remedy the issues with the allocation approach by combining the benefits of both density-level set and mode-seeking methods for non-parametric density-based clustering. The methodology for the new approach is described. An experimental analysis on simulated and real-world data demonstrates the benefits of integrating these two formulations. Finally, a modified version of the approach is presented, that incorporates instance-level constraints in the clustering scheme. This modified version of our approach is shown to achieve excellent performance for an important problem in computer vision, multi-image matching.

- *Chapter 6 - Experimental Comparison:* This chapter provides an experimental evaluation of the methods introduced in Chapter 4 and Chapter 5. The novel approaches are assessed in comparison with prominent non-parametric density-based clustering methods. The methods are each applied to ten real-world datasets and their performance is quantified using two popular validation indices as well as the execution time of each algorithm. Subsequently, an analysis of the parameter space for each algorithm is provided. It is shown that the methods developed as part of this thesis achieve excellent clusterings over a broad range of parameter values. Guidance on hyper-parameter tuning is also provided.

- *Chapter 7 - Density Peaks for Parametric Clustering:* This chapter applies the density peaks clustering algorithm for an important problem in parametric clustering, namely providing initializations for the EM algorithm. A novel method is introduced that applies the peak-finding approach to generate an inclusive set of initial mean estimates from the data pool. Subsequently, an efficient pruning strategy is described. Redundant exemplars are pruned by penalizing a convex objective function that is well justified in the context of mixture modelling. Guarantees on the quality of the initialization are discussed and a method for finding analytical solutions to the convex optimization problem is provided. An experimental analysis verifies that the method executes efficiently and outperforms prominent competitor methods in practice. The selection of hyper-parameters is discussed.

- *Chapter 8 - Conclusions:* The final chapter of the thesis gives a brief summary of the research conducted and the conclusions reached. Several suggestions for further research are suggested. These suggestions include using mixture density estimates as inputs for the density peaks clustering algorithm, methods to improve the performance of non-parametric clustering algorithms for high-dimensional datasets, and the related tasks of manifold and subspace clustering. The development of model selection criteria for non-parametric methods is also discussed as an important issue for future research to consider. Two extensions for the REM algorithm are proposed, the first incorporates sparse covariance structures into the implementation of the approach, and the second extends the REM formulation for mixtures of component distributions other than the multivariate Gaussian. The final proposal encourages the consideration of providing initializations using the density peaks clustering method for mixture models with estimation algorithms other than EM. A potential avenue of research using Riemannian optimization is discussed.

2 Literature Review

Many hundreds of clustering methods have been proposed in the literature, far too many to exhaustively review. A taxonomy of clustering methods is available in Xu and Tian (2015). Broadly, the most popular and prominent clustering methods are based on some notion of distance or dissimilarity. Hierarchical algorithms aim to recursively find nested clusters by merging or splitting groups based on some notion of their similarity. Partitional methods, such as k-means, use iterative assignment to minimize a distance-based objective. While such approaches are interpretable and conceptually simple, they are based on heuristic notions of cluster structure and, thus, they lack a precise definition of the clusters inhibiting the use of formal inferential techniques. Furthermore, we are prevented from evaluating the clustering returned by such approaches or comparing with alternatives.

Without a clear criterion available to prefer one clustering over another, or even guarantee that the given clustering algorithm is suitable for the task, the clustering problem is inherently ill-posed (Domany, 1999). In a widely cited work, clustering was judged to be a field "where rigorous methodology is still striving to emerge" (Meilă, 2007).

An attempt to formalize the clustering task, and navigate a path from the ill-posedness of heuristic methods, relates clusters to the probability density function assumed to underlie and generate the observed data set. Several reasons for such an idea being appealing are provided in Casa (2019). Firstly, it links the clustering task to some well-defined population goal allowing a notion of successful (and unsuccessful) clustering to be defined. Furthermore, by relating the clusters to properties of the underlying

density, as opposed to defining cluster notion in relation to the observed samples, one allows for a partitioning of the entire sample space rather than just the observed data. This allows the clustering of new data, if it is provided. Thirdly, following this notion leads to the number of clusters detected becoming an intrinsic property of the data generating mechanism, and thus the choice of the number of clusters to return becomes a model selection problem, or at least a topic to be considered in the modelling process. The concept of linking clusters to the underlying probability density function has become increasingly accepted in recent years, with Carlsson and Mémoli declaring that "density needs to be incorporated into ... clustering procedures" (Carlsson and Mémoli, 2013).

Menardi (2016) describes the two distinct directions in which this formulation of the clustering problem has been pursued in research. The parametric, or model-based, approach takes the underlying probability density function to be a mixture of sub-populations, each with an assumed parametric form. The clusters are conceived to have a one-to-one relationship with the mixture components. This approach is prominent in applications and remains an active field of research. The second, less common, density-based clustering approach is henceforth referred to as non-parametric or modal clustering. In this formulation, a correspondence is drawn between the clusters and the modes of the underlying density. While both approaches have similar motivation in linking the conception of the clusters to features of the underlying density, they exhibit different practicalities, estimation procedures, and capabilities. Furthermore, they have, until recently, been pursued by separate communities with little interaction.

The conception of clusters for parametric and non-parametric density-based clustering methods are introduced, and the popular literature is reviewed. Furthermore, prominent methods used to estimate the cluster in each formulation are described.

2.1 Parametric Density-Based Clustering

2.1.1 Population Clusters

The first work to associate mixture models to clustering is often said to be the thesis of Wolfe (1963) in which the definition of a cluster is "a distribution which is one of the components of a mixture of distributions". It was noted by McNicholas (2016a) that a similar definition of cluster, or type, was defined earlier in Tiedeman (1955). The definition of the clustering problem given by Tiedeman is illuminating for the discussion of how components of a mixture distribution are conceived as clusters. Tiedeman considers several observation matrices each of which generates a Gaussian random variable. He asks if one "throw[s] away the type identification of each observation set" to leave a "mixed series of unknown density form", can one "solve the problem of reconstructing the ... density functions of original types?"

The review of McNicholas (2016a) proposes a refined definition of a cluster in the context of mixture models. In this conception, "a cluster is a unimodal component within an appropriate finite mixture model". A discussion of this definition, including its relation to definitions of a cluster based solely on modes of the density and how one can judge the appropriateness of the mixture model, is provided in McNicholas (2016b). The key commonality for each definition, however, is the relationship between components of a mixture distribution and clusters in the model-based clustering formulation.

Let $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ denote the data matrix: $\boldsymbol{X}^T = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$, where the superscript T is the transpose operator. We say that $\boldsymbol{X}$ arises from a mixture distribution if, for all $\boldsymbol{x} \in \boldsymbol{X}$, its density can be written as

$$f(\boldsymbol{x}|\boldsymbol{\Theta}) = \sum_{j=1}^{m} \pi_j f_j(\boldsymbol{x}|\boldsymbol{\theta}_j), \tag{2.1}$$

where $\pi_j > 0$, such that $\sum_{j=1}^{m} \pi_j = 1$, are called mixing proportions, $f_j(\boldsymbol{x}|\boldsymbol{\theta}_j)$ is the

jth component density, and $\Theta = (\boldsymbol{\pi}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m)$ where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_m)$, is the vector of parameters. Typically, the components densities are taken to be of the same type, i.e., $f_j(\boldsymbol{x}|\boldsymbol{\theta}_j) = f(\boldsymbol{x}|\boldsymbol{\theta}_j)$ for all $j = 1, \ldots, m$. The mixture density $f(\boldsymbol{x}|\boldsymbol{\Theta})$ is said to be an m-component finite mixture density.

The key benefit of this approach is that it allows for the definition of the ideal clustering, in terms of the underlying population density. Given the formulation in (2.1), the population clustering $\boldsymbol{\mathcal{C}} = \{\boldsymbol{\mathcal{C}}_1, \ldots, \boldsymbol{\mathcal{C}}_m\}$, induced by the m-component finite mixture density $f(\boldsymbol{x}|\boldsymbol{\Theta})$, has ideal clusters defined as

$$\boldsymbol{\mathcal{C}}_j = \{\boldsymbol{x} \in \mathbb{R}^p : \pi_j f_j(\boldsymbol{x}|\boldsymbol{\theta}_j) \geq \pi_k f_k(\boldsymbol{x}|\boldsymbol{\theta}_k), \forall j \neq k\},$$

for $j = 1, \ldots, m$.

Most commonly, the component distributions $f(\boldsymbol{x}|\boldsymbol{\theta}_j)$ are multivariate Gaussian distributions. In this case, $f(\boldsymbol{x}|\boldsymbol{\theta}_j)$ is a $\phi(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ density function, and $\boldsymbol{\theta}_j = (\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, consisting of the mean $\boldsymbol{\mu}_j$ and a covariance matrix $\boldsymbol{\Sigma}_j \succ 0$ for the jth mixture component. The multivariate Gaussian density has the form

$$\phi(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = |2\pi\boldsymbol{\Sigma}_j|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_j)\right\}.$$

Data generated by mixtures of multivariate Gaussian densities are characterized by components centered at the means $\boldsymbol{\mu}_j$, with the density of data points increasing nearer the mean. The corresponding surfaces of constant density are ellipsoidal.

Figure 2.1 shows the density contours for a two-dimensional finite Gaussian mixture model with two mixture components. The parameters are $\pi_1 = 0.35$, $\pi_2 = 0.65$, $\boldsymbol{\mu}_1 = (0, 0)$, $\boldsymbol{\mu}_2 = (5, 5)$,

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1.25 & 0.43 \\ 0.43 & 1.75 \end{bmatrix}, \quad \text{and } \boldsymbol{\Sigma}_2 = \begin{bmatrix} 2.93 & -0.25 \\ -0.25 & 2.07 \end{bmatrix}.$$

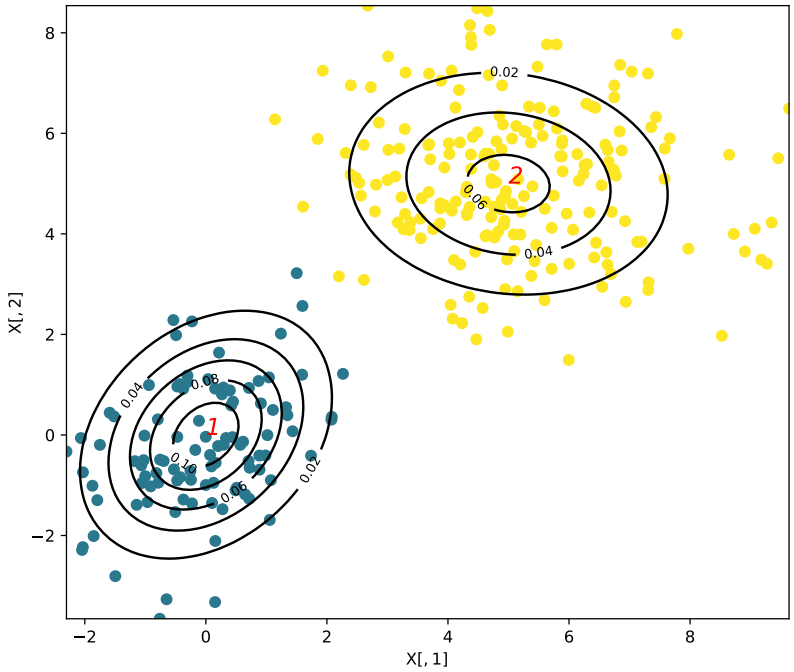


Figure 2.1: Contours of the density function for a two-component finite mixture model. The points show a sample of $n = 300$ points simulated from the density with the colour denoting the mixture component from which they were generated.

As the dimension of the data increases, mixture models struggle with heavy over-parametrization. A p-dimensional random variable following an m-component finite Gaussian mixture model has a total of

$$m - 1 + mp + \frac{mp(p+1)}{2}$$

free parameters: $m - 1$ from the mixing proportions; mp from the means; and $mp(p+1)/2$ from estimating the covariance matrices. As the number of parameters in the covariance matrix grows super-linearly with the dimension, it is common to introduce elements of parsimony in the mixture approach. The covariance matrices can be forced to be spherical, i.e., $\mathbf{\Sigma}_j = \lambda_j \mathbf{1}_p$, common across all clusters, i.e., $\mathbf{\Sigma}_j = \mathbf{\Sigma}$, or a combination, i.e., $\mathbf{\Sigma}_j = \lambda \mathbf{1}_p$. Such restrictive constraints may sacrifice too much clustering quality in the name of parsimony for a particular dataset. More flexible constraints were introduced by Banfield and Raftery (1993), who consider the eigen-decomposition

Type	Model	Volume	Shape	Orientation	$\boldsymbol{\Sigma}_j$
Spherical	EII	Equal	Spherical		$\lambda\mathbf{1}$
	VII	Variable	Spherical		$\lambda_j\mathbf{1}$
Diagonal	EEI	Equal	Equal	Axis-Aligned	$\lambda\boldsymbol{\Delta}$
	VEI	Variable	Equal	Axis-Aligned	$\lambda_j\boldsymbol{\Delta}$
	EVI	Equal	Variable	Axis-Aligned	$\lambda\boldsymbol{\Delta}_j$
	VVI	Variable	Variable	Axis-Aligned	$\lambda_j\boldsymbol{\Delta}_j$
General	EEE	Equal	Equal	Equal	$\lambda\boldsymbol{\Gamma}\boldsymbol{\Delta}\boldsymbol{\Gamma}^T$
	VEE	Variable	Equal	Equal	$\lambda_j\boldsymbol{\Gamma}\boldsymbol{\Delta}\boldsymbol{\Gamma}^T$
	EVE	Equal	Variable	Equal	$\lambda\boldsymbol{\Gamma}\boldsymbol{\Delta}_j\boldsymbol{\Gamma}^T$
	EEV	Equal	Equal	Variable	$\lambda\boldsymbol{\Gamma}_j\boldsymbol{\Delta}\boldsymbol{\Gamma}_j^T$
	VVE	Variable	Variable	Equal	$\lambda_j\boldsymbol{\Gamma}\boldsymbol{\Delta}_j\boldsymbol{\Gamma}^T$
	VEV	Variable	Equal	Variable	$\lambda_j\boldsymbol{\Gamma}_j\boldsymbol{\Delta}\boldsymbol{\Gamma}_j^T$
	EVV	Equal	Variable	Variable	$\lambda\boldsymbol{\Gamma}_j\boldsymbol{\Delta}_j\boldsymbol{\Gamma}_j^T$
	VVV	Variable	Variable	Variable	$\lambda_j\boldsymbol{\Gamma}_j\boldsymbol{\Delta}_j\boldsymbol{\Gamma}_j^T$

Table 2.1: The 14 Gaussian parsimonious mixture models grouped into three categories: spherical, diagonal, and general.

of the covariance matrices, i.e.,

$$\boldsymbol{\Sigma}_j = \lambda_j\boldsymbol{\Gamma}_j\boldsymbol{\Delta}_j\boldsymbol{\Gamma}_j^T,$$

where $\lambda_j = |\boldsymbol{\Sigma}_j|^{1/p}$, $\boldsymbol{\Gamma}_j$ is the matrix of eigenvectors of $\boldsymbol{\Sigma}_j$ and $\boldsymbol{\Delta}_j$ is the diagonal matrix of normalized eigenvalues of $\boldsymbol{\Sigma}_j$ such that $|\boldsymbol{\Delta}_j| = 1$. Eigenvalue decomposition of the covariance matrix allows for controlling the geometry of the component, where λ_j represents the volume, $\boldsymbol{\Delta}_j$ controls the shape, and $\boldsymbol{\Gamma}_j$ the orientation of the component. Celeux and Govaert (1995) introduce a family of Gaussian parsimonious mixture models by imposing constraints on the elements of the decomposed covariance structure. A summary of the models is available in Table 2.1.

The mixture model framework has been applied for component densities other than multivariate Gaussian. A review of model-based clustering approaches for continuous data that are based on finite mixture models other than the Gaussian mixture model is provided in Bouveyron et al. (2019, Chapter 9). Such approaches are useful when the clusters exhibit heavy tails, or non-elliptical structure which cannot be well-accounted for with a Gaussian component. Mixtures of multivariate-t distributions, skew-normal distributions, skew-t distributions, as well as various transformation methods are con-

sidered.

2.1.2 Estimation Procedures

To obtain a partition of a dataset from the mixture model formulation, the parameter vector $\boldsymbol{\Theta}$ must be estimated. The most common way to do this is through the maximization of the likelihood. Maximization is carried out using the Expectation-Maximization (EM) algorithm. EM offers a general approach to likelihood maximization in a variety of incomplete data situations (Dempster et al., 1977). A detailed introduction to EM for finite mixture models is available in Bouveyron et al. (2019, Chapter 2). This approach involves augmenting the interpretation of the dataset, from consisting solely of instances $\{\boldsymbol{x}_i\}_{i=1}^n$ to a set of n multivariate observations $\{(\boldsymbol{x}_i, \boldsymbol{z}_i)\}_{i=1}^n$, in which $\boldsymbol{x}_i$ is observed as before and $\boldsymbol{z}_i$ is unobserved. If the $(\boldsymbol{x}_i, \boldsymbol{z}_i)$ are independent and identically distributed according to a probability distribution f with parameters $\boldsymbol{\Theta}$, then the complete-data likelihood is

$$\mathcal{L}_C(\boldsymbol{X}, \boldsymbol{Z}|\Theta) = \prod_{i=1}^{n} f(\boldsymbol{x}_i, \boldsymbol{z}_i|\Theta),$$

where $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ as before, and $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n)$. As the $\boldsymbol{z}$ are not observed, we formulate the observed data likelihood, often referred to as just the likelihood, by integrating the unobserved data out of the complete-data likelihood

$$\mathcal{L}(\boldsymbol{X}|\Theta) = \int \mathcal{L}_C(\boldsymbol{X}, \boldsymbol{Z}|\Theta) d\boldsymbol{Z}.$$

This can be written for a Gaussian mixture model as

$$\mathcal{L}(\boldsymbol{X}|\boldsymbol{\pi}, \{\boldsymbol{\mu}_j\}_{j=1}^m, \{\boldsymbol{\Sigma}_j\}_{j=1}^m) = \prod_{i=1}^{n} \sum_{j=1}^{m} \pi_j \phi(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j).$$

In general, it is convenient to work with the likelihood function after a log transformation is applied. We henceforth refer to this as the log-likelihood. The log-likelihood

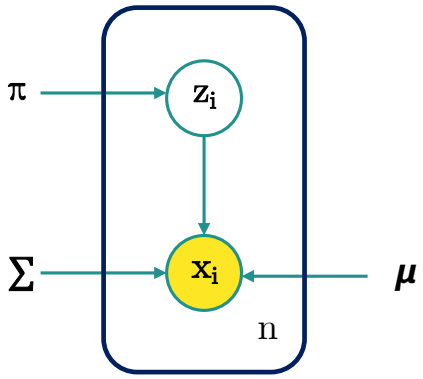


Figure 2.2: Graphical model representing a Gaussian mixture model. The random variables, $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$ are shown in the circles. The observed data is shown in yellow, with the unobserved variables shown in white. The blue box contains only the n conditionally independent instances. The unknown model parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}$, and $\boldsymbol{\Sigma}$ are also shown. In this representation, an edge between two nodes is used to indicate that the corresponding variables are conditionally independent given the intermediate variables.

for a Gaussian mixture model is

$$\ell(\boldsymbol{X}|\boldsymbol{\pi}, \{\boldsymbol{\mu}_j\}_{j=1}^m, \{\boldsymbol{\Sigma}_j\}_{j=1}^m) = \sum_{i=1}^n \sum_{j=1}^m \pi_j \phi(\boldsymbol{x}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j). \tag{2.2}$$

To estimate the parameters of the mixture model, the EM algorithm alternates between two steps. The first, "E-step", computes the conditional expectation of the complete data log-likelihood from the observed data and the current values estimated for the parameters. The second, "M-step", updates the parameter values to maximize the expected log-likelihood from the "E-step".

To apply the EM approach for clustering with finite mixture models, we take the unobserved data $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{im})$ to be the unobserved partition of the data, with

$$z_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{x}_i \text{ is generated by component } j, \\ 0 & \text{otherwise.} \end{cases}$$

The Gaussian mixture model is summarized in a graphical model in Figure 2.2. The unobserved data $\boldsymbol{z}_i$ are assumed to be independent and identically distributed, according

to a multinomial distribution with m categories with event probabilities $(\pi_1, \ldots, \pi_m)$. Taking, by assumption, that the density of an observation $\boldsymbol{x}_i$ given $\boldsymbol{z}_i$ is by $\prod_{j=1}^{m} \phi(\boldsymbol{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, the resulting complete-data likelihood is

$$\mathcal{L}_C(\boldsymbol{\pi}, \{\boldsymbol{\mu}_j\}_{j=1}^m, \{\boldsymbol{\Sigma}_j\}_{j=1}^m, \{\boldsymbol{z}_i\}_{i=1}^n|\boldsymbol{X}) = \prod_{i=1}^{n} \pi_j \phi(\boldsymbol{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)^{z_{ij}},$$

and the log-likelihood maximized by the EM algorithm is

$$\ell_C(\boldsymbol{\pi}, \{\boldsymbol{\mu}_j\}_{j=1}^m, \{\boldsymbol{\Sigma}_j\}_{j=1}^m, \{\boldsymbol{z}_i\}_{i=1}^n|\boldsymbol{X}) = \sum_{i=1}^{n} \sum_{j=1}^{m} z_{ij} \log\left(\pi_j \phi(\boldsymbol{x}_i|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)\right).$$

The E-step of the EM algorithm at any iteration updates the estimates $\hat{z}_{ij}$ by

$$\hat{z}_{ij} = \frac{\hat{\pi}_j \phi(\boldsymbol{x}_i|\hat{\boldsymbol{\mu}}_j, \hat{\boldsymbol{\Sigma}}_j)}{\sum_{v=1}^{m} \hat{\pi}_v \phi(\boldsymbol{x}_i|\hat{\boldsymbol{\mu}}_v, \hat{\boldsymbol{\Sigma}}_v)},$$

where $\hat{\pi}_j$, $\hat{\boldsymbol{\mu}}_j$, and $\hat{\boldsymbol{\Sigma}}_j$ are the values of π_j, $\boldsymbol{\mu}_j$, and $\boldsymbol{\Sigma}_j$ at the current iteration respectively. This estimate is the conditional expectation of z_{ij} given the current parameter values, and the observed data $\boldsymbol{X}$. It represents the estimated conditional probability that observation i originates from the jth component in the mixture. It is henceforth referred to as the responsibility.

For the M-step, the estimates of the mixture parameters have closed-form expressions computed using the data and the responsibilities $\hat{z}_{ij}$. The component probabilities and mean parameter estimates are updated as

$$\hat{\pi}_j = \frac{\sum_{i=1}^{n} \hat{z}_{ij}}{n}; \quad \hat{\boldsymbol{\mu}}_j = \frac{\sum_{i=1}^{n} \hat{z}_{ij} \boldsymbol{x}_i}{\sum_{i=1}^{n} \hat{z}_{ij}}, \quad \text{for } 1 \leq j \leq m.$$

The estimate of the covariance parameters $\{\hat{\boldsymbol{\Sigma}}_j\}_{j=1}^m$ depends on the chosen parameterization, as outlined in Table 2.1 above. For the model with no constraints on the

covariance matrix, the estimate is updated as

$$\hat{\boldsymbol{\Sigma}}_j = \frac{\sum_{i=1}^{n} \hat{z}_{ij}(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_j)(\boldsymbol{x}_i - \hat{\boldsymbol{\mu}}_j)^T}{\sum_{i=1}^{n} \hat{z}_{ij}}, \quad \text{for } 1 \leq j \leq m.$$

The closed-form expressions for the updates of the remaining thirteen parameterizations of the covariance parameter are given in Celeux and Govaert (1995).

The E-step and the M-step are iterated until convergence to a local maximum of the log-likelihood function. The criteria most commonly used to assess convergence is slow changes in the log-likelihood between iterations. Once convergence has been achieved, the final partition of the data is determined using a maximum likelihood classification, namely $\boldsymbol{x}_i$ is assigned to component $h = \arg\max_{1 \leq j \leq m} \hat{z}_{ij}$.

Despite being successfully deployed in a variety of different real data applications, EM has shown certain limitations when used in clustering with Gaussian mixture models (Bishop, 2006, Chapter 9). Firstly, it may converge to a singularity at which the likelihood is infinite, leading to meaningless estimates. Furthermore, the rate of convergence of the EM algorithm can be slow. The hill-climbing nature of the algorithm, coupled with the multi-modal surface of the log-likelihood function, means the resulting solution is a local optimum in the neighborhood of the initial guess. As such, the performance of EM is sensitive to the initialization. The simplest stochastic strategy is random initialization. Jin et al. (2016) proved that, with high probability, the EM algorithm with random initialization will converge to bad local maxima, whose log-likelihood could be arbitrarily worse than that of the global maximum. A prominent deterministic method, implemented in the R package *mclust* (Scrucca et al., 2016), initializes the EM algorithm with the solution of model-based Gaussian hierarchical clustering.

The challenges in initializing the EM algorithm are compounded by the fact that, generally, the true number clusters is unknown. A common practice is to run the EM algorithm with an initialization method to estimate a set of models corresponding to different numbers of mixture components, different specifications for the component

densities, or distinct parameterizations of the component densities. Subsequently, the optimal clustering model is selected via a model selection criterion.

Model selection is usually carried out using an information criterion, with the Bayesian Information Criterion (BIC) (Schwarz, 1978) being the most prominent. It is defined for a Gaussian mixture model as

$$\text{BIC} = 2\ell\left(\boldsymbol{X}|\hat{\boldsymbol{\pi}}, \{\hat{\boldsymbol{\mu}}_j\}_{j=1}^m, \{\hat{\boldsymbol{\Sigma}}_j\}_{j=1}^m\right) - \zeta \log n,$$

where $\ell(\cdot)$ is defined in (2.2) and ζ is the number of free parameters in the model, acting as a proxy for the complexity of the model. As such, the BIC represents the likelihood of the data under the estimated model, penalized by a sample-size dependent penalty that encourages the selection of more parsimonious models. The use of the BIC is motivated through an asymptotic approximation of the log posterior probability of the models assessed (Kass and Raftery, 1995). While the usual regularity conditions used by Schwarz in the development of the BIC are not generally satisfied by mixture models, it has been shown that the BIC gives consistent estimates of the number of components in a mixture model. Furthermore, the BIC has achieved excellent performance in many practical studies and remains the standard model selection criterion for mixture models.

Nevertheless, alternatives to the BIC for mixture model selection exist. The Integrated Completed Likelihood (ICL) (Biernacki et al., 2003) further penalizes the BIC by subtracting the estimated mean entropy.

$$\text{ICL} \approxeq \text{BIC} + 2\sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{1}\left(j = \underset{1\leq h\leq m}{\arg\max}\, \hat{z}_{ih}\right) \log \hat{z}_{ij},$$

where $\mathbf{1}(\cdot)$ is the identity function. The second term reflects the uncertainty in the final partition using the entropy of the responsibilities for an instance. As such, the ICL tends to select more parsimonious models, where the separation among clusters is more clear.

A third model selection criterion, the Akaike Information Criterion (AIC) Akaike (1974) is defined as

$$\text{AIC} = 2\ell\left(\boldsymbol{X}|\hat{\boldsymbol{\pi}}, \{\hat{\boldsymbol{\mu}}_j\}_{j=1}^m, \{\hat{\boldsymbol{\Sigma}}_j\}_{j=1}^m\right) - 2\zeta.$$

The AIC aims to minimize the Kullback-Leibler divergence between the chosen model and the true probability density function.

A detailed review of prominent model selection criteria for mixture models is available in Celeux et al. (2019). The viability of such alternatives is not discussed herein, and only the criteria introduced above are used for model selection for model-based clustering.

2.2 Non-Parametric Clustering

2.2.1 Population Clusters

As with all statistical procedures, there exist parametric and non-parametric methodologies for density-based clustering. The non-parametric formulation of density-based clustering, also termed modal clustering, conceives of clusters as regions of high density, separated from each other by regions of low density. Such a notion of cluster is attractive for several reasons: (1) the clusters are free to assume any shape in contrast to its parametric counterpart; (2) it is associated to features of the underlying probability density without requiring strong assumptions about the density itself; and (3) the number of clusters is a feature of the the data generating mechanism and can be determined as part of the estimation procedure.

While such a notion of density-based clustering is appealing, modal clustering research stalled for many years due to the infeasible computation required by many methods. Research in this area has resumed in recent years, but work remains scattered and lacks the cohesion of parametric clustering research. A recent review by Menardi (2015) provides a useful introduction to the field, including the conception of modal

clusters and methods used to recover them.

An early definition of a cluster in this field was provided by Carmichael and Julius (1968). In this work, the authors develop the maxim used regularly in non-parametric clustering, namely that clusters are regions of continuous, relatively high-density points in the space mutually separated by continuous regions of relatively empty space. They formulate the concept of relatedness between instances in this light. This definition was refined by Wishart (1969), asserting that "clusters should be distinct data modes, independently of their shapes and variance". Such a definition clearly separates the non-parametric formulation of density-based clustering from the mixture model approach, as defined previously by McNicholas. Several years later, Hartigan (1975) proposed the concept of density-contour clusters, as the maximal connected subsets of density-level sets. These density-contour clusters are to be "regions ... where the densities are high surrounded by regions where the densities are low".

While these early attempts to specify the notion of a cluster in modal clustering are heuristic, they relate the concept of population clusters to features of the probability density function, namely the modes (i.e., the local maxima). By linking the clusters to features of the population density, the ill-posedness of the clustering problem is resolved.

Hartigan (1975) makes an attempt to provide a precise link between the clusters determined by modal clustering and the underlying population density. If the distribution has a density f, given some $\lambda \geq 0$, the λ-level set of f is defined as $L(\lambda) = \{\boldsymbol{x} : f(\boldsymbol{x}) \geq \lambda\}$. Then the population clusters associated with the level λ are the connected components of $L(\lambda)$. This definition captures the notion of high-density regions mutually separated by regions of low density. While this approach clearly defines the population target, the notion of population clusters depends on the level λ. Furthermore, there are many situations where it is not possible to observe the entire cluster structure using one level λ. One such situation is shown in Figure 2.3. To navigate this situation, it is often recommended to consider the cluster structure

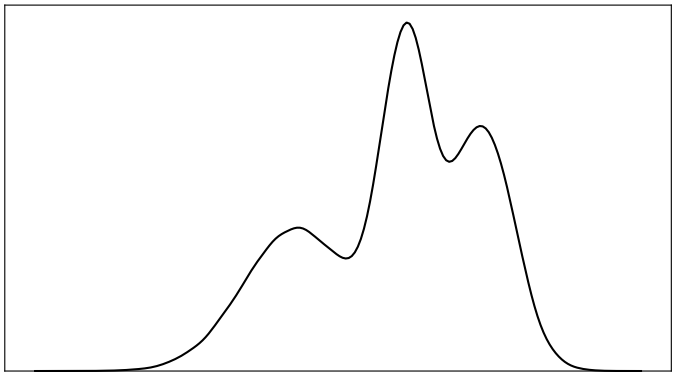


Figure 2.3: A univariate trimodal density for which no value of λ exists that captures the whole cluster structure using a level set.

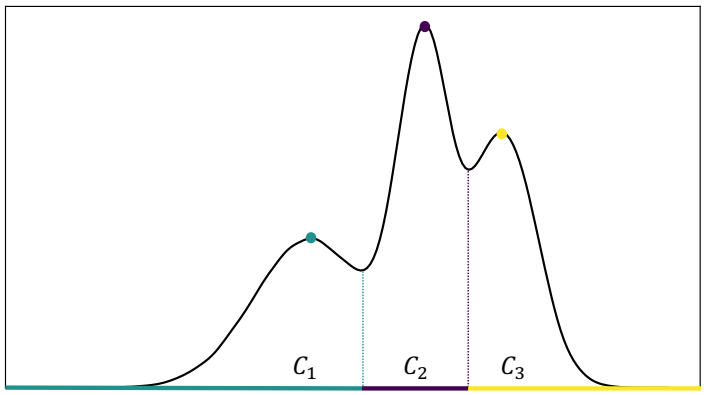


Figure 2.4: The univariate density of Figure 2.3 with the domain of attraction of each mode highlighted.

for various values of λ.

A more rigorous attempt at definition is provided by Stuetzle (2003), defining clusters in terms of the correspondence between the observed instances and the density modes. Precisely, a cluster is understood to be the "domain of attraction" of an associated mode. It is then possible to highlight correspondence between modal regions of the density and the clusters, with the modes being the archetypes of the clusters themselves.

For $p = 1$, the solution is immediate for continuous densities, the points at which f has a local maximum are representatives of the clusters and the cluster boundaries are the points are which f has a local minimum. Considering Figure 2.4, the population clusters are clear. To extend this definition for $p > 1$, Ray and Lindsay (2005) focus on determining the domains of attraction by seeking the $(p-1)$-dimensional manifolds,

termed ridges, that connect neighboring peaks. They use an analogy of a mountain range, equating the non-parametric clustering problem to the problem of describing the surface features of a land mass, where the elevation at a point (x_1, x_2) is equated with the bivariate density $f(x_1, x_2)$.

> The local maxima of the density are the peaks, and their location, together with elevation, provides a first-order description of topography. But in a richer sense, mountains are usually aggregated into mountain ranges, in which the neighboring peaks are connected through ridges. The perceived separation of two neighboring peaks is then determined by the elevation at the lowest point on this ridge, the saddle point between them.

An attempt to translate these concepts to population goal of modal clustering was provided in Chacón (2015). There, Chacón defines the ideal population goal for modal clustering as the recovery of the unstable manifolds of the negative gradient flow corresponding to local maxima of f. Assume that f is a Morse function, i.e., f is smooth enough to have non-degenerate critical points, and denote by $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ the modes of f (i.e., the local maxima). Let the path $\nu_{\boldsymbol{x}} : \mathbb{R} \to \mathbb{R}^p$ satisfy

$$\nu_{\boldsymbol{x}}(0) = \boldsymbol{x}, \quad \text{and } \nu'_{\boldsymbol{x}}(t) = \nabla f(\nu_{\boldsymbol{x}}(t)).$$

For a mode $\boldsymbol{x}_j$, its attraction region C_j is the set of points $\boldsymbol{x} \in \boldsymbol{X}$ that satisfy

$$\lim_{t \to \infty} \nu_{\boldsymbol{x}}(t) = \boldsymbol{x}_j.$$

Continuing the analogy, a modal cluster in this setting is "the region of the terrain that would be flooded by a fountain emanating from a peak of the mountain range". This definition, however, requires strict assumptions about the smoothness of f and the non-degeneracy of its critical points. Linking the definition of attraction region to the gradient flow has also been used to define modal clusters in Wasserman et al. (2014) and Arias-Castro et al. (2016). A bivariate bimodal example, including two modes

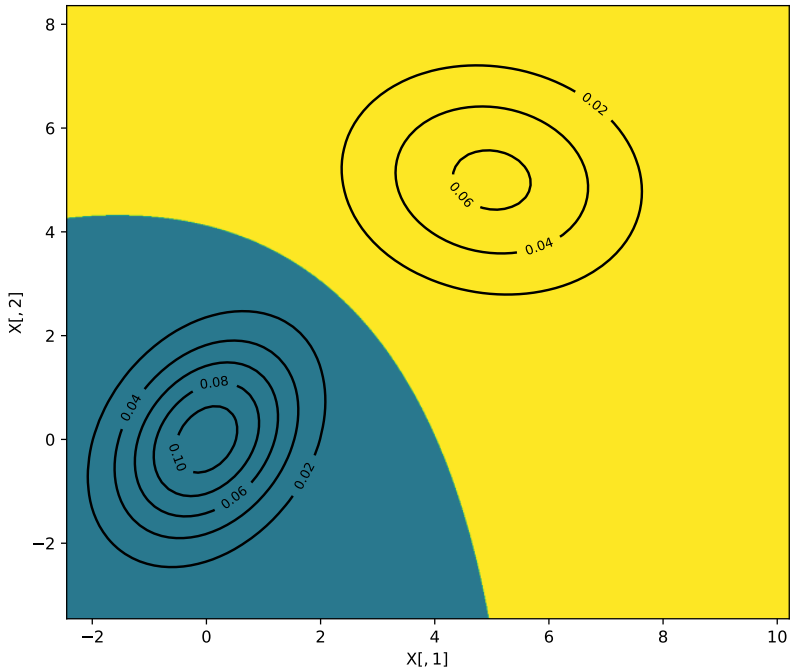


Figure 2.5: Ideal modal population clusters for the two-component mixture model example introduced in Section 2.1.1.

and a line indicating the border between the associated attraction regions is given in Figure 2.5. It is noted by Menardi (2015) that using gradient flow methods to define population clusters assumes that critical points are isolated and have distinct critical values. This fails for non-standard densities, such as those with plateaux. To navigate these issues, mild assumptions about the underlying density, and its regularity, are often required in modal clustering methods (Kpotufe and von Luxburg, 2011; Dasgupta and Kpotufe, 2014; Jiang, 2017b).

2.2.2 Estimation Procedures

The key requirement of all conceptions of population clusters in modal clustering is the probability density function f. In almost all situations, the density function is unknown during the clustering process. As such, a high quality estimate of f is required to ensure a high quality clustering. The primary estimator used in the literature is the kernel density estimate (KDE) (Rosenblatt, 1956; Parzen, 1962). KDEs are a foundational element of non-parametric statistics, due to their simple computation and performance in practical application.

Kernel Function	Equation $K(u)$	Kernel Function	Equation $K(u)$
Uniform	$\frac{1}{2}$	Gaussian	$\frac{1}{\sqrt{2\pi}} e^{\left(-\frac{1}{2}u^2\right)}$
Sigmoid	$\frac{2}{\pi}\frac{1}{e^u+e^{-u}}$	Cosine	$\frac{\pi}{4}\cos\left(\frac{\pi}{2}u\right)$
Triangular	$1-\lvert u\rvert$	Epanechnikov	$\frac{3}{4}(1-u^2)$

Table 2.2: Commonly used univariate kernel functions. Note that each kernel is supported on $u \in [-1, 1]$.

The definition of a KDE begins with the choice of a kernel function $K : \mathbb{R}^p \to \mathbb{R}_{\geq 0}$ where $\mathbb{R}_{\geq 0}$ denotes non-negative real numbers such that

$$\int_{\mathbb{R}^p} K(u)du = 1.$$

Typically, $K(\cdot)$ is taken to be a non-increasing, smooth, and symmetric function. Some common choices for the kernel function K can be found in Table 2.2. The KDE also requires specification of the bandwidth $h > 0$ in the univariate case. In the multivariate case, the scalar bandwidth can be used to adapt the KDE to multivariate data, i.e., $\mathbf{H} = h^2\boldsymbol{I}$ or a bandwidth matrix $\mathbf{H}$ may be used to generalize from the scalar bandwidth. The bandwidth matrix $\mathbf{H}$ is a positive definite and symmetric $p \times p$ matrix. The KDE is thus given by

$$\hat{f}_{\mathbf{H}}(\boldsymbol{x}) = \frac{1}{n} \cdot |\mathbf{H}|^{-p/2} \sum_{i=1}^{n} K\left(\mathbf{H}^{-1/2}(\boldsymbol{x} - \boldsymbol{x}_i)\right).$$

It has been proved that the choice of kernel function has little impact on the quality of the density estimate. In contrast, appropriate selection of the bandwidth matrix $\mathbf{H}$ is crucial. For simplicity, we restrict the review to scalar bandwidths, i.e., $\mathbf{H} = h^2\boldsymbol{I}$. We thus denote $K_{\mathbf{H}}$ and $\hat{f}_{\mathbf{H}}$ as K_h and $\hat{f}_h$ respectively. To this end, a number of methodologies for bandwidth selection have been proposed in the literature. A review of such approaches is beyond the scope of this thesis and can be found in Scott (2015) and Chacón and Duong (2018).

Significant research has aimed to quantify the quality of the KDE. A basic measure

of accuracy of the estimator $\hat{f}_h$ is the mean squared error at an arbitrary point $\boldsymbol{x} \in \mathbb{R}^p$.

$$\text{MSE} = \text{MSE}(\boldsymbol{x}) = \mathbb{E}_f \left[(\hat{f}_h(\boldsymbol{x}) - f(\boldsymbol{x}))^2 \right],$$

where "MSE" stands for mean squared error and $\mathbb{E}_f$ denotes the expectation with respect to the distribution f. Tsybakov (1997, Chapter 1) shows that if f is taken to be α-Hölder continuous, under the optimal choice of $h \approx n^{1/(2\alpha+p)}$, then

$$\text{MSE}(\boldsymbol{x}) = O\left(n^{\frac{\alpha}{(2\alpha+p)}}\right), \quad n \to \infty,$$

uniformly in $\boldsymbol{x} \in \mathbb{R}^p$. Furthermore, Jiang (2017c) provides the same rates for finite-sample ℓ_∞ bounds that hold uniformly across bandwidths with probability independent of the bandwidths. Specifically, if f is again taken to be α-Hölder continuous, under the optimal choice of $h \approx n^{1/(2\alpha+d)}$, then $|\hat{f}_h - f|_\infty \lessapprox n^{\alpha/(2\alpha+d)}$ with probability $1 - 1/n$.

Given the significance of bandwidth selection for KDE, it has proved beneficial to consider KDE-based procedures with adaptive bandwidths - i.e., when the bandwidths change depending on the region of the data. Consider again the fixed kernel estimator

$$\hat{f}_h(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{n} K_h(\boldsymbol{x} - \boldsymbol{x}_i).$$

A prominent generalization of this approach was introduced in Loftsgaarden and Quesenberry (1965). They take the bandwidth $h_{\boldsymbol{x}}$ to be a function of the prediction instance and the observed dataset $\boldsymbol{X}$. Setting $h_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{X})$ equal to the distance $r_k(\boldsymbol{x})$ from the prediction instance to the k-th nearest sample in the observed dataset yields

$$h_{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{X}) = r_k(\boldsymbol{x}) \approx \left(\frac{k}{n \cdot v_p \cdot f(\boldsymbol{x})} \right)^{1/p},$$

where v_p is the volume of the unit sphere in $\mathbb{R}^p$. If this adaptive bandwidth is used with a uniform kernel with bin widths that adapt to $\boldsymbol{x}$, the resulting density estimator

is termed the k-th nearest neighbor (k-NN) estimator. The k-NN estimator, denoted $\hat{f}_k$ is given as

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

where $r_k(\boldsymbol{x})$ and v_p are as before.

The k-NN estimator was shown by Loftsgaarden and Quesenberry (1965) to be a consistent estimate when the unknown density f is continuous on $\mathbb{R}^p$. While is it well-known and widely used in application, theoretical analysis has proved more challenging than the fixed bandwidth KDE. More general consistency results have been proved since the work of Loftsgaarden and Quesenberry (1965). Notably Moore and Yackel (1977), who show that for f Lipschitz in a neighborhood of a point $\boldsymbol{x}$ with $f(\boldsymbol{x}) > 0$, and $k = k(n)$ satisfying $k \to \infty$ and $k/n^{2/(2+p)} \to 0$, the k-NN estimator is asymptotically normal, i.e., $\sqrt{k}(\hat{f}_k(\boldsymbol{x}) - f(\boldsymbol{x}))/f(\boldsymbol{x}) \xrightarrow{\mathcal{D}} \mathcal{N}(0,1)$. As such, under the stated conditions on k, it can be expected that $|\hat{f}_k(\boldsymbol{x}) - f(\boldsymbol{x})| \lessapprox f(\boldsymbol{x})/\sqrt{k}$. The work of Biau et al. (2011) demonstrate that this result can be achieved in expectation for $n = n(\boldsymbol{x})$ sufficiently large. In particular, the conditions on k introduced in that work allow for setting $k \approx n^{4/(4+p)}$ yielding a minimax-optimal mean square error

$$\mathrm{MSE}(\boldsymbol{x}) \lessapprox \frac{f(\boldsymbol{x})^2}{k} = O\left(n^{-\frac{4}{(4+p)}}\right).$$

Dasgupta and Kpotufe (2014) provide an important contribution to understanding the conditions under which high probability bounds on $|\hat{f}_k(\boldsymbol{x}) - f(\boldsymbol{x})|$ are possible. If f is taken to be locally α-Hölder continuous at a point $\boldsymbol{x}$, setting $k = \Theta\left(n^{2\alpha/(2\alpha+p)}\right)$ yields a minimax-optimal rate of $|\hat{f}_k(\boldsymbol{x}) - f(\boldsymbol{x})| = O\left(-n^{\alpha/(2\alpha+d)}\right)$.

With high quality density estimates secured, we now consider the work investigating the ability of density estimators to recover the modes of a distribution. The most common approach to understanding the consistency of estimators is to estimate a mode $\boldsymbol{x}_j$ as $\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in \mathbb{R}^p} \hat{f}(\boldsymbol{x})$, where $\hat{f}$ is an estimate of f such as the KDE $\hat{f}_h$ or the k-NN estimator $\hat{f}_k$. This approach is termed indirect mode estimation by Devroye (1979),

as the data is used to compute a density estimator, from which the estimate of mode is then computed. Direct estimates of the modes, in the terms introduced by Devroye (1979), use a simpler approach, estimating the mode as $\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}\in\boldsymbol{X}} \hat{f}(\boldsymbol{x})$. For KDE, Abraham et al. (2004) show that this direct estimator behaves asymptotically as the indirect estimator. Dasgupta and Kpotufe (2014) demonstrate that the direct estimate of the mode using the k-NN density estimator also consistently recovers the mode of a unimodal density, and also provide conditions to ensure the estimator converges at a minimax-optimal rate. Jiang (2017c) adapts this analysis for KDE, proving that the direct estimator is a rate-optimal estimator of the mode under finite samples with appropriate bandwidth choice. In both Dasgupta and Kpotufe (2014) and Jiang (2017c), a simple scheme extends these results to mode recovery in multi-modal distributions. Connected components of nearest neighbor graphs at appropriate levels of the estimated density are used to isolate modes away from each other before applying the mode recovery analysis for unimodal distributions.

The theoretical analysis describes how connected components of nearest neighbor graphs can be used to provide guarantees on mode recovery for KDE and k-NN estimators, but these works do not give guidance on how clusters can be extracted once the mode estimates are recovered.

Connected components of nearest neighbor graphs can be used to estimate modal clusters as conceived by Hartigan (1975). As introduced in Section 2.2.1, modal clusters are connected components of the level set $L(\lambda)$. To estimate these clusters, a straightforward approach replaces f with a non-parametric estimate $\hat{f}$. Obtaining a partition using this approach can be challenging for multivariate data. A solution is provided by graph theory. Consider a graph $G(\boldsymbol{X}, E)$ consisting of the vertex set $\boldsymbol{X}$ and the edge set E. We begin with the definition of connectedness. 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 $G(\boldsymbol{X}, E)$. A connected component of a graph

$G(\boldsymbol{X}, E)$, 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: $E(\boldsymbol{S}) = \{\{\boldsymbol{x}_i, \boldsymbol{x}_j\} \in E : \boldsymbol{x}_i \in \boldsymbol{S}, \boldsymbol{x}_j \in \boldsymbol{S}\}$.[1] 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 $\boldsymbol{X}$.

The subgraph $G(\lambda)$ with vertices $\{\boldsymbol{x} \in \boldsymbol{X} : \hat{f}(\boldsymbol{x}) \geq \lambda\}$ is constructed by removing the vertices of G with estimated density less than λ, and all edges associated with these vertices. The connected components of the graph $G(\lambda)$ are determined by the observations connected by an edge, or a sequence of edges, in the graph. Maier et al. (2009) provide guidance on the number of neighbors and the method for including edges to optimally construct nearest neighbor graphs for identifying clusters. Rinaldo and Wasserman (2010) discuss the conditions that f must satisfy for modal clustering to be successfully applied.

Menardi (2015), in her review, provides a summary of other methods for constructing the graph including the Delaunay triangulation (Azzalini and Torelli, 2007) where edges between vertices if they share a boundary in the corresponding Voronoi partition of the space, and density-informed graphs (Stuetzle and Nugent, 2010) where edges are added between two vertices if no significant decline in the estimated density is observed between them.

As previously demonstrated in Figure 2.3, there may not exist a single density level λ such that each mode of the underlying density lies in a unique connected component of $L(\lambda)$. This issue with this formulation of modal clustering was recognized at its conception by Hartigan (1975). Rather than focusing on the components of the level set at a single density level, one can consider the number of connected components in the level set at the density level changes. To summarize the numbers of components at each level, Hartigan (1975) introduced the cluster tree. The cluster tree of f are the component sets of the level set $L(\lambda)$ for $\lambda \geq 0$. The components form a tree hierarchy

[1] The collision in terminology between the topological definition of connected components of a population level set $A \subseteq L(\lambda)$ and the connected component of a graph $G(\boldsymbol{X}, E)$ is avoided by referring instead to component sets induced by the connected components of the graph.

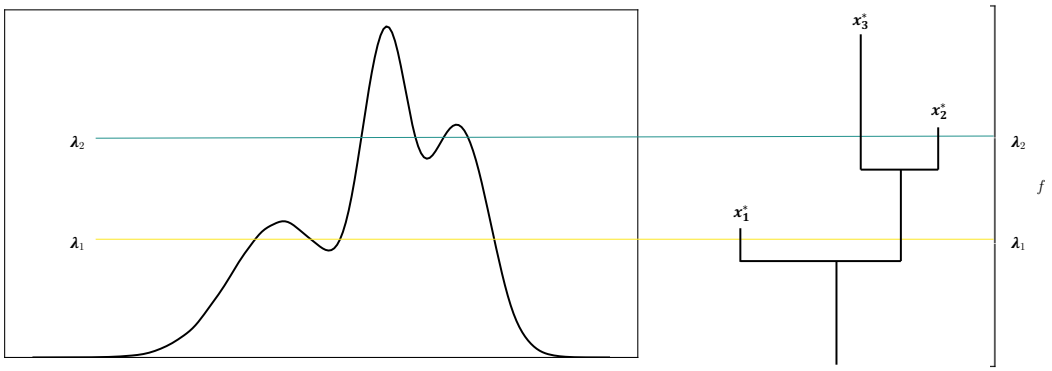


Figure 2.6: The cluster tree for the density introduced in Figure 2.3. The cluster tree shows the density levels at which the level sets split into descendant components.

where for any two components, $\boldsymbol{S}, \boldsymbol{S}'$, either $\boldsymbol{S} \cap \boldsymbol{S}' = \varnothing$ or one is a descendant of another, i.e., $\boldsymbol{S} \subset \boldsymbol{S}'$ or $\boldsymbol{S}' \subset \boldsymbol{S}$. The cluster tree for the trimodal univariate example density introduced previously is shown in Figure 2.6.

Many works have considered the problem of recovering the connected components of a density-level set at one level (Tsybakov, 1997; Maier et al., 2009; Rinaldo and Wasserman, 2010). In contrast, a smaller number of papers have aimed at estimating simultaneously all level sets of the unknown density and recovering the cluster tree as a whole. Hierarchical methods such as the single-linkage clustering algorithm have been shown to be partially consistent. Hartigan (1981) shows that single-linkage is consistent for $p = 1$, but consistency fails for $p \geq 2$. Chaudhuri et al. (2014) present two methods that consistently estimate the cluster tree. The first adapts single linkage hierarchical clustering by iteratively removing instances below the density level λ. The second approach demonstrates that a k-nearest neighbor graph can consistently recover the cluster tree for $k > 1$.

Related to density-level set methods is perhaps the most prominent density-based clustering algorithm, density-based spatial clustering of applications with noise (DBSCAN) (Ester et al., 1996). DBSCAN proceeds by computing the empirical density of each observed data point, using an estimator based on the proximity of the point to other points in the sample. Points whose densities are above a user-specified threshold are designated as core-points. A neighborhood graph of the core-points is constructed,

and the remaining instances are based on the connected components of the graph. DBSCAN has proved incredibly popular in application due to its conceptual simplicity and excellent performance. As DBSCAN was not introduced explicitly in relation to probability theory, until recently, little theoretical work to understand it was done. Several analyses have conceptualized DBSCAN as a density-level set estimator, and shown that it can consistently recover density-level sets (Sriperumbudur and Steinwart, 2012; Jiang, 2017a; Wang et al., 2017). The application of DBSCAN to contemporary datasets can prove challenging. While it was claimed that DBSCAN executes in $O(n \log(n))$ time, it was shown by Gan and Tao (2015) that DBSCAN requires at least $\Omega(n^{4/3})$ time to complete for data with more than three dimensions. There are several implementations of DBSCAN for large datasets that aim to ameliorate it's execution time by partitioning the feature space and paralellizing computation (Lulli et al., 2016; Song and Lee, 2018; Wang et al., 2020a).

DBSCAN is only able to provide a clustering for one density level. A popular generalization, Hierarchical DBSCAN (HDBSCAN), generalizes DBSCAN allowing clusters to be extracted at different levels of the density. HDBSCAN produces a version of the cluster tree, with DBSCAN clusterings at each level. An overview of DBSCAN's and HDBSCAN's many derivative methods is available in Campello et al. (2020).

A second strand of methods, termed 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 in a manner coherent with the definition of modal clustering given in Section 2.2.1. Such approaches begin with a density estimate $\hat{f}$ (typically the KDE $\hat{f}_h$) and then move each point $\boldsymbol{x}_i$ towards a mode of $\hat{f}$ evolving the trajectory $\boldsymbol{x}_i^{(t)}$, $t > 0$ starting from $\boldsymbol{x}_i^{(0)} = \boldsymbol{x}_i$ and ascending the gradient $\nabla \hat{f}(\boldsymbol{x}_i^{(t)})$. Termination criteria are required to stop the evolution and a clustering rule determines how to merge trajectory end-points.

Mean shift, introduced by Fukunaga and Hostetler (1975), and further developed by Cheng (1995) and Comaniciu and Peter (2002), is a popular mode-seeking clustering

algorithm. Mean shift is based on a rule for evolving the trajectories $\boldsymbol{x}_i^{(t)}$ when the density estimate is a KDE and the kernel function $K(u)$ can be written as $\psi(\|u\|^2)$ for a convex function $\psi(\cdot)$. This requirement does not inhibit the use of the prominent kernels of Table 2.2. The mean shift update rule is

$$\boldsymbol{x}_i^{(t+1)} = \arg\max_{\boldsymbol{x}\in\mathbb{R}^p} \frac{1}{n}\sum_{j=1}^{n} \|\boldsymbol{x}-\boldsymbol{x}_j\|_2^2 \psi'(\|\boldsymbol{x}_i^{(t)}-\boldsymbol{x}_j\|_2^2) \tag{2.3}$$

$$= \frac{\sum_{j=1}^{n} \psi'(\|\boldsymbol{x}_i^{(t)}-\boldsymbol{x}_j\|_2^2)\boldsymbol{x}_j}{\sum_{j=1}^{n} \psi'(\|\boldsymbol{x}_i^{(t)}-\boldsymbol{x}_j\|_2^2)}. \tag{2.4}$$

Mean shift moves the instance $\boldsymbol{x}_i$ along the path of steepest ascent of the KDE until the convergence in the limit. The final partition of the data is obtained by grouping the instances that converge to the same mode.

A related approach, medoid shift (Sheikh et al., 2007), modifies mean shift by constraining the trajectories $\boldsymbol{x}_i^{(t)}$ to pass through the points in $\boldsymbol{X}$. This leads to several benefits: (1) there is no need to iterate the trajectory updates, as once $\boldsymbol{x}_i^{(0)}$ is updated to $\boldsymbol{x}_i^{(1)} = \boldsymbol{x}_j^{(0)}$ for some $j \neq i$, the remaining updates are entirely determined by the updates of $\boldsymbol{x}_j^{(0)}$; and (2) the need to specify the stopping and merging criteria is obviated as the conditions are met exactly. Unfortunately, maximizing (2.3) while restricted to the dataset is computationally taxing, inhibiting practical application of this approach.

To circumvent the costly run time of medoid shift, Vedaldi and Soatto (2008) proposed a fast sample-based method, termed quick shift. Quick shift, rather than computing the gradient of the density, simply moves each instance $\boldsymbol{x}_i$ to its nearest neighbor of higher empirical density $\hat{f}_h$. Defining $b(\boldsymbol{x}_i) = \arg\min_{\boldsymbol{x}_j\in\boldsymbol{X}}\{\|\boldsymbol{x}_j-\boldsymbol{x}_i\| : \hat{f}_h(\boldsymbol{x}_i) < \hat{f}_h(\boldsymbol{x}_j)\}$, the update for $\boldsymbol{x}_i$ is thus

$$\boldsymbol{x}_i^{(1)} = b(\boldsymbol{x}_i).$$

It is noted by the authors that as there is no a-priori upper bound on the distances of the shift $\boldsymbol{x}_i^{(0)}$ to $\boldsymbol{x}_i^{(1)}$, the method will connect all points into a single tree. To return a partition of the data, the authors introduce a segmentation parameter τ, such

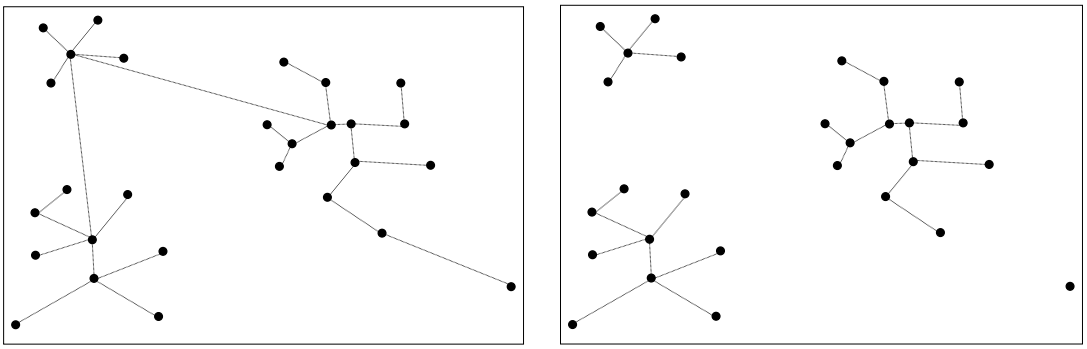


Figure 2.7: (Adapted from Figure 1 of Jiang (2017b)) Impact of the tuning parameter τ for quick shift clustering. Left: If τ is set to ∞, the quick shift procedure returns one cluster tree, with the sample of highest empirical density at its head. Right: Setting τ as a smaller value removes branches of the tree with length greater than τ. This yields the partition of the data into four groups (three clusters and one singleton point).

that the branches of the tree are removed if the length is longer than the threshold τ. Varying the parameter τ allows for clusterings at higher and lower resolutions to be investigated. The impact of the parameter τ is shown in Figure 2.7.

Quick shift has proved popular in the field of computer vision, with demonstrated applications in motion segmentation (Ayvaci and Soatto, 2009) and object localization (Fulkerson et al., 2009) for example, as it provides partitions of data at resolutions which are easily tuned by the user. Theoretical analyses of quick shift are scant in the literature. The only prominent work is that of Jiang (2017b). There, quick shift is shown to consistently estimate the significant modes of the underlying density, where significance of the modes is related to the segmentation parameter τ. Jiang also provides guarantees that quick shift correctly assigns instances to their associated mode. This ensure that quick shift returns clusterings consistent with the modal conception of clusters introduced in Section 2.2.1.

While the parameter τ provides an intuitive way to tune the clusterings returned by quick shift, appropriate tuning requires a knowledge of the distances between modes. Furthermore, having only one segmentation parameter thresholding the distances between modes can lead to outlying points being selected as modes, as can be seen on the right of Figure 2.7.

The density peaks method introduced in Rodriguez and Laio (2014) was not developed

as a response to the deficiencies of quick shift, yet it does offer a potential remedy, providing an intuitive method for sample-based mode detection, and a natural method for screening outlying points. Rodriguez and Laio (2014) use a crude estimate of the empirical density. For an instance $\boldsymbol{x}_i$, the density estimate is

$$\hat{f}_\rho(\boldsymbol{x}_i) = \sum_{j=1}^{n} \mathbf{1}\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\| < d_c\right), \tag{2.5}$$

where $\mathbf{1}(\cdot)$ is the identity function. This quantity counts the number of data points within a threshold distance d_c of $\boldsymbol{x}_i$. The intuition is that instances generated from higher density regions of the underlying probability distribution should have more instances close to them than those generated from more sparse regions of the density. This approach is easily understood and naturally adapts to the scale of the data, but requires $O(n^2)$ computation, limiting its use for large datasets.

The quantity $b(\boldsymbol{x}_i)$ is adapted for this density estimate as $b(\boldsymbol{x}_i) = \arg\min_{\boldsymbol{x}_j \in \boldsymbol{X}}\{\|\boldsymbol{x}_j - \boldsymbol{x}_i\| : \hat{f}_\rho(\boldsymbol{x}_i) < \hat{f}_\rho(\boldsymbol{x}_j)\}$. The method requires computation of the distance from each instance to its nearest neighbor of higher empirical density, denoted here as $\omega(\boldsymbol{x}_i)$, i.e.,

$$\omega(\boldsymbol{x}_i) = \|\boldsymbol{x}_i - b(\boldsymbol{x}_i)\|. \tag{2.6}$$

The true modes of the density are estimated using a decision plot, a scatter plot of $\{(\hat{f}_\rho(\boldsymbol{x}_i), \omega(\boldsymbol{x}_i)) : \boldsymbol{x}_i \in \boldsymbol{X}\}$. Intuitively, the instances that best estimate the modes of the underlying density will be those have that (1) have high empirical density and (2) are at a relatively large distance from points of higher empirical density. As such, the modes are estimated as the extreme instances on the decision plot. Once the mode estimates have been selected from the decision plot, density peaks clustering assigns instances to clusters using the same methodology as quick shift. Instances are assigned to the same cluster as their nearest neighbor of higher empirical density until a mode estimate is reached. This leads each sample point to a mode without having to recompute the density estimator at any other points. The partition of the data is extracted by grouping together instances that are assigned to the same mode. The

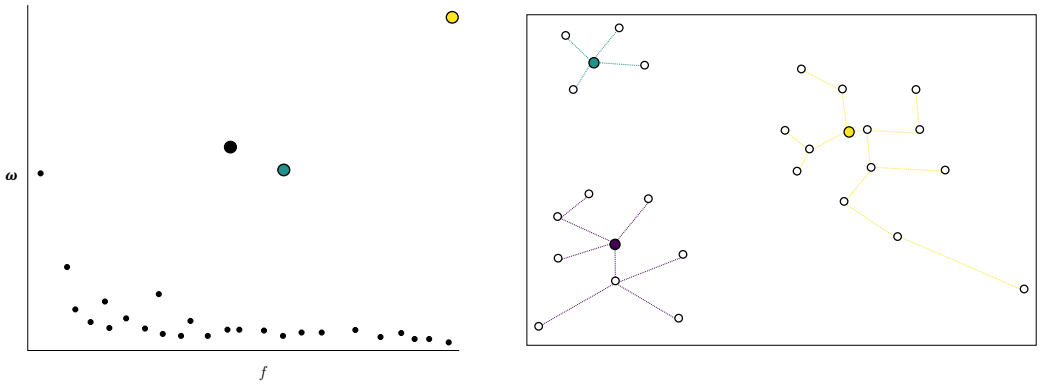


Figure 2.8: Left: The decision plot of the density peaks clustering method for the data introduced in Figure 2.7. Three estimated modes are clearly seen as extreme points in the decision plot. Right: The allocation of instances to modal clusters is the same as quick shift.

decision plot and the resulting clustering for the example data introduced in Figure 2.7 are shown in Figure 2.8.

Theoretical analysis of the density peaks method has lagged its prominence in applications. While an abundance of papers have demonstrated the ability of the peak-finding method to provide high-quality clusterings in applications (Lu et al., 2015; Ding et al., 2017; Li and Wong, 2018; Wang et al., 2018c; Platero-Rochart et al., 2022), there is, to the best of our knowledge, only one previous work analyzing the density peaks method in the terms of the probability theory considered here. Verdinelli and Wasserman (2018) analyze the decision plot, considering first the plot if the true density f was known. Assuming that f is continuous and three times differentiable, they provide guarantees that the best estimators of the modes in the sample are indeed close to the true modes of the density. Furthermore, they show that the value of $\omega(\boldsymbol{x})$ is bounded away from zero for these points. They provide guarantees on the limiting distribution of the quantity $n \cdot \omega(\boldsymbol{x})^p$ as $n \to \infty$. If $\boldsymbol{x}$ is a mode, $n \cdot \omega(\boldsymbol{x})^p \to \infty$ as $n \to \infty$. For points that are not modes $n \cdot \omega(\boldsymbol{x})^p$ is shown to converge to an exponential random variable with parameter related to the density $f(\boldsymbol{x})$. Using these insights, they derive a rule for selecting modes from the decision plot using a robust linear regression of log of the density estimates $\log \hat{f}(\boldsymbol{x})$ against the log of the distance values $\log \omega(\boldsymbol{x})$.

Adaptations of the density peaks clustering methods have proliferated in research in

recent years. Many works focus on improving the density estimator from the crude and computationally expensive estimate employed in Rodriguez and Laio (2014). A popular approach estimates the local density by applying kernel functions on nearest neighbors (see Xie et al. (2016) and Yaohui et al. (2017)), while other methods estimate the local density using the ratio of maximal distance to average distance in the set of nearest neighbors (Hou and Pelillo, 2016). The scale of these density estimates has a great influence on the selection of centers from the decision plot. Recent works have further improved the execution time of DPC by incorporating fast nearest neighbor methods, for example, the FastDP method of Sieranoja and Fränti (2019).

A second strand of research develops methods to automate the selection of centers from the decision plot, using criteria such as the silhouette index and the generalized extreme value distribution (Wang and Xu, 2017; Ding et al., 2018; Wang et al., 2020b). Previous attempts to improve the detection of low density modes involve incorporating information about points of lower local density into the density calculation (Chen et al., 2018) and the distance calculation (Li and Tang, 2018). Such approaches are sensitive to small variations in the underlying density.

Model selection methods for non-parametric density-based clustering methods are significantly less developed than for parametric model-based clustering methods. The primary reason for this is the hard assignment produced by non-parametric methods. For hard clustering algorithms, instances are assigned to one cluster only and no information about the uncertainty regarding the assignment is provided. By contrast, model-based clustering methods provide an assignment vector for each instance, describing a probability distribution over the clusters. From an assignment vector, it is possible to intuit two forms of uncertainty regarding the cluster assignment. The first corresponds to the uncertainty caused by the density estimator $\hat{f}$ compared to the underlying density f. The second form of uncertainty, termed the population uncertainty by Chen et al. (2016), captures how uncertain the relationship between an instance and the true density modes is. For example, if an instance lies on the boundary between the attraction regions of two modes, the assignment vector should reflect this

uncertainty. Quantifying the uncertainty of the assignment of each instance allows for the computation of model selection criteria for the method as a whole. However, while a model selection approach for a soft clustering adaptation of mean shift was introduced in Chen et al. (2016), it is not applicable to other non-parametric clustering methods.

Several internal validation indices have been proposed in the literature, and reviewed by Liu et al. (2010). However, the authors conclude that no measure performs consistently well for all datasets. The problem of developing broadly applicable validation indices is a challenging one, and no method been widely adopted for use in the assessment of new methods. As a result, non-parametric clustering methods are typically assessed on datasets for which true class labels are available using external validation indices such as the adjusted Rand index (Hubert and Arabie, 1985), the adjusted mutual information (Vinh et al., 2010), and the normalized mutual information (Strehl and Ghosh, 2002). The issues with this assessment regime have been well covered, yet it remains the standard approach in this field. As such, validation through external indices forms the basis for many of the performance assessments in this work.

3 Density Peaks Clustering

3.1 Summary

As discussed in Section 2.2.2, there have been many variants of the peak-finding method proposed in the literature. The formulation of the peak-finding method used herein is now formally introduced. Furthermore, a theoretical analysis of the ability of the peak-finding method to detect modes and clusters in the data is provided. Finally, limitations of the peak-finding approach are demonstrated using an illustrative experimental analysis.

3.2 The Method

The peak-finding method in Rodriguez and Laio (2014) 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* $\mathbb{R}^p$*.*

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 Rodriguez and Laio (2014), the decision plot is the scatter plot of $\{(\hat{f}_k(\boldsymbol{x}), \omega(\boldsymbol{x})) : \boldsymbol{x} \in \boldsymbol{X}\}$. A second plot, herein referred to as the peak-finding plot captures the values of the peak-finding criterion. Assuming they are sorted in decreasing order, the peak-finding plot is the scatter plot of $\{(i, \gamma(\boldsymbol{x}_i)) : i = 1, \ldots, n\}$. 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. $\widehat{\mathcal{M}} = \{\boldsymbol{x} \in \boldsymbol{X} : \hat{f}_k(\boldsymbol{x}) \geq l, \omega(\boldsymbol{x}) \geq \tau\}$.

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 plot $\{(\hat{f}_k(\boldsymbol{x}), \omega(\boldsymbol{x})) : \boldsymbol{x} \in \boldsymbol{X}\}$.
3: Sort the $\boldsymbol{x}$'s in decreasing order of $\gamma(\boldsymbol{x})$ values.
4: Create the peak-finding plot $\{(i, \gamma(\boldsymbol{x}_i)) : i = 1, \ldots, n\}$.
5: Select the estimated modes using the thresholds l and τ, i.e., $\{\boldsymbol{x} \in \boldsymbol{X} : \hat{f}_k(\boldsymbol{x}) \geq l, \omega(\boldsymbol{x}) \geq \tau\}$
6: Add the estimated modes $\{\boldsymbol{x}_j\}_{j=1}^m$ to $\widehat{\mathcal{M}}$.
7: **for** each $\boldsymbol{x}$ in $\boldsymbol{X} \backslash \widehat{\mathcal{M}}$ **do**
8: Add a directed edge from $\boldsymbol{x}$ to $b(\boldsymbol{x})$.
9: **end for**
10: **for** each estimated mode $\boldsymbol{x} \in \widehat{\mathcal{M}}$ **do**
11: Let $\boldsymbol{C}$ be the collection of the points connected by any directed path in $\vec{G}(\boldsymbol{X}, \vec{E})$ that terminates at $\boldsymbol{x}$.
12: Add $\boldsymbol{C} \cup \boldsymbol{x}$ to $\widehat{\mathcal{C}}$.
13: **end for**
14: **return** $\widehat{\mathcal{C}}$

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. Density peaks clustering produces the decision plot and, having sorted the values for the peak-finding criterion $\gamma(\boldsymbol{x})$, the peak-finding plot (Lines 2-4).

Density peaks clustering requests the user to select estimated modes using these two plots as reference. The estimated modes $\{\boldsymbol{x}_j\}_{j=1}^m$ are then added to $\widehat{\mathcal{M}}$ (Lines 5-6). 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 7-9). 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 10-13). Proceeding in this way, each sample point will be assigned to a unique cluster.

3.3 Theoretical Analysis

Density peaks clustering is a simple and popular procedure that recovers modes from intuitive plots, and provides clustering assignments to the appropriate modes. The quality of the clusterings provided by density peaks clustering has been thoroughly demonstrated in practice, as discussed in Section 2.2.2. Yet no previous work has provided guarantees on the ability of density peaks clustering to recover modes consistently. This dearth of research has also hindered understanding of the similarities and differences between density peaks clustering and peer modal clustering methods. We show that density peaks clustering can recover the modes and the associated cluster assignments with strong consistency guarantees. This analysis is adapted from the theoretical work in Dasgupta and Kpotufe (2014) and Jiang (2017b).

We assume that f is α-Hölder continuous and lower bounded on $\mathcal{X}$.

Assumption 1 (Hölder Continuity)**.** *f is Hölder continuous on the compact support $\mathcal{X} \subseteq \mathbb{R}^p$, i.e., $\exists r, L, \alpha > 0$ such that for all $\boldsymbol{x}' \in B(\boldsymbol{x}, r)$, $|f(\boldsymbol{x}) - f(\boldsymbol{x}')| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|^\alpha$ $\forall \boldsymbol{x} \in \mathcal{X}$. Furthermore, there exists $\lambda_0 > 0$ such that $\inf_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}) \geq \lambda_0$.*

As the density peaks clustering procedure moves instances to nearby areas of higher density incrementally, areas of the density where there is little or no change in the density are problematic. As such, it is assumed that the level sets of f are continuous with respect to the density level. The ϵ-interior of a set A is denoted as $A_\epsilon^\circ = \{\boldsymbol{x} \in A : \min_{\boldsymbol{x}' \in \partial A} |\boldsymbol{x} - \boldsymbol{x}'| \geq \epsilon\}$, where ∂A is the boundary of A. An illustrative example is provided in Figure 3.1.

Assumption 2 (Uniform Continuity of the Level Sets)**.** *For every $\epsilon > 0$, there exists $\delta > 0$ such that for $0 < \lambda \leq \lambda' \leq \|f\|_\infty$ with $|\lambda - \lambda'| < \delta$, then $L(\lambda)_\epsilon^\circ \subseteq L(\lambda')$.*

We now provide the uniform bounds on the k-NN density estimator required for the analysis. The results follow Lemma 3 and Lemma 4 of Dasgupta and Kpotufe (2014).

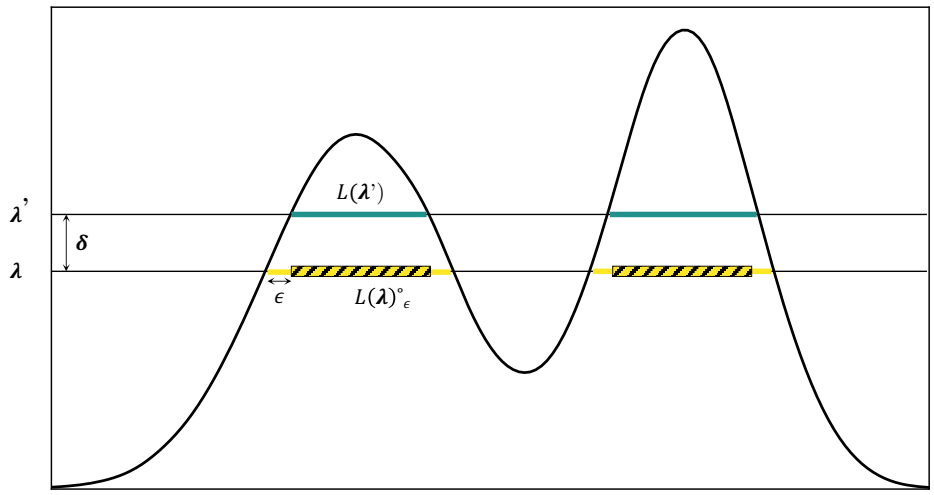


Figure 3.1: Illustrative example of Assumption 2. The ϵ-interior of the level set set $L(\lambda)$, namely $L(\lambda)^\circ_\epsilon$, is seen to be contained within $L(\lambda')$, the level set at the nearby density level λ'.

Lemma 1. *Let $\zeta > 0$. Suppose f satisfies Assumptions 1 and 2. Suppose also that $k = k(n)$ is chosen such that $\log^2 n/k \to 0$ and $n^{4/(4+p)} \to 0$. Then there exists a constant $c = c(f)$, depending on f, such that the following holds if $n \geq c^2_{\zeta n}$, with probability at least $1 - \zeta$.*

$$\sup_{\boldsymbol{x} \in \boldsymbol{X}} |\hat{f}_k(\boldsymbol{x}) - f(\boldsymbol{x})| \leq c \left(\frac{c_{\zeta n}}{\sqrt{k}} + \left(\frac{k}{n} \right)^{\alpha/(2\alpha+p)} \right),$$

where $c_{\zeta n} = 16 \log(2/\zeta) \sqrt{p \log n}$.

The analysis of the density peaks clustering algorithm to detect modes of the underlying probability density function, we begin with the definition of modes of the underlying density.

Definition 4. *The modes of f is the set $\mathcal{M} = \{\boldsymbol{x} : \exists r > 0, \forall \boldsymbol{x}' \in B(\boldsymbol{x}, r),\ f(\boldsymbol{x}') < f(\boldsymbol{x})\}$.*

It is also assumed that the modes have negative definite Hessian.

Assumption 3 (Negative Definite Hessian)**.** *We denote the gradient of f by ∇f and the Hessian by $\nabla^2 f$. $\nabla^2 f(\boldsymbol{x})$ is negative definite for every $\boldsymbol{x} \in \mathcal{M}$.*

An implication of Assumption 3 is that for $\boldsymbol{x} \in \mathcal{M}$, f is well-approximated in a neighborhood by a quadratic function. This is summarized in the following lemma.

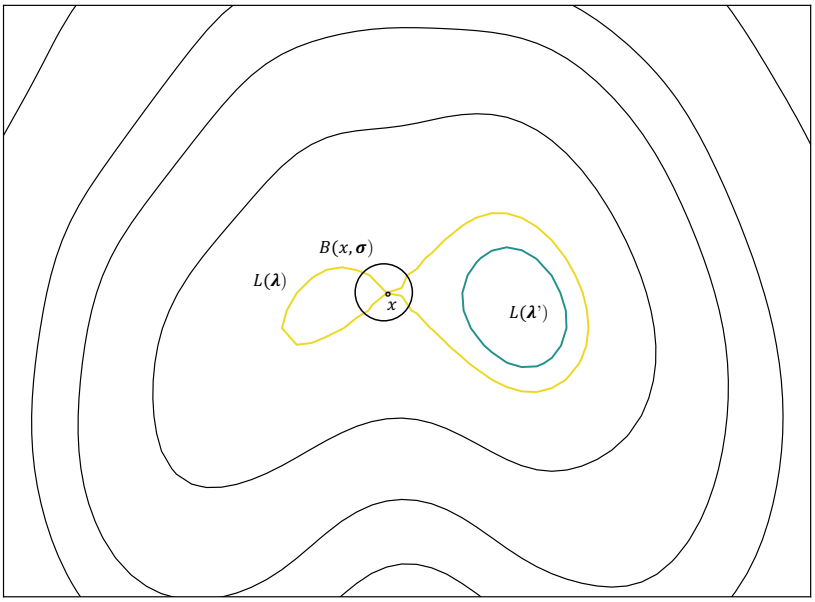


Figure 3.2: Density contour plot of a density that satisfies the requirements of Assumption 2, but not Assumption 4. Here, the ϵ-interior of the level set $L(\lambda)$ can be shown to be contained in $L(\lambda')$ for some $\epsilon > 0$, thus satisfying Assumption 2. However, the level set $L(\lambda)$ becomes arbitrarily thin at the point $\boldsymbol{x} \in \partial L(\lambda)$, contravening the requirements of Assumption 4.

Lemma 2 (Lemma 5 of Dasgupta and Kpotufe (2014))**.** *Let f satisfy Assumptions 1-3. There exists $r_{\mathcal{M}}, \hat{c}, \check{c} > 0$ such that the following holds for all $\boldsymbol{x} \in \mathcal{M}$ simultaneously.*

$$\check{c} \cdot \|\boldsymbol{x} - \boldsymbol{x}'\|^2 \leq f(\boldsymbol{x}) - f(\boldsymbol{x}') \leq \hat{c} \cdot \|\boldsymbol{x} - \boldsymbol{x}'\|^2,$$

for all $\boldsymbol{x}' \in A_{\boldsymbol{x}}$, where $A_{\boldsymbol{x}}$ is a connected component of the level set $L(\lambda)$, where $\lambda = \inf_{\boldsymbol{x}' \in B(\boldsymbol{x}, r_{\mathcal{M}})} f(\boldsymbol{x}')$ which contains the mode $\boldsymbol{x}$ but does not intersect with other modes in $\mathcal{M}$.

The next assumption is required to ensure that the level sets of f are not arbitrarily thin as long as we are a sufficient distance from the modes.

Assumption 4 (Level Set Regularity)**.** *For each $\sigma, r > 0$, there exists $\eta > 0$ such that the following holds for all connected components A of the level set $L(\lambda)$ with $\lambda > 0$ and $A \nsubseteq \cup_{\boldsymbol{x} \in \mathcal{M}} B(\boldsymbol{x}, r)$. If $\boldsymbol{x}$ lies on the boundary of A (i.e, $\boldsymbol{x} \in \partial A$), then $Vol(B(\boldsymbol{x}, \sigma) \cap A) > \eta$, where $Vol(\cdot)$ is volume with respect to the uniform measure on $\mathbb{R}^p$.*

An example contrasting the requirements imposed on the level sets by Assumption 2 and Assumption 4 is given in Figure 3.2.

We now provide the mode estimation results for the k-NN density estimator given in Dasgupta and Kpotufe (2014). The estimator used to predict the modes is the simple direct estimator, selecting the maximizer of $\hat{f}_k$ out of the finite sample $\boldsymbol{X}$.

Lemma 3 (Theorem 2 of Dasgupta and Kpotufe (2014))**.** *Supposed the Assumptions 1-4 hold. Let $\bar{r} > 0$. There exists $N_{\boldsymbol{x},\zeta}$ such that the following holds for $n \geq N_{\boldsymbol{x},\zeta}$. Let $r_{\mathcal{M}}, \hat{c}, \check{c}$ be as in Lemma 2. Suppose k satisfies*

$$\left(\frac{24 c_{\zeta n} f(\boldsymbol{x})}{\check{c} r_{\mathcal{M}}^2}\right)^2 \leq k \leq \left(\frac{1}{2}\sqrt{\frac{c_{\zeta n}}{\hat{c}}}\right)^{4p/(4+p)} f(\boldsymbol{x})^{(2p+4)/(4+p)} \left(\frac{\upsilon_p \cdot n}{4}\right)^{4/(4+p)}.$$

Suppose $\boldsymbol{x}^* \in \mathcal{M}$ and $\boldsymbol{x}^*$ is the unique maximizer of f on $B(\boldsymbol{x}^*, \bar{r})$. Then letting $\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in B(\boldsymbol{x}^*, \bar{r}) \cap \boldsymbol{X}} \hat{f}_k(\boldsymbol{x})$, we have with probability at least $1 - 2\zeta$

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\| \leq \sqrt{\frac{24 c_{\zeta n}}{\check{c}} f(\boldsymbol{x}^*)} \cdot \frac{1}{k^{1/4}}.$$

This result guarantees, with high probability, that every true mode of the underlying density are estimated consistently with the observed instances with the highest values of the k-NN density estimator. The conditions on k, while opaque, allow for a wide range of setting of k. For Hölder assumptions on f, the conditions are seen to allow for the range

$$c_1 \cdot \log(n) \leq k \leq c_2 \cdot n^{4/(4+p)},$$

where c_1, c_2 are constants that depend on $f(\boldsymbol{x})$, but are independent of k and n. The choice $k = \Theta(\log^2(n))$ is seen to be always admissible for n sufficiently large. Assuming k is set in the range given in Lemma 3, the magnitude of the $k^{1/4}$ term in the denominator dominates the other terms in the bound, notably the $\sqrt{\log n}$ that arises in the constant $c_{\zeta n}$, thus shrinking the size of the bound and guaranteeing the quality of the mode estimate.

Following Jiang (2017b), we now define a stronger notion of a mode that allows clearer analysis of the peak-finding criterion.

Definition 5. *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.*

If $\boldsymbol{x}^*$ is an $(r, \theta, \nu)^+$-mode, the definition states that $f(\boldsymbol{x}^*) \geq \nu + \theta$. Also, $\boldsymbol{x}$ is a maximizer of f in a larger ball of radius r by at least θ for points outside the region of quadratic decay $B(\boldsymbol{x}^*, r_{\mathcal{M}})$, as defined in Lemma 2, i.e., $f(\boldsymbol{x}^*) \geq f(\boldsymbol{x}') - \theta$, $\forall \boldsymbol{x}' \in B(\boldsymbol{x}^*, r) \backslash B(\boldsymbol{x}^*, r_{\mathcal{M}})$.

Recall that the density peaks clustering algorithm requires setting of thresholds for the values of the density estimate $\hat{f}_k(\boldsymbol{x})$ and the distance to a nearest neighbor of higher estimated density, $\omega(\boldsymbol{x})$. Taking the thresholds as τ and l for the density and distance values respectively, we show that $\widehat{\mathcal{M}}$ contains unique and consistent estimates of the $(\tau + \epsilon, \theta, l)^+$-modes of f, for $\theta, \epsilon > 0$.

Theorem 1 (Adapted from Theorem 2 of Jiang (2017b))**.** *Let $\boldsymbol{x}^* \in \mathcal{M}^+_{\tau+\epsilon,\theta,l}$ be a $(\tau + \epsilon, \theta, l)^+$-mode of f, where $\theta, \epsilon > 0$. Let k be chosen in agreement with Lemma 3. Then there exists $C > 0$ depending on f such that the following holds for n sufficiently large with probability at least $1 - \zeta$. For each $\boldsymbol{x}^* \in \mathcal{M}^+_{\tau+\epsilon,\theta,l} \backslash \mathcal{M}^-_{\tau-\epsilon,\theta,l}$, there exists a unique $\hat{\boldsymbol{x}} \in \widehat{\mathcal{M}}$ such that*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\| \leq \sqrt{\frac{24 c_{\zeta n}}{\check{c}} f(\boldsymbol{x}^*)} \cdot \frac{1}{k^{1/4}}.$$

Proof. Following the same route as Jiang (2017b), suppose that $\boldsymbol{x}^* \in \mathcal{M}^+_{\tau+\epsilon,\theta,l} \backslash \mathcal{M}^-_{\tau-\epsilon,\theta,l}$. Let $\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in B(\boldsymbol{x}^*, \tau) \cap \boldsymbol{X}} \hat{f}_k(\boldsymbol{x})$. First, we show that $\hat{\boldsymbol{x}} \in \widehat{\mathcal{M}}$.

From Lemma 3, we have that

$$\|\boldsymbol{x}^* - \hat{\boldsymbol{x}}\| \leq \sqrt{\frac{24 c_{\zeta n}}{\check{c}} f(\boldsymbol{x}^*)} \cdot \frac{1}{k^{1/4}}.$$

Let $\tilde{r} = \sqrt{\frac{24c_{\zeta n}}{\check{c}} f(\boldsymbol{x}^*)} \cdot \frac{1}{k^{1/4}}$. It remains to show that $\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in B(\hat{\boldsymbol{x}}, \tau) \cap \boldsymbol{X}} \hat{f}_k(\boldsymbol{x})$. We have $B(\hat{\boldsymbol{x}}, \tau) \subseteq B(\boldsymbol{x}^*, \tau + \tilde{r})$. Choose n sufficiently large such that simultaneously (i) $\tilde{r} < \epsilon$; (ii) by Lemma 1 $\sup_{\boldsymbol{x} \in \mathbb{R}^p} |\hat{f}_k(\boldsymbol{x}) - f(\boldsymbol{x})| < \theta/4$; and (iii) $\tilde{r}^2 < \theta/(4\hat{c})$. Now

$$
\begin{aligned}
\sup_{\boldsymbol{x} \in B(\boldsymbol{x}^*, \tau+\tilde{r}) \setminus B(\boldsymbol{x}^*, \tau)} \hat{f}_k(\boldsymbol{x}) &\leq \sup_{\boldsymbol{x} \in B(\boldsymbol{x}^*, \tau+\tilde{r}) \setminus B(\boldsymbol{x}^*, \tau)} f(\boldsymbol{x}) + \theta/4 && \text{(By (ii))} \\
&\leq f(\boldsymbol{x}^*) - 3\theta/4 && (\text{As} \in \mathcal{M}^+_{\tau+\epsilon,\theta,l} \setminus \mathcal{M}^-_{\tau-\epsilon,\theta,l}) \\
&\leq f(\hat{\boldsymbol{x}}) + \hat{c}\tilde{r}^2 - 3\theta/4 && \text{(Lemma 3)} \\
&< f(\hat{\boldsymbol{x}}) - \theta/2 && \text{(By (iii))} \\
&< \hat{f}_k(\hat{\boldsymbol{x}}). && \text{(By (ii))}
\end{aligned}
$$

Therefore, we have $\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in B(\hat{\boldsymbol{x}}, \tau) \cap \boldsymbol{X}} \hat{f}_k(\boldsymbol{x})$. Furthermore, by (ii) we have that $\hat{f}_k(x) > l - \theta$. Hence, $\hat{\boldsymbol{x}} \in \widehat{\mathcal{M}}$.

Now we show that it is unique. Suppose that $\hat{\boldsymbol{x}}' \in \widehat{\mathcal{M}}$ such that $\|\hat{\boldsymbol{x}}' - \boldsymbol{x}^*\| \leq \tau/2$. Then, both $\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in B(\hat{\boldsymbol{x}}, \tau) \cap \boldsymbol{X}} \hat{f}_k(\boldsymbol{x})$ and $\hat{\boldsymbol{x}}' = \arg\max_{\boldsymbol{x} \in B(\hat{\boldsymbol{x}}', \tau) \cap \boldsymbol{X}} \hat{f}_k(\boldsymbol{x})$. However, choosing n sufficiently large such that $\tilde{r} < \tau/2$ yields $\hat{\boldsymbol{x}} \in B(\hat{\boldsymbol{x}}', \tau)$, implying $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}'$. □

Theorem 1 proves that the density peaks clustering algorithm 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. The guarantee above states that for every true mode of the underlying density, there exists a consistent estimator in the set of estimated modes. As such, there is an injection between the set of true modes and the set of estimated modes. Later, in Chapter 4, a procedure will be introduced that allows for theoretical guarantees regarding a bijective estimator of the modes of the population probability distribution.

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 Jiang et al. (2018) can be applied directly to density peaks clustering. We provide the

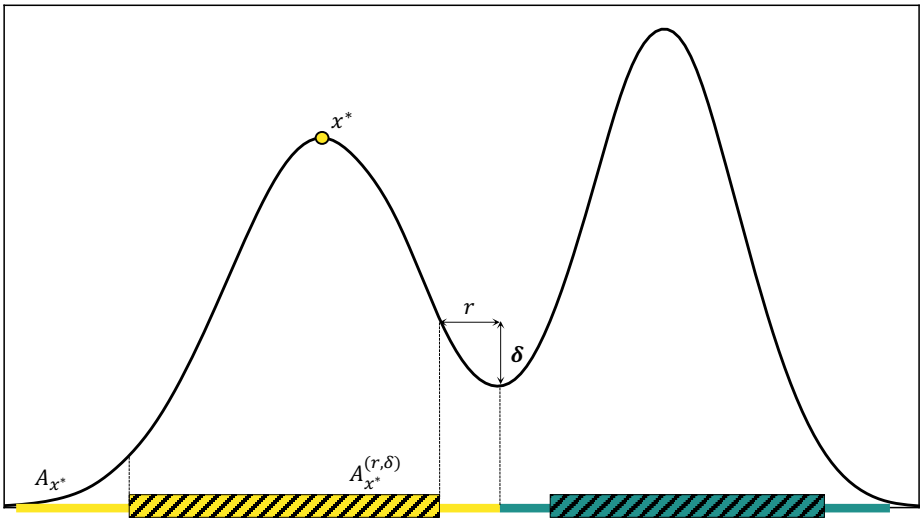


Figure 3.3: An illustrative example of the (r, δ)-interior of an attraction region $\mathcal{A}_{\boldsymbol{x}^*}$, denoted $\mathcal{A}_{\boldsymbol{x}^*}^{(r,\delta)}$, associated with a mode $\boldsymbol{x}^*$.

relevant results below.

First we formalize the definition for the attraction regions of a mode. The concept is the same as introduced for mean shift in Section 2.2.2. The attraction region of a particular mode are all points that flow towards the mode following the direction of the gradient of the underlying density.

Definition 6. *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* $\lim_{t\to\infty} \nu_{\boldsymbol{x}}(t) = \boldsymbol{x}^*$.

We require the additional regularity assumption that the modes lie in the interior of the attraction regions.

Assumption 5. *There exists* $r_0 > 0$ *such that* $B(\boldsymbol{x}^*, r_0) \subset \mathcal{A}_{\boldsymbol{x}^*}$ *for every* $\boldsymbol{x}^* \in \mathcal{M}$.

It is shown that density peaks clustering 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 7. *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*

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

The definition captures the notion that 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. The parameter r controls the distance about the interior of the attraction region and the parameter δ captures the magnitude of the density decrease. An illustrative example is given in Figure 3.3.

The main result states that, as long as the modes are sufficiently well-estimated, the assignment method of density peaks clustering will correctly cluster the (r, δ)-interiors of the attraction regions with high probability.

Theorem 2 (Adapted from Theorem 2 of Jiang et al. (2018))**.** *Suppose the Assumptions 1-5 hold. Let* $0 < r < r_0$*, where* r_0 *is as defined in Assumption 5, and* $\delta, \zeta > 0$*. Suppose that* $k = k(n)$ *is chosen such that* $\log^2 n/k \to 0$ *and* $n^{4/(4+p)}/k \to 0$*. Suppose that* $\boldsymbol{x}^* \in \mathcal{M}$ *is a mode of the underlying density and* $\hat{\boldsymbol{x}}$ *is a mode estimate returned by Algorithm 1 such that*

$$\|\hat{\boldsymbol{x}} - \boldsymbol{x}^*\| \leq \frac{r}{4}.$$

Then, for $n = n(f, \delta, \zeta, r)$ *sufficiently large, depending on* f*,* δ*,* ζ *and* r*, the following holds with probability at least* $1-2\zeta$ *uniformly in* $\boldsymbol{x} \in \mathcal{A}_{\boldsymbol{x}^*}^{(r,\delta)} \cap \boldsymbol{X}$*: density peaks clustering clusters* $\boldsymbol{x}$ *to the cluster corresponding to* $\boldsymbol{x}^*$*.*

Proof. To prove this theorem, we require a uniform concentration bound on balls intersected with density-level sets of f. The following result states that if such a set has enough probability mass, then it will contain a sample point with high probability.

Lemma 4 (Lemma 3 of Jiang et al. (2018))**.** *Let* $\mathcal{E} = \{B(\boldsymbol{x}, s) \cap L(\lambda) : \boldsymbol{x} \in \mathbb{R}^p, s >$

$0, \lambda > 0\}$. *Then with probability at least* $1 - \zeta$ *uniformly for all* $E \in \mathcal{E}$

$$\mathcal{F}(E) \geq c_{\zeta n} \frac{\sqrt{p \log n}}{n} \implies E \cap \boldsymbol{X} \neq \varnothing.$$

Now suppose that $\boldsymbol{x} \in \mathcal{A}_{\boldsymbol{x}^*}^{(r,\delta)} \cap \boldsymbol{X}$. Density peaks clustering gives a directed path $\boldsymbol{x} \to \boldsymbol{x}_1 \to \boldsymbol{x}_2 \to \cdots \to \boldsymbol{x}_T$, where $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{T-1}$ are not modes and $\boldsymbol{x}_T$ is a mode. Suppose also that $\boldsymbol{x}_T \neq \boldsymbol{x}^*$, the mode associated with the attraction region $\mathcal{A}_{\boldsymbol{x}^*}$.

It is shown first that $\|\boldsymbol{x}_i - \boldsymbol{x}_{i+1}\| \leq r/2$ for $i = 1, \ldots, T-1$. By Assumption 4, there exists $\nu > 0$ and $\eta > 0$ such that the following holds for $i = 1, \ldots T-1$:

$$Vol\left(B(\boldsymbol{x}_i, r/2) \cap L(f(\boldsymbol{x}_i) + \nu)\right) \geq \eta.$$

Thus, as the density is lower bounded by λ_0, we have

$$\mathcal{F}\left(B(\boldsymbol{x}_i, r/2) \cap L(f(\boldsymbol{x}_i) + \nu)\right) \geq \eta\lambda_0.$$

Then, using Lemma 4, for n sufficiently large such that $\eta\lambda_0 > c_{\zeta n}\frac{\sqrt{p \log n}}{n}$, with probability at least $1 - \zeta$ there exists a sample point $\boldsymbol{x}_i' \in B(\boldsymbol{x}_i, r/2) \cap L(f(\boldsymbol{x}_i) + \nu)$ for $i = 1, \ldots, T-1$.

Next, if n is chosen sufficiently large such that, by Lemma 1, we have with probability at least $1 - \zeta$ that

$$\sup_{\boldsymbol{x} \in \mathcal{X}} |\hat{f}_k(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \min\{\nu, \delta\}/3.$$

As a result, we have

$$\begin{aligned}
\hat{f}_k(\boldsymbol{x}_i') &\geq f(\boldsymbol{x}_i') - \nu/3 \\
&\geq f(\boldsymbol{x}_i) + 2\nu/3 \\
&\geq \hat{f}_k(\boldsymbol{x}_i) + \nu/3 \\
&> \hat{f}_k(\boldsymbol{x}_i).
\end{aligned}$$

Furthermore, $\|\boldsymbol{x}_i - \boldsymbol{x}_i'\| \leq r/2$ and $\boldsymbol{x}_i' \in \boldsymbol{X}$. Thus, it follows that $\|\boldsymbol{x}_i - \boldsymbol{x}_{i+1}\| \leq r/2$ for $i = 1, \ldots, T-1$.

Next, let $\pi : [0,1] \to \mathbb{R}^p$ be the piecewise linear path defined by $\pi(j/L) = \boldsymbol{x}_j$ for $j = 1, \ldots, L$. Let $t_2 = \min\{t \in [0,1] : \pi(t) \in \partial\mathcal{A}_{\boldsymbol{x}^*}\}$. Then, by the definition of the (r,δ)-interior of the attraction region $\mathcal{A}_{\boldsymbol{x}^*}^{(r,\delta)}$, there exists $0 \leq t_1 < t_2$ such that $\boldsymbol{x} = \pi(t_1)$ and $\boldsymbol{y} = \pi(t_2)$ satisfies $\boldsymbol{y} \in \partial\mathcal{A}_{\boldsymbol{x}^*}$ and

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

As such, one can find indices $l, m \in \{1, \ldots, T-1\}$ such that $l \leq m$, $\|\boldsymbol{x}_l - \boldsymbol{x}_m\| \leq r$ and $\|\boldsymbol{x}_m - \boldsymbol{y}\| \leq r$. Thus, $f(\boldsymbol{x}_l) \geq f(\boldsymbol{x}_m) + \delta$, but $\hat{f}_k(\boldsymbol{x}_l) \leq \hat{f}_k(\boldsymbol{x}_m)$. However, we have

$$\begin{aligned}
\hat{f}_k(\boldsymbol{x}_l) &\geq f(\boldsymbol{x}_l) - \nu/3 \\
&\geq f(\boldsymbol{x}_m) + 2\nu/3 \\
&\geq \hat{f}_k(\boldsymbol{x}_m) + \nu/3 \\
&> \hat{f}_k(\boldsymbol{x}_m),
\end{aligned}$$

which is a contradiction as required. □

The results in this section are naturally adapted if a KDE is used in place of the k-NN estimator. Taking the KDE as

$$\hat{f}_h(\boldsymbol{x}) = \frac{1}{n \cdot h^p} \sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right),$$

where $K(\cdot)$ is the a spherically symmetric, non-increasing function that experiences exponential decay, such as the kernels detailed in Table 2.2. In place of Lemma 1, the following result on the uniform KDE bound is provided from Theorem 2 of Jiang (2017c).

Lemma 5. *Suppose f satisfies Assumptions 1 and 2. Then there exists a constant*

$c' = c'(f, K)$, depending on f and K, such that the following holds with probability at least $1 - 1/n$ uniformly in $h > (\log n/n)^{1/p}$.

$$\sup_{\boldsymbol{x} \in \mathbb{R}^p} |\hat{f}_h(\boldsymbol{x}) - f(\boldsymbol{x})| \leq c' \cdot \left(h^\alpha + \sqrt{\frac{\log n}{n \cdot h^p}} \right).$$

3.4 Illustrative Analysis

The theoretical analysis of Section 3.3 provides guarantees on the ability of the density peaks clustering algorithm to (1) consistently recover modes of the underlying density and (2) correctly assign instances to the cluster of their associated mode. Such results are based on the assumption of a sample size large enough that the error of density estimator can be bounded.

In this section, we provide an analysis of the density peaks clustering algorithm when applied to datasets with a small sample size. We consider five illustrative datasets from the scikit-learn clustering demonstration (Pedregosa et al., 2011). Taken together, the datasets provide an understanding of the density peaks clustering algorithm and the type of clusters it returns. The datasets are detailed as follows:

- *Anisotropic Gaussian*: This dataset consists of three well-separated Gaussian components that are anisotropically distributed, i.e., the covariance matrices of the components are not axis-aligned. The components have the same prior probability and common covariance matrices.

- *Unequal Variance Gaussian*: This dataset consists of three isotropically distributed Gaussian components with a small amount of overlap between components. Each component has the same prior probability, with three different variance levels.

- *Noisy Circles*: This dataset consists of two clusters. The samples form concentric circles with noise about each, but no overlap. The prior probability of both clusters is equal, with the inner cluster having higher density resulting from the

smaller radius.

- *Noisy Moons*: This dataset consists of two clusters. The samples form two crescents with noise about each, but no overlap. The prior probability of an instance belonging to the top crescent is $\pi_1 = 0.6$ and for the bottom crescent is $\pi_2 = 0.4$.

- *No Cluster Structure*: This dataset consists of points generated according to a uniform distribution within the range $[0, 1]$ for each axis. There is no geometrically intuitive cluster structure present in the data.

For each dataset, a sample of $n = 1500$ instances were drawn. We provide the results of the density peaks clustering method for both the k-NN estimator and the KDE. We assess the performance of the k-NN estimator for $k = 10$ ($\approxeq \log(n)$) and $k = 40$ ($\approx \sqrt{n}$). The KDE uses a Gaussian kernel and the bandwidth of the KDE is set to a certain proportion of the average sample variance of the data in each direction. This bandwidth was chosen as it naturally adapts to the scale of the data. The proportions assessed are $1/10$ and $1/25$. For the k-NN estimator, higher values of k will lead to a smoother density estimate. For the KDE, larger bandwidths also lead to smoother density estimates.

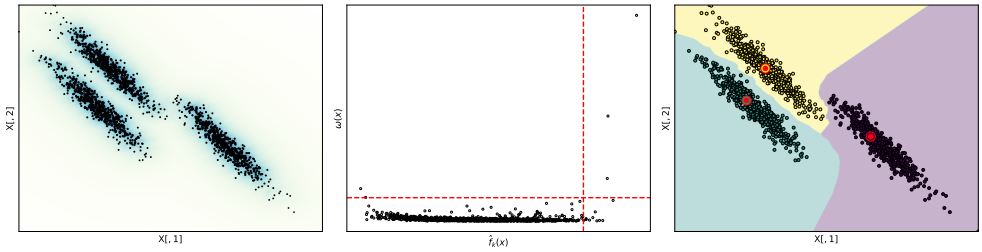
In Figures 3.4-3.7 we show the density estimate, the decision plot, the clustering of the data and the attraction regions of each of the modes for each dataset. For each dataset, presented in the left figure is the density estimator, with darker regions indicating higher density. The center figures are the decision plots with the thresholds τ and l used to select modes highlighted in red. The thresholds were set to return approximately the correct number of clusters for each dataset. Instances in the top right quadrant formed by the thresholds are taken as the mode estimates. The right figures show the clusterings returned by the density peaks clustering method with the given density estimate and the chosen modes. The estimated modes are highlighted in red. The shaded regions correspond to the attraction regions of each mode estimate.

Assessing first the density estimators considered for use in Algorithm 1. For all

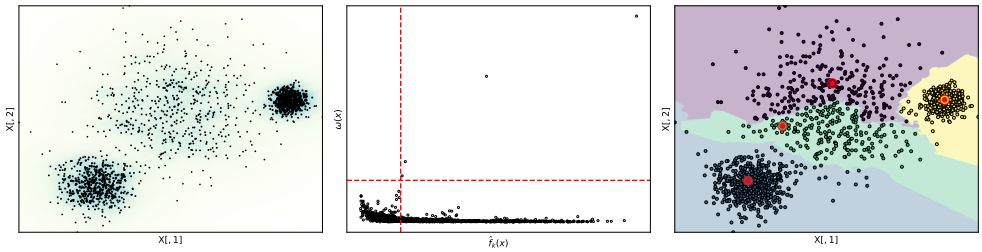
datasets, bar the No Structure dataset, the quality of the density estimation appear consisent across estimators and the various input parameters considered. The population density used to generate the data is identifiable in each of the four cases. For the No Structure dataset, the density estimators providing noisier estimates, namely the k-NN estimator with $k = 10$ shown in Figure 3.4 and the KDE with the proportion set to 1/25 shown in Figure 3.6, appear to erroneously detect high and low density regions in the data. The density is seen to be significantly higher in neighborhoods of the observed samples. This is not the case for the alternate parameterizations, where no clear population structure is evident.

The second feature of the density peaks clustering algorithm analyzed is the decision plot, provided to enable the estimation of the modes from the dataset. The decision plots are seen in the middle panels of each figure and the estimated modes are highlighted in red on the right panels of each figure. The decision plots have mixed performance. For the Anisotropic Gaussian dataset, the three modes are easily seen in the decision plot for each density estimator used. These modes translate to reasonable archetypes of the clusters in each case. For the Unequal Variance Gaussian dataset, the decision plot correctly proposes modes from the high density clusters in the data. These estimates are clear from the decision plot. Furthermore, estimates from the low density cluster can be chosen provided the thresholds are set carefully. These estimates are more easily seen for the k-NN density estimator, particularly when $k = 40$, than for the KDE. While the situation is less clear than for the Anisotropic Gaussian dataset, the decision plot is still said to have performed well for this dataset.

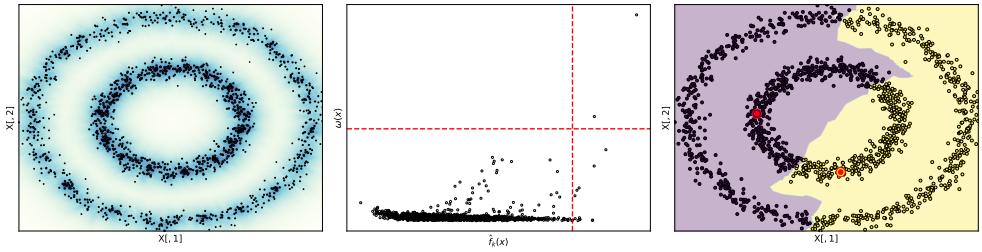
The remaining datasets each contain areas of relatively uniform density; for the Noisy Circles dataset, the clusters are generated as circles of uniform density, about which data is generated with Gaussian noise; for the Noisy Moons dataset, the clusters are similarly generated as crescents of uniform density, about which data is generated with Gaussian noise; and the population density of the No Cluster Structure dataset is entirely uniform. This poses challenges for the density peaks clustering method. The key issue is that, for regions of relative uniformity in the density, the small sample



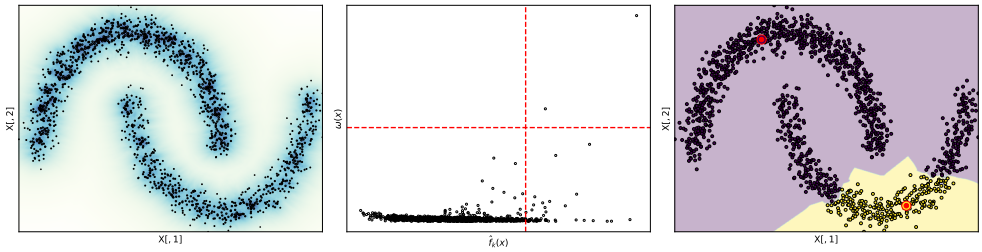
(a) Anisotropic Gaussian Dataset



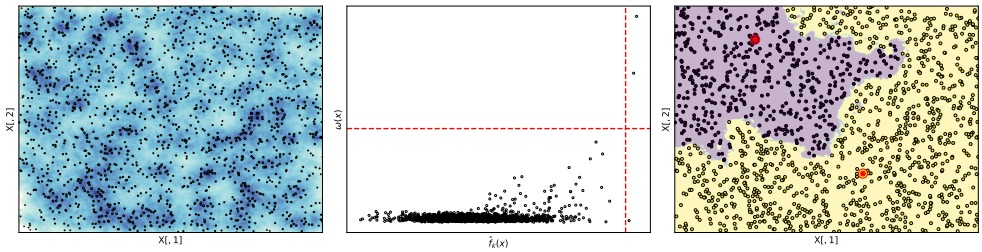
(b) Unequal Variance Gaussian Dataset



(c) Noisy Circles Dataset

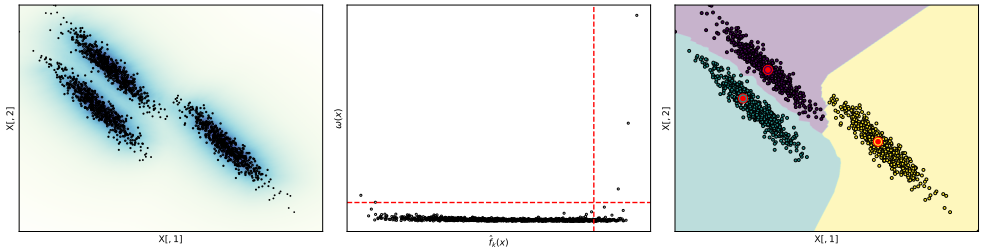


(d) Noisy Moons Dataset

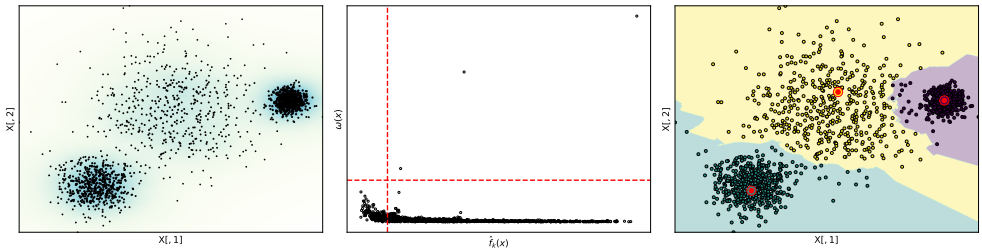


(e) No Cluster Structure Dataset

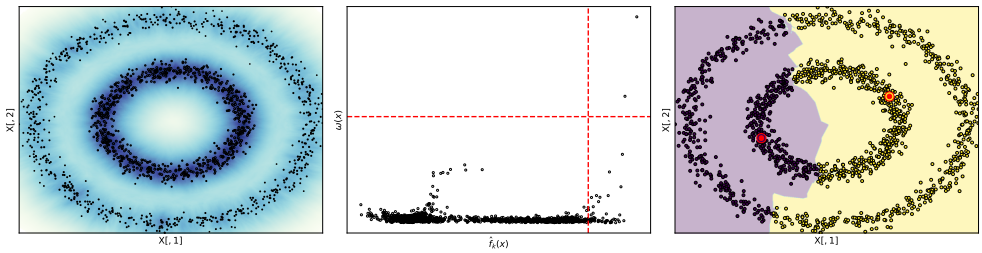
Figure 3.4: Density peaks clustering of illustrative datasets. The k-NN estimator is used here with $k = 10$.



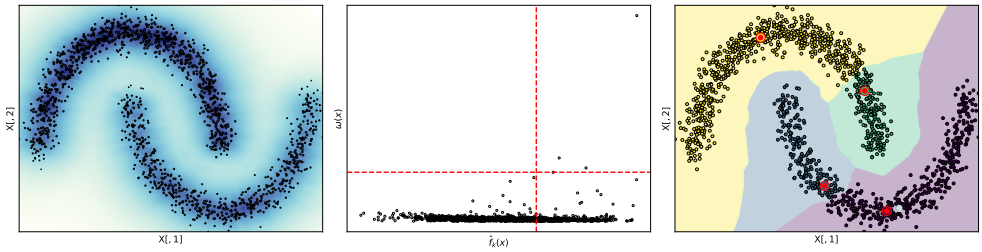
(a) Anisotropic Gaussian Dataset



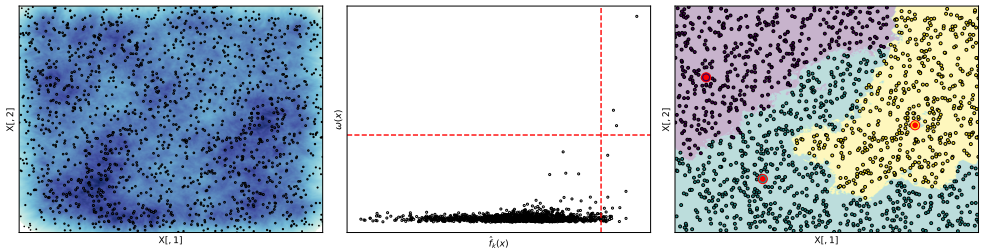
(b) Unequal Variance Gaussian Dataset



(c) Noisy Circles Dataset

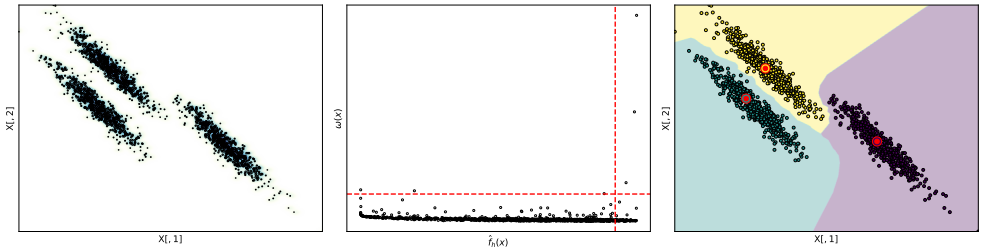


(d) Noisy Moons Dataset

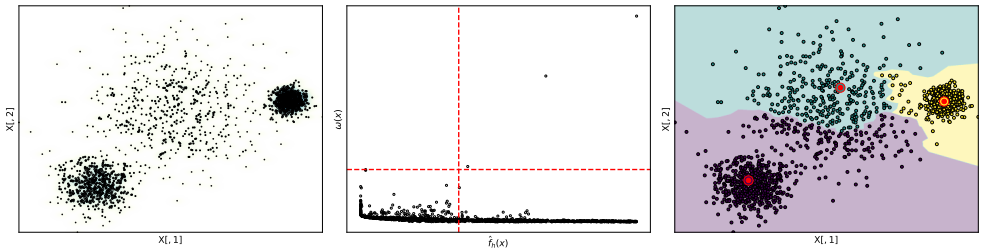


(e) No Cluster Structure Dataset

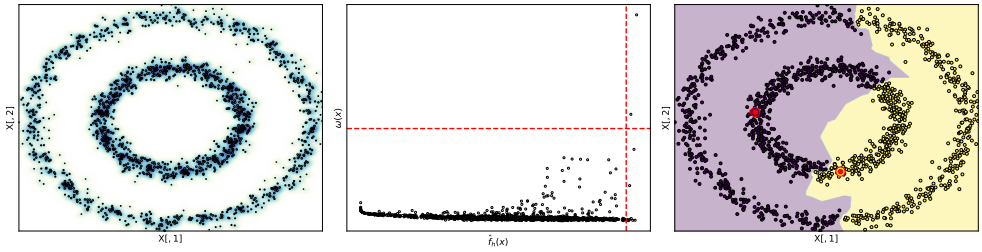
Figure 3.5: Density peaks clustering of illustrative datasets. The k-NN density estimator is used here with $k = 40$.



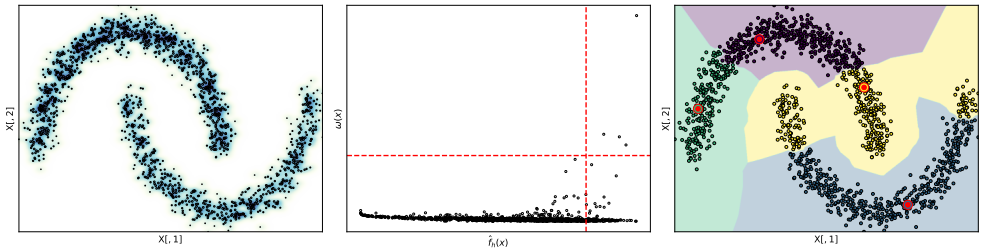
(a) Anisotropic Gaussian Dataset



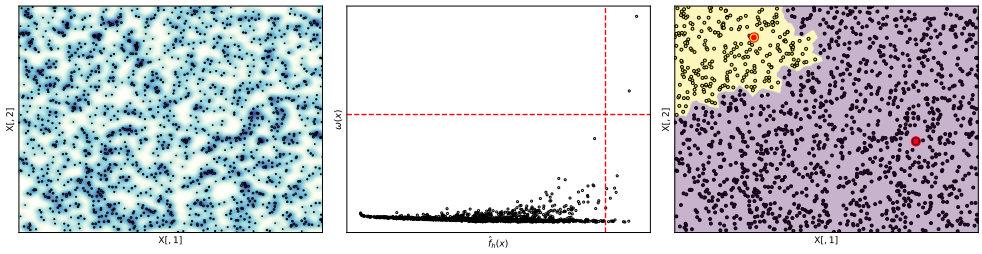
(b) Unequal Variance Gaussian Dataset



(c) Noisy Circles Dataset

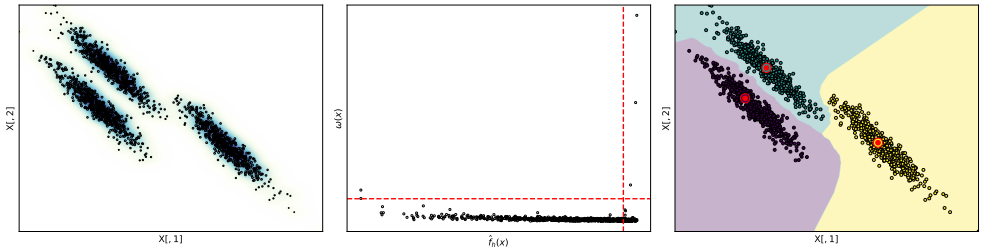


(d) Noisy Moons Dataset

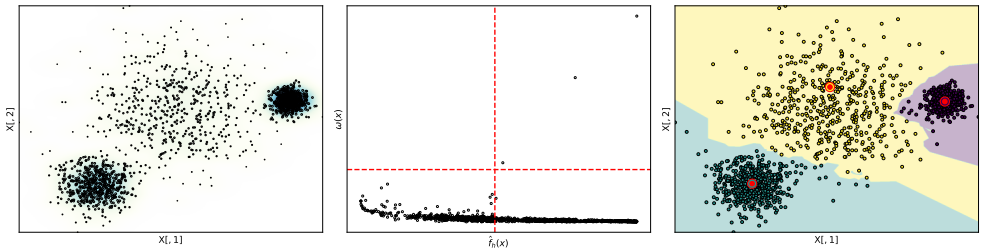


(e) No Cluster Structure Dataset

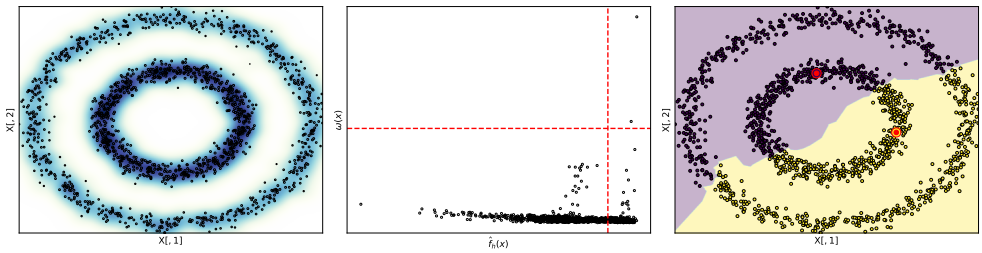
Figure 3.6: Density peaks clustering of illustrative datasets. The KDE is used here with the bandwidth h set to 1/25 of the average sample variance in each direction.



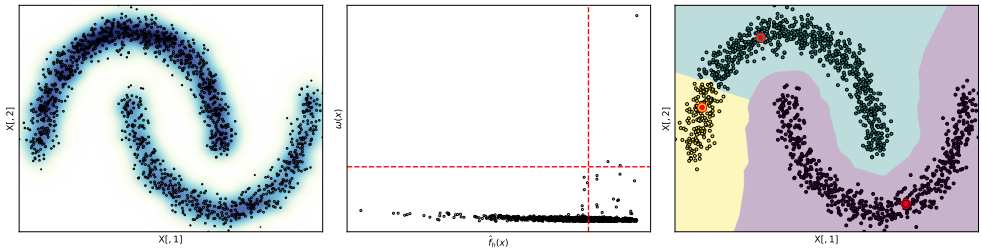
(a) Anisotropic Gaussian Dataset



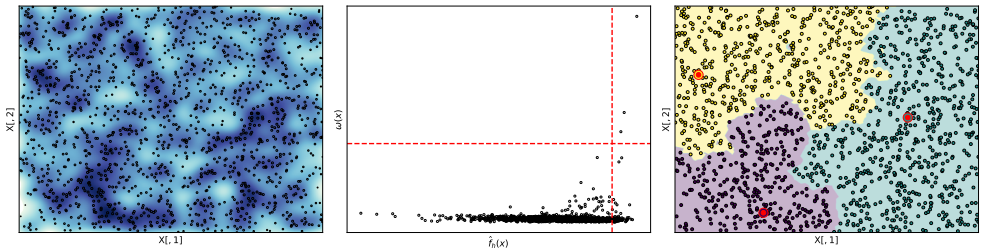
(b) Unequal Variance Gaussian Dataset



(c) Noisy Circles Dataset



(d) Noisy Moons Dataset



(e) No Cluster Structure Dataset

Figure 3.7: Density peaks clustering of illustrative datasets. The KDE is used here with the bandwidth h set to 1/10 of the average sample variance in each direction.

size leads to noisy density estimates. Such noise in the density estimate $\hat{f}$ leads to instances $\boldsymbol{x} = \arg\max_{\boldsymbol{x}' \in B(\boldsymbol{x}, r)} \hat{f}(\boldsymbol{x})$, some $r > 0$ that are locally maximal for $\hat{f}$ being selected as modes, when they are not maximizers of an r-radius ball $B(\boldsymbol{x}, r)$ for the true density f.

For the Noisy Circles dataset, there are multiple modes are suggested from the high density cluster for every density estimator used. While a set of points from the low density cluster can be seen extended from the main group of points in the bottom left quadrant of the decision graph for each density estimator, it is not possible to specify thresholds to recover one mode from each cluster. The user must instead balance the risk of oversegmentation of the high density cluster against the risk of not capturing any representatives from the lower density cluster.

These issues are also manifest in the decision plot for the Noisy Moons dataset. There, modes are proposed for each cluster, but it is not obvious from the decision plot that two modes should be selected. Instead, there is one mode clearly visible in the decision plots, i.e., the instance of maximal density in the high density cluster and a group of other instances that could credibly be selected as modes by a user. This issue is present for each density estimator assessed, indicating that the deficiency lies in the decision plot methodology and can not be improved using alternative parameter tunings.

The same issues are again observed for the No Cluster Structure dataset. Noise in the density estimate leads to erroneous modes being selected from the decision plot. This issue is not ameliorated by using smoother density estimates. The issue lies in the formulation of the density peaks clustering method and the notion of mode estimates used therein.

The method of selecting mode estimates from the decision plot is seen to perform well when the density of the cluster is high and concentrated near the mode and decays as the distance from the mode increases, such as for the Gaussian components of the Anisotropic Gaussian and Unequal Variance Gaussian datasets. The performance is significantly degraded when the density is relatively uniform for broad regions of the

data. This is seen clearly for the Noisy Circles and Noisy Moons datasets, where the true number of clusters is not obvious from the decision graph, and for the No Cluster Structure dataset, where erroneous modes are suggested by the decision plot. The challenges faced trying to detect clusters without clear and distinct modes reflect a limitation of the theoretical analysis presented in Section 3.3. There, in Assumption 3, we require that the clusters contain individual modes and that the density decays approximately quadratically in a region about the mode. This assumption is not satisfied for any of the Noisy Moons, Noisy Circles and No Cluster Structure datasets.

Finally, the assignment method of density peaks clustering is assessed. In the right panels of each figure the allocation of observed instances to clusters is shown. The associated attraction regions for each mode are represented using the background colours. The assignment strategy is shown to perform well for the Anisotropic Gaussian dataset and, accepting the issues with mode estimation, for the Unequal Variance Gaussian and No Cluster Structure datasets also. The allocation of instances to clusters for the Noisy Circles and Noisy Moons datasets is problematic. There, the allocation runs contrary to geometric intuition about the clusters. In both cases the allocation assigns instances to clusters across areas of very low density in the dataset. This is particularly clear for the Noisy Moons dataset, when the k-NN density estimator with $k = 10$ is used. There, despite correct selection of the modes for each cluster, the allocation mechanism incorrectly assigns nearly all of the lower density cluster to the high density cluster. Such instances are assigned across regions of very low density, as confirmed by inspecting the density plot in the left figure. This demonstrates that, while the sample-based analogy to gradient ascent of the density used to allocate instances is consistent for large sample sizes, it can be shown to fail for data of typical size.

In sum, the density peaks clustering framework is capable at detecting high quality mode estimates and clusterings for datasets containing clusters with clear point modes about which the density decays, such as Gaussian components. This performance is consistent across different density estimators and parametrizations. The framework struggles when the high density regions of the data are relatively uniform. In this

case, both the mode selection method and assignment strategy of the density peaks clustering algorithm are shown to be susceptible to errors caused by noise in the density estimate.

3.5 Conclusion

In this chapter, we introduced the density peaks clustering algorithm, as formulated for the remainder of this work. The k-NN density estimator, the peak-finding criterion and the decision plots are formalized in Section 3.2. The algorithm used to produce clusterings is detailed in Algorithm 1. In Section 3.3, the theoretical guarantees on the quality of the modes estimated using the peak-finding criterion and the assignment of instances to their respective clusters were provided. Such results, the first in the context of density peaks clustering, show that the method of Algorithm 1 fits the conception of non-parametric density-based clustering methods as described in Section 2.2.1. Finally, the density peaks clustering method was implemented for five illustrative datasets. The results demonstrate the ability of the k-NN density estimator to provide high-quality estimates of the underlying density. Mode selection from the decision plot, while intuitive, was shown to be a subjective where reasonable interpretations may greatly differ. Also shown was the sensitivity of the density peaks clustering method to noise in this estimate and issues caused by the allocation mechanism allocating instances across regions of very low density. In the next sections, two novel methods are introduced that retain the intuitive nature of the density peaks clustering method, while allowing for extended theoretical guarantees, improved robustness to fluctuations in the density estimate, and allocation mechanisms that respect the geometry of the clusters.

4 Modal-Set Detection with the Peak-Finding Criterion

4.1 Summary

As discussed previously, the density peaks clustering algorithm detects modes as points with high density and large distance to points of higher density, and hence often fails to adequately represent clusters with areas of relatively uniform density. In this chapter, we develop an improved clustering algorithm, aiming at enhancing the applicability of the peak-finding technique. The improvements are twofold: (1) the algorithm is robust to noise in the density estimate and thus detects clusters at varying densities; (2) the algorithm is competent at deciding the correct number of clusters, even when the number of clusters is very high. Improvements in the clustering performance of the novel algorithm relative to the density peaks clustering method are the result of directing the peak-finding technique to discover modal sets, rather than point modes. We present a theoretical analysis of our approach and experimental results to verify that our algorithm works well in practice and executes efficiently. We demonstrate a potential application of this method for unsupervised face recognition.

4.2 Introduction

A key drawback of the density peaks clustering procedure is that the points with maximal values of the peak-finding criterion are often poor representations of the clusters.

This is most prevalent when the data contain both high- and low-density clusters and is compounded when the true density is relatively uniform over large regions. Noise in the density estimate leads to multiple points from high-density clusters being erroneously selected as centers, while true clusters of lower density are ignored. Previous attempts to improve the detection of low-density modes involve incorporating information about points of lower local density into the density (Chen et al., 2018) and the distance (Li and Tang, 2018). However, these approaches do not remedy the harms caused by variations in the underlying density estimate. To formulate a robust mode-finding procedure, we direct the density peaks clustering procedure to identify locally high density regions of the dataset.

QuickShift++ (Jiang and Kpotufe, 2017; Jiang et al., 2018) improves on both mean shift and quick shift by modelling locally high-density regions using cluster cores. Cluster cores extend the concept of point modes to sets of points of arbitrary shape, size and density level. Cluster cores are parameterized by $\beta \in (0, 1)$, which determines how much the density can fluctuate within a cluster. Using cluster cores instead of point modes reduces the risk of selecting multiple centers from a high-density cluster as they are less sensitive to the chance variation that occurs in the empirical density estimate. As a result, cluster cores better capture locally high-density regions of the support $\mathcal{X}$.

We introduce Density Core Finding (DCF), a novel clustering method that uses the peak-finding criterion to estimate these cluster cores. The peak-finding criterion is computed for each point, and the instance with maximum value is selected as a center. The cluster core containing this center is then found, and all instances belonging to it are removed from consideration as potential centers. The algorithm continues detecting cluster cores until no more remain in the data. The allocation procedure is unchanged from the density peaks clustering method of Algorithm 1, each non-center point is allocated to the same cluster as its nearest neighbor of higher empirical density. The benefits offered by DCF are illustrated in Figure 4.1.

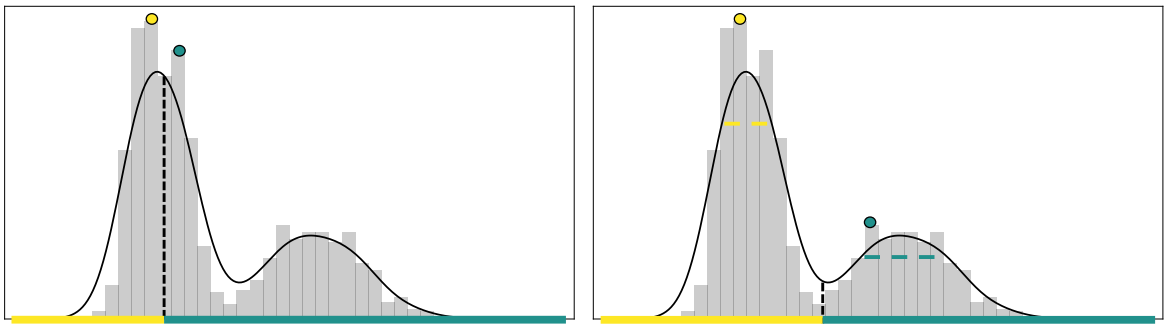


Figure 4.1: An illustration containing two clusters. The black curve represents the underlying density and the grey histogram represents a sample from the density. Left: Density peaks clustering incorrectly selects both centers from the first cluster, as the noise in the density estimate causes the peak-finding method to favor the high-density cluster. Right: Cluster cores, represented by dashed lines, better represent the cluster centers.

In our theoretical analysis, we discuss the necessary conditions for DCF to be guaranteed to recover all of the true modes in the data, as well as certain difficulties imposed by directing the search with the peak-finding criterion. We demonstrate that DCF recovers clusters of the same quality as QuickShift++, while being faster in execution. The improved quality of the clustering results from DCF compared to DPC and QuickShift++ is illustrated using a range of real-world datasets. Finally, we demonstrate the superiority of DCF over competitor methods in a popular application of density-based clustering: large-scale face recognition.

4.3 Related Work

The notion of cluster cores originates in the analysis of the cluster tree, as introduced in Section 2.2.2. Considering again the example given in Figure 2.6, the cluster tree at the level λ_1 shows two branches, representing the two connected components of the level set $L(\lambda_1)$. One of these components is clearly related to the mode $\boldsymbol{x}_1^*$ as $\boldsymbol{x}_1^*$ is the only mode contained within it. This component is said to be the core of the cluster associated with $\boldsymbol{x}_1^*$. The remaining component is not yet associated with only one mode, as it contains both $\boldsymbol{x}_2^*$ and $\boldsymbol{x}_3^*$. The remaining two cores are observed at the level λ_2. A core is termed by Menardi (2015) to be the largest level-set connected

components that contain one mode only. They are detected at the density level at which a branch of the cluster tree splits into more than one child branch.

Jiang and Kpotufe (2017) introduce an alternate notion, termed modal-sets. Modal-sets generalize the concept of a point mode as they are subsets of $\mathcal{X}$ where f is locally maximal. The procedure they introduce estimates the modal-sets by searching the instances in descending order of estimated density. For each instance, an estimate of the level set at the level of its estimated density is found. If the subset of this estimated level set containing the point is disconnected from all previous modal-set estimates, it is accepted as a modal-set. Once all points have been assessed, the points not contained in modal-sets are assigned to the cluster of their nearest modal-set, in terms of the Euclidean distance. The authors provide consistency guarantees on the recovery of true modal-sets in the data. A subsequent work Jiang et al. (2018) synthesizes the notion of modal-sets and cores to define cluster cores. The concept of cluster cores extend modal-sets by allowing the underlying density within a cluster core to vary, and parameterizes the amount of variation by $\beta \in (0, 1)$. This better accounts for fluctuations in the estimated density observed when clustering real data. They introduce QuickShift++ which improves on the procedure of Jiang and Kpotufe (2017) by using a hill-climbing procedure to allocate points to the same cluster as their nearest neighbor of higher local density.

Cluster cores have obvious benefits for the density peaks clustering procedure, resolving the issues caused by clusters of varying density, and exacerbated by noise in the density estimate. Incorporating cluster cores into the density peaks clustering procedure also leads to improvements over the QuickShift++ method. Firstly, the peak-finding criterion provides a more efficient ordering for the search of level sets than empirical density alone. Coupled with the termination criteria to be introduced later, we demonstrate that the number of assessments completed is reduced on average by 98%.

4.4 Our Method

An example of the operation of DCF is illustrated in Figure 4.2. The figure depicts the Noisy Circles dataset containing two clusters, a high-density cluster (inner circle) and a low-density cluster (outer circle). The density peaks clustering method, as seen on the left of the figure, proceeds by searching for the points with maximal values of the peak-finding criterion. This method erroneously selects the multiple centers from the inner cluster. The allocation mechanism incorrectly assigns all points in the outer cluster. For this example, seven points in the inner cluster have larger value of the peak-finding criterion than the maximum value in the outer cluster.

The DCF procedure can be seen on the right of the figure. We first select the instance of maximum density as the first peak. However, DCF proceeds to compute the cluster core associated with this point, the highlighted green points, and remove all elements of the core from consideration as centers. Of those remaining, the point with maximal value of the peak-finding criterion is in the outer cluster. The associated cluster core is visible in yellow. As no edge in the k-NN graph exists between this cluster core and the first cluster core, it is accepted as a valid cluster core. The algorithm's termination procedure is invoked when assessing a third center. The third center is selected as before; however, as the cluster core associated with this point contains all of the instances in the dataset, the algorithm terminates.

4.4.1 Notation and Definitions

The formal definitions and notations for the peak-finding part of this approach are the same as those introduced in Section 3.2. Similarly, Assumptions 1-5 introduced in Section 3.4 are taken as before. Following Jiang et al. (2018), we define the cluster core with respect to a fixed fluctuation parameter β. In what follows, a closed connected set refers to a set which cannot be divided into two disjoint non-empty closed sets.

Definition 8. *Let* $0 < \beta < 1$. *A closed and connected set* $\boldsymbol{M} \subset \mathcal{X}$ *is a cluster core if*

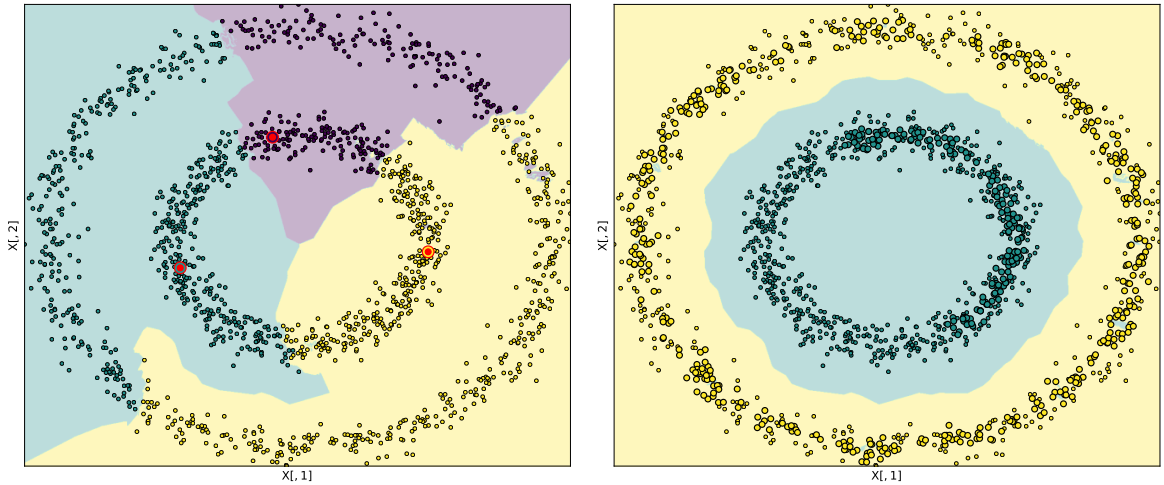


Figure 4.2: The Noisy Circles dataset example containing two clusters, one high-density cluster (the inner circle) and one low-density cluster (the outer circle). Left: DPC incorrectly selects multiple centers from the high-density cluster. Right: DCF uses the darkened points as the respective cluster cores, resolving the error in clustering.

$\boldsymbol{M}$ *is a maximal connected subset of* $\{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) \geq (1-\beta) \cdot \max_{x' \in \boldsymbol{M}} f(\boldsymbol{x}')\}$.

Varying the fluctuation parameter β determines the magnitude of the cluster cores. It is also shown in Jiang et al. (2018) that if $\boldsymbol{M}_1$, $\boldsymbol{M}_2$ are distinct cluster cores of f then $\boldsymbol{M}_1 \cap \boldsymbol{M}_2 = \varnothing$, namely cluster cores do not overlap.

To estimate the cluster cores, we use the level sets of the mutual k-NN graph.

Definition 9. *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 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.*

The level sets of G, namely $G(\lambda)$, are defined as before. It has been shown in Chaudhuri et al. (2014) that the λ-level connected components of f are well approximated by the component sets of $G(\lambda)$. As the value of λ decreases, the component sets of $G(\lambda)$ are hierarchically nested.

Algorithm 2: DCF Algorithm

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

Initialisation: $\widehat{\mathcal{M}} = \varnothing$, $Assessed = \varnothing$, $\vec{G}(\boldsymbol{X}, \vec{E})$, a directed graph with $\boldsymbol{X}$ as vertices and no edges, $\vec{E} = \varnothing$.
Sort the $\boldsymbol{x}$'s in decreasing order of γ values.
Find $\boldsymbol{x} = \arg\max_{\boldsymbol{x} \in \boldsymbol{X}} \gamma(\boldsymbol{x})$.
Define $\lambda := \hat{f}_k(x)$.
Let $\boldsymbol{S}_\beta(\boldsymbol{x})$ be the connected component of $G(\lambda - \beta\lambda)$ containing $\boldsymbol{x}$.
Add $\boldsymbol{S}_\beta(\boldsymbol{x})$ to $\widehat{\mathcal{M}}$ and to *Assessed*.
repeat
    Find $\boldsymbol{x} = \arg\max_{\boldsymbol{x} \in \boldsymbol{X}} \{\gamma(\boldsymbol{x}) : \boldsymbol{x} \notin Assessed\}$.
    Define $\lambda := \hat{f}_k(\boldsymbol{x})$.
    Let $\boldsymbol{S}_\beta(\boldsymbol{x})$ be the connected component of $G(\lambda - \beta\lambda)$ containing $\boldsymbol{x}$.
    Add $\boldsymbol{S}_\beta(\boldsymbol{x})$ to *Assessed*.
    if $\boldsymbol{S}_\beta(\boldsymbol{x})$ is disjoint from all cluster cores in $\widehat{\mathcal{M}}$ **then**
        Add $\boldsymbol{S}_\beta(\boldsymbol{x})$ to $\widehat{\mathcal{M}}$.
    end if
until $\boldsymbol{X} \subseteq Assessed$.
for each $\boldsymbol{x}$ in $\boldsymbol{X} \backslash \widehat{\mathcal{M}}$ **do**
    Add a directed edge from $\boldsymbol{x}$ to $b(\boldsymbol{x})$.
end for
for each estimated cluster core $\widehat{\boldsymbol{M}} \in \widehat{\mathcal{M}}$ **do**
    Let $\boldsymbol{C}$ be the collection of the points connected by any directed path in $\vec{G}(\boldsymbol{X}, \vec{E})$ that terminates in $\widehat{\boldsymbol{M}}$.
    Add $\boldsymbol{C} \cup \widehat{\boldsymbol{M}}$ to $\widehat{\mathcal{C}}$.
end for
return $\widehat{\mathcal{C}}$

4.4.2 The DCF Algorithm

The DCF algorithm takes as input the dataset $\boldsymbol{X}$ and uses parameters k and β to return the final set of clusters $\widehat{\mathcal{C}}$. Initially, the set of estimated cluster cores $\widehat{\mathcal{M}} = \varnothing$ and the cluster assignment graph $\vec{G}(\boldsymbol{X}, \vec{E})$ is initialised with vertices as the points of $\boldsymbol{X}$ and no edges. DCF computes the peak-finding criterion for each point and selects the instance $\boldsymbol{x}$ with maximal value (Lines 1-2). The density level is set to $\lambda - \beta\lambda$ where $\lambda = \hat{f}_k(\boldsymbol{x})$. The level set graph $G(\lambda - \beta\lambda)$ is found, and the component set $\boldsymbol{S}_\beta(\boldsymbol{x})$ containing x is the first cluster core and is added to $\widehat{\mathcal{M}}$. All points from $\boldsymbol{S}_\beta(\boldsymbol{x})$ are added to *Assessed* and thus excluded from further consideration (Lines 3-5).

Next, the instance $\boldsymbol{x}$ with maximal value of the peak-finding criterion yet to be assessed is selected. The density level is set to $\lambda - \beta\lambda$ where $\lambda = \hat{f}_k(\boldsymbol{x})$ (Lines 7-8). The level set graph $G(\lambda - \beta\lambda)$ is updated and the component set of $G(\lambda - \beta\lambda)$ containing $\boldsymbol{x}$, namely $\boldsymbol{S}_\beta(\boldsymbol{x})$, is found. Firstly, all points in $\boldsymbol{S}_\beta(\boldsymbol{x})$ are added the set *Assessed* and hence excluded from future consideration as cluster cores (Lines 9-10). Then, if $\boldsymbol{S}_\beta(\boldsymbol{x})$ is disjoint from all sets in $\widehat{\mathcal{M}}$, then $\boldsymbol{S}_\beta(\boldsymbol{x})$ is added to $\widehat{\mathcal{M}}$ (Lines 11-13). The algorithm proceeds adding will be treated as a cluster core and cluster cores to $\widehat{\mathcal{M}}$ until the termination criteria is met, i.e., all points have been added to *Assessed* (Line 14).

After the set of estimated cluster cores has been returned, edges are added to the graph $\vec{G}(\boldsymbol{X}, \vec{E})$ from each non-core point $\boldsymbol{x}$ to $b(\boldsymbol{x})$ (Lines 15-17). The points in a cluster core $\widehat{\boldsymbol{M}}$ together with all the vertices that have paths terminating in $\widehat{\boldsymbol{M}}$ form a cluster that is added to $\widehat{\mathcal{C}}$ (Lines 18-22). Proceeding in this way, each sample point will be assigned to a unique cluster.

4.5 Analysis of DCF

DCF selects the first point in the ordered sequence (of remaining points) to identify the cluster core and then removes all the points in the cluster core from the ordered

sequence. As DCF assesses only a small fraction ($\approx 2\%$) of the data points, and the scale of the distance $\omega(\boldsymbol{x})$ is problem specific, it is not guaranteed that DCF will detect all the cluster cores.

4.5.1 Theoretical Analysis

In this section, we provide theoretical insights into DCF in two forms: firstly giving guarantees about the performance of DCF for samples of any size; secondly, we provide an analysis showing that DCF extends the theoretical guarantees available to the density peaks clustering method in Section 3.3. We demonstrate that DCF can, with high probability, estimate each cluster core of the underlying probability density bijectively.

Operational Guarantees

Here, we provide guarantees on the detection of disconnected clusters with DCF. We describe a problematic case in which DCF will fail to detect the true clusters in the data, and subsequently demonstrate why such a case is unlikely to occur in real datasets. Finally, we provide guarantees related to the termination criterion used in DCF.

Proposition 1. *Any cluster that corresponds to a component set in the mutual k-NN graph will be recovered by DCF.*

Proof. If the cluster corresponds to a component set in the mutual k-NN graph G, the related cluster core is a subgraph of the component set and hence is disconnected with other component sets in the graph. The if statement in Line 11 of Algorithm 4.4.2 will always be satisfied. According to the definition of a mutual k-NN graph, the points of the cluster that are not in the cluster core will always be linked to a point in the cluster core by a directed path. □

It is immediately clear from Prop. 1 that

Corollary 1. *If all clusters are mutually disconnected in the graph G, then DCF will*

recover the exact clustering.

We here describe a scenario in which DCF fails to detect the true clusters in the data. Consider the mode estimate $\boldsymbol{x}_T$ at which the procedure will terminate, with density $\hat{f}_k(\boldsymbol{x}_T) = \lambda_T$, and distance to its nearest neighbor of higher local density $\omega(\boldsymbol{x}_T)$. The peak-finding criterion value at $\boldsymbol{x}_T$ is $\gamma(\boldsymbol{x}_T)$. When $\lambda_T - \beta\lambda_T$ is less than the minimum density level in the data. The level set graph $G(\lambda_T - \beta\lambda_T)$ contains all the points in $\boldsymbol{X}$, and hence the algorithm terminates. Consider the illustrative example in Figure 4.3. Here we see a component set $\boldsymbol{S}_\beta(\boldsymbol{x}_T)$ containing $\boldsymbol{x}_T$. As the density level of all points in $\boldsymbol{S}_\beta(\boldsymbol{x}_T)$ is at least $\lambda_T - \beta\lambda_T$, once $\boldsymbol{x}_T$ is assessed the algorithm will terminate. The point $\boldsymbol{x}^*$ with density $\hat{f}_k(\boldsymbol{x}^*) = \lambda^*$, will only be assessed by DCF if $\gamma(\boldsymbol{x}^*) > \gamma(\boldsymbol{x}_T)$. As $\lambda^* > \lambda_T$, we require

$$\omega(\boldsymbol{x}^*) > \frac{\lambda_T}{\lambda^*}\omega(\boldsymbol{x}_T), \tag{4.1}$$

for $\boldsymbol{x}^*$ to be assessed. As the quantity $\omega(\boldsymbol{x}_T)$ is not bounded in general, we cannot guarantee that all modes will be assessed in the DCF procedure.

It should be noted that the scenario described above is unusual in the peak-finding context. Previous work has highlighted the inability of the peak-finding criterion to

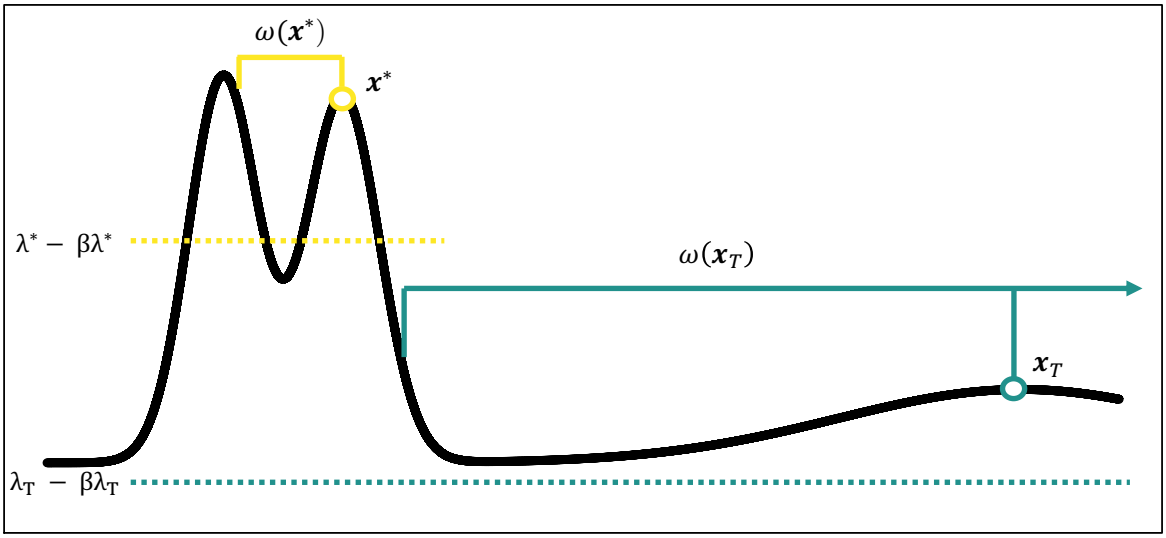


Figure 4.3: An illustration of the difficulties posed by the peak-finding criterion. Note that the termination point $\boldsymbol{x}_T$ has potentially unbounded $\omega(\boldsymbol{x}_T)$, inhibiting detection of the cluster core containing $\boldsymbol{x}^*$.

detect low-density clusters. To understand why this scenario is unlikely in practice, we adapt results from studies of mutual k-NN graphs to show that the probability of there being a path in the connected graph between $\boldsymbol{x}_T$ and any other cluster decreases as the distance $\omega(\boldsymbol{x}_T)$ increases (Maier et al., 2009).

The distance between sets $\boldsymbol{A}, \boldsymbol{B} \subseteq \mathbb{R}^p$ is $d(\boldsymbol{A}, \boldsymbol{B}) = \inf\{\|\boldsymbol{x} - \boldsymbol{y}\| : \boldsymbol{x} \in \boldsymbol{A}, \boldsymbol{y} \in \boldsymbol{B}\}$. We denote the closest connected component to $\boldsymbol{S}_\beta(\boldsymbol{x}_T)$ as $\boldsymbol{S}'_\beta(\boldsymbol{x}_T)$. We assume that there exists $\tilde{\beta} > 0$ such that $d(\boldsymbol{S}_{\tilde{\beta}}(\boldsymbol{x}_T), \boldsymbol{S}'_{\tilde{\beta}}(\boldsymbol{x}_T)) \geq u_T > 0$ for some u_T representing the lower bound between the components. Note that $u_T \leq \omega(\boldsymbol{x}_T)$ as $\omega(\boldsymbol{x}_T)$ is the distance from $\boldsymbol{x}_T \in \boldsymbol{S}_{\tilde{\beta}}(\boldsymbol{x}_T)$ to a point in the component $\boldsymbol{S}'_{\tilde{\beta}}(\boldsymbol{x}_T)$. We require a lower bound on the probability mass of balls of radius u_T around points in $\boldsymbol{S}_{\tilde{\beta}}(\boldsymbol{x}_T)$

$$\rho_T \leq \inf_{\boldsymbol{x} \in \boldsymbol{S}_{\tilde{\beta}}(\boldsymbol{x}_T)} \mu\left(B(\boldsymbol{x}, u_T)\right),$$

where $\mu(\cdot)$ is the Lebesgue measure on $\mathbb{R}^p$. As the distance from $\boldsymbol{S}_{\tilde{\beta}}(\boldsymbol{x}_T)$ to all other clusters increases, the value of ρ_T increases. We denote by $r_k(\boldsymbol{A}) = \max_{\boldsymbol{x} \in \boldsymbol{A}} r_k(\boldsymbol{x})$ the maximal k-NN radius of the points in $\boldsymbol{A}$. Finally, it is required to denote by $\mathcal{D}_\beta$ the event in which $\|\hat{f}_k(\boldsymbol{x}) - f(\boldsymbol{x})\| \leq \tilde{\beta}\lambda_T$ for all sample points in $\boldsymbol{S}_{\tilde{\beta}}(\boldsymbol{x}_T) \cup \boldsymbol{S}'_{\tilde{\beta}}(\boldsymbol{x}_T)$.

Proposition 2 (Prop. 6 of Maier et al. (2009))**.** *Let $\mathcal{I}_\beta$ denote the event that the subgraph of samples in $\boldsymbol{S}_{\tilde{\beta}}(\boldsymbol{x}_T)$ is isolated in $G(\lambda_T - \beta\lambda_T)$. Then, given $\beta < \tilde{\beta}$, $k < \frac{\rho_T n}{2} - 2\log(\mu(\boldsymbol{S}_{\tilde{\beta}}(\boldsymbol{x}_T))n)$ we obtain*

$$\begin{aligned}
\Pr\left((\mathcal{I}_\beta)^c\right) &\leq \Pr\left(r_k(\boldsymbol{S}_{\tilde{\beta}}(\boldsymbol{x}_T)) \geq u_T\right) + \Pr\left(\mathcal{D}_\beta\right) \\
&\leq \exp\left(-\frac{n-1}{2}\left(\frac{\rho_T}{2} - \frac{k-1}{n-1}\right)\right) + \Pr\left(\mathcal{D}_\beta\right).
\end{aligned}$$

From this result, we see that the probability of the set being connected in the graph decreases as the quantity ρ_T, and thus u_T, increases. To interpret this result in the context of DCF, consider fixing the pairwise distances of points within $\boldsymbol{S}_\beta(\boldsymbol{x}_T)$, and

increasing $\omega(\boldsymbol{x}_T)$. We see that as the magnitude of $\omega(\boldsymbol{x}_T)$ increases, the magnitude of u_T increases accordingly, and the probability of $\boldsymbol{x}_T$ being connected to the remainder of the graph decreases.

There are several statements regarding the termination level that can be guaranteed.

Proposition 3. *If* DCF *terminates at $\boldsymbol{x}_T$ with termination density level $\lambda_T - \beta\lambda_T$, $\lambda_T - \beta\lambda_T$ is at least as low as the lowest dip in density between clusters in $\boldsymbol{X}$.*

Proof. Following the discussion above, if $\boldsymbol{x}_T$ is the termination point, then $\lambda_T - \beta\lambda_T$ is at most the minimum local density of the points in the data. The result follows. □

While obvious, this statement ensures that the termination density level is low when there is a reasonable degree of separation between clusters in the data.

Proposition 4. *If DCF assesses a point $\boldsymbol{x}$ with $\hat{f}_k(\boldsymbol{x}) = \lambda < \lambda_T$ and $\boldsymbol{x}$ lies in the same connected component of G as $\boldsymbol{x}_T$, the connected component of $G(\lambda - \beta\lambda)$ containing $\boldsymbol{x}$ will not be accepted as a cluster core.*

Proof. Taking the set of points in the connected component of G containing $\boldsymbol{x}$ and $\boldsymbol{x}_T$ to be $\boldsymbol{X}'$ and the connected component of $G(\lambda - \beta\lambda)$ to be $\boldsymbol{S}_\beta(\boldsymbol{x})$. For $\boldsymbol{S}_\beta(\boldsymbol{x})$ to be accepted as an estimated cluster core, we require $A_\beta(\boldsymbol{x})$ to be disjoint from all cluster cores in $\widehat{\mathcal{M}}$. But $\lambda < \lambda_T$, hence $G(\lambda_T - \beta\lambda_T) \subseteq G(\lambda - \beta\lambda)$. As $\boldsymbol{x}_T$ is the termination point, $G(\lambda_T - \beta\lambda_T)$ is connected and $\boldsymbol{X}' \subseteq G(\lambda_T - \beta\lambda_T)$. Hence, $\boldsymbol{S}_\beta(\boldsymbol{x})$ is not disjoint from any cluster core in $\widehat{\mathcal{M}}$. □

This guarantees that no modes are missed even if the termination density level is high.

Consistency Guarantees

The notion of cluster cores can be understood, from a different perspective, as a method for pruning spurious estimates from the set of estimated modes. The cluster cores func-

tion in a similar way to the pruning method of Chaudhuri et al. (2014). There, the authors consider the problem of estimating the cluster tree using nearest neighbor graphs. Suppose $\boldsymbol{S}$, $\boldsymbol{S}' \subset \boldsymbol{X}$ are not connected in $G(\lambda)$ at some level λ. They demonstrate that a procedure which reconnects $\boldsymbol{S}$ and $\boldsymbol{S}'$ if they are connected in $G(\lambda')$ where $\lambda' < \lambda$ is a nearby lower level of the density correctly prunes spurious separations in the cluster tree.

Dasgupta and Kpotufe (2014) translates this framework for mode detection. As the component sets of the cluster tree will, if correctly estimated, each contain one mode of the underlying density, the pruning method allows for bijective estimation of the true modes above a certain density level λ using nearest neighbor graphs. Further, since $\lambda \to 0$ as $n \to \infty$, they demonstrate that their procedure consistently prunes false modes. The analogy to cluster cores is easily drawn. The DCF procedure will only retain an estimated mode, say $\boldsymbol{x}^*$, with $\hat{f}_k(\boldsymbol{x}^*) = \lambda^*$ if is is contained in a separate component set of the graph $G(\lambda^* - \beta\lambda^*)$. The process of determining a cluster core is thus equivalent to the pruning procedure undertaken in Dasgupta and Kpotufe (2014). The correspondence allows for the following result, given in previous work on cluster cores in Jiang et al. (2018), stating that the cluster core estimates returned by DCF estimate the cluster cores of f bijectively and consistently. Before proceeding, the definition of the Hausdorff distance between two sets is required.

Definition 10. *For $\boldsymbol{M} \subset \mathcal{X}$, $\boldsymbol{x} \in \mathcal{X}$, let $d(\boldsymbol{x}, \boldsymbol{M}) \coloneqq \inf_{\boldsymbol{x}' \in \boldsymbol{M}} \|\boldsymbol{x} - \boldsymbol{x}'\|$. The Hausdorff distance between $\boldsymbol{A}, \boldsymbol{B} \subset \mathcal{X}$ is defined as $d(\boldsymbol{A}, \boldsymbol{B}) \coloneqq \max\{\sup_{\boldsymbol{x} \in \boldsymbol{A}} d(\boldsymbol{x}, \boldsymbol{B}), \sup_{\boldsymbol{x}' \in \boldsymbol{B}} d(\boldsymbol{x}', \boldsymbol{A})\}$.*

Theorem 3 (Adapted from Theorem 1 of Jiang et al. (2018))**.** *Suppose Assumptions 1-4 hold. Further assume the event in Proposition 2, namely that no cluster core has been ignored by early termination of the algorithm. Let $0 < \beta < 1$ and $\epsilon, \zeta > 0$ and suppose that $k = k(n)$ is chosen such that $\log^2 n/k \to 0$ and $n^{4/(4+p)}/k \to 0$. Let $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_m$ be the cluster cores of f. Then for n sufficiently large depending on f, ζ, ϵ and β, with probability at least $1 - \zeta$, DCF returns m cluster core 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 $i = 1, \ldots, m$.*

Proof. To prove this theorem, we require (1) guarantees on the rate at which a cluster core $\boldsymbol{M}$ is approximated by some estimate in $\widehat{\mathcal{M}}$ and (2) guarantees that any estimated cluster core in $\widehat{\mathcal{M}}$ at a sufficiently high level, corresponds to a true cluster core of f at a nearby level.

The first result required is Theorem 3 of Jiang and Kpotufe (2017). In that work, the theorem is proved for all densities f. We present the theorem adapted for the assumption that the density is Hölder continuous, in Assumption 1.

Theorem 4 (Theorem 3 of Jiang and Kpotufe (2017))**.** *Let $\zeta > 0$ and $\boldsymbol{M}$ be a cluster core. Select $k = k(n)$ such that $\log^2(n)/k \to 0$ and $n^{4/(4+p)}/k \to 0$ as $n \to \infty$. Then with probability at least $1 - \zeta$, there exists $\widehat{\boldsymbol{M}} \in \widehat{\mathcal{M}}$ such that*

$$d(\boldsymbol{M}, \widehat{\boldsymbol{M}}) \leq \sqrt{\frac{8c_{\zeta n}}{\check{c}} \cdot \lambda} \cdot \frac{1}{k^{1/4}},$$

where $\lambda = \max_{\boldsymbol{x} \in \boldsymbol{M}} f(\boldsymbol{x})$.

Thus for all cluster cores $\boldsymbol{M}$ of f, $\widehat{\mathcal{M}}$ contains at estimate of it at the given rate. For instance, the choice $k = O(n^{4/(4+p)})$ optimizes the rate to $O(n^{-1/(4+p)})$. The second result adapts Theorem 4 of Jiang and Kpotufe (2017), again presented for Hölder continuous densities.

Theorem 5 (Theorem 4 of Jiang and Kpotufe (2017))**.** *Let $0 < \zeta < 1$ and let k be chosen such that $\log^2(n)/k \to 0$ and $n^{4/(4+p)}/k \to 0$ as $n \to \infty$. There exists $\lambda_0 = \lambda_0(n, k)$ such that the following holds with probability at least $1 - \zeta$. All cluster core estimates in $\widehat{\mathcal{M}}$ chosen with maximum density λ at level $\lambda \geq \lambda_0$ can be injectively mapped to cluster cores $\left\{\boldsymbol{M} : \lambda_{\boldsymbol{M}} \geq \min_{\boldsymbol{x} \in \boldsymbol{X} : f_k(\boldsymbol{x}) \geq \lambda - \beta\lambda} f(\boldsymbol{x})\right\}$. Furthermore, $\lambda_0 \to 0$ as $n \to \infty$.*

Thus under the assumption that f is Hölder continuous, any estimate above the level λ_0 corresponds to a true cluster core of f. Taken together, these results immediately imply the statement in the theorem. The second result guarantees that for n sufficiently

large, any cluster core $\boldsymbol{M}_i$ will have an estimate $\widehat{\boldsymbol{M}}_i \in \widehat{\mathcal{M}}$. The first result shows that the estimate $\widehat{\boldsymbol{M}}_i \to \boldsymbol{M}_i$ in Hausdorff distance, yielding both $\boldsymbol{M}_i \cap \boldsymbol{X} \subseteq \widehat{\boldsymbol{M}}_i$ and $\widehat{\boldsymbol{M}}_i \subseteq \boldsymbol{M}_i + B(0, \epsilon)$ for $i = 1, \ldots, m$.

□

This result shows that DCF recovers estimates of the population cluster cores bijectively. Further, the result in Theorem 2 shows that the assignment strategy of DCF allows for the correct clustering the the attraction regions associated with the cluster cores.

4.5.2 Complexity Analysis

The most computation-intensive task is creating the mutual k-NN graph which requires $O(nk \log(n))$ operations on average. Another major computational burden is finding, for each point, its nearest neighbor of higher density. For the points which do not have a point of higher density in their neighbors, this requires $O(n)$ operations. In practice, the proportion of instances without a point of higher density in their neighbors is typically less than 1%, as long as k is not too small. As such, for the vast majority of instances, this requires $O(k)$ operations. Assessing each cluster core requires $O(nk)$ operations. The assignment mechanism requires $O(n)$ operations. As such, we see that the complexity of DCF is near linear in n and k.

4.5.3 Simulated Experiments

In order to demonstrate that DCF does not, in practice, ignore modes due to extreme values of $\omega(\boldsymbol{x}_T)$ at termination, we compare DCF and QuickShift++ on a range of simulated datasets.

The analysis in the previous section indicates that the performance of DCF depends on the degree of separation of clusters. To assess the impact of separation on our method, we generate data as proposed in Dasgupta (1999) and Verbeek et al. (2003). Gaussian

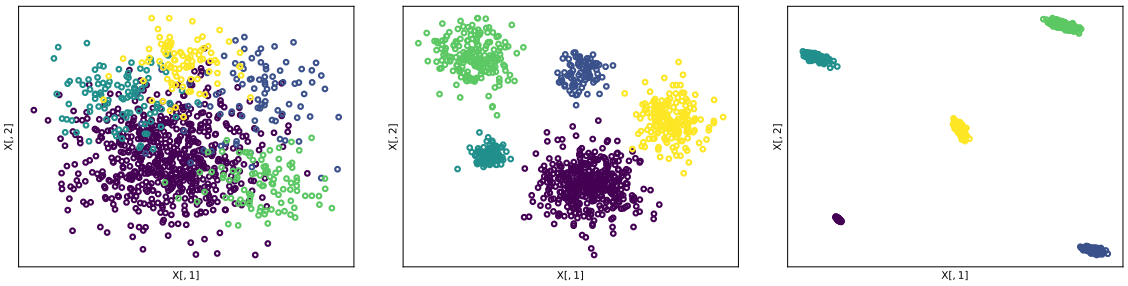


Figure 4.4: Three of the generated datasets used in Section 4.5.3, with separation values $c = 0.2, 1.5$ and 4.2 respectively.

components are sampled so that their means satisfy the following inequality:

$$\forall_{i \neq j} : \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\| \geq c \max_{i,j} \{\text{tr}(\boldsymbol{\Sigma}_i), \text{tr}(\boldsymbol{\Sigma}_j)\},$$

where c controls the degree of separation of the clusters. For each mixture component, we fix the eccentricity (the ratio of the largest to the smallest eigenvalue of the covariance matrix) as one. We generate mixtures of five components with data sizes ranging from $n = 1000$ to $n = 2000$ in increments of 20, data of dimension $p = \{2, 5\}$, and degrees of separation ranging from $c = 0$ (low) to $c = 5$ (high) in increments of 0.05. For each configuration we repeat 10 times. Three of the generated datasets are show in Figure 4.4. For both DCF and QuickShift++, we assess $k = \{40, 50\}$ and $\beta = \{0.3, 0.5, 0.7\}$.

In Figure 4.5, the similarities in performance of DCF and QuickShift++ are shown. In the left plot, we compare the number of cores recovered from the data as the separation parameter c increases. The number of cores returned by DCF tracks the number returned by QuickShift++. In fact, the number of cores returned by DCF is fractionally higher than that of QuickShift++ for a large range of values of the parameter c. This is a direct result of the different approaches to the ordering of data points. This result demonstrates the ability of DCF to recover modes in the data. The quality of the clusterings returned by both methods, measured by the Adjusted Rand Index (ARI), were within 1% for each value of c and, as a result, are not shown.

On the right of Figure 4.5, we include an analysis of the fraction of points assessed

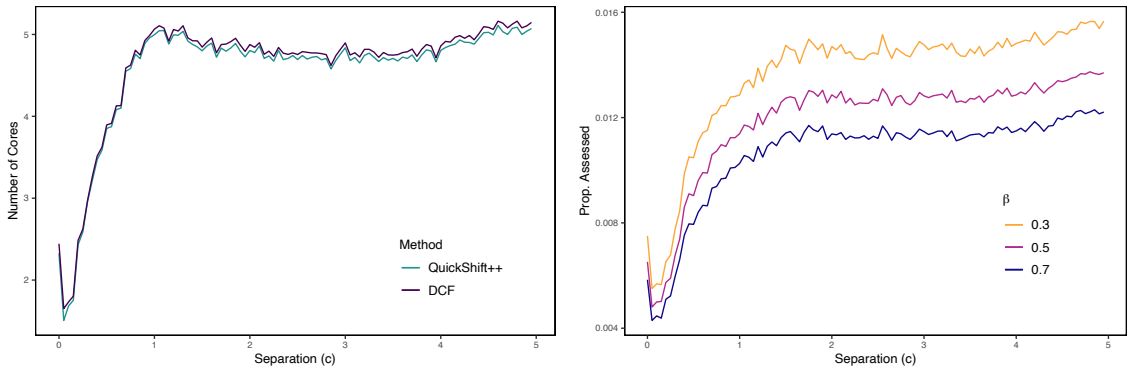


Figure 4.5: Left: The number of cluster cores (#Cores) recovered from each dataset as a function of the separation parameter c for both methods. These results are averaged over the different data sizes, dimensions, repetitions, and parameter values. Right: Plotted is the proportion of instances assessed as potential modes (Prop. Assessed) in DCF for values of $\beta = \{0.3, 0.5, 0.7\}$ as a function of the separation parameter c.

by DCF for three values of $\beta = \{0.3, 0.5, 0.7\}$. Recall that QuickShift++ assesses every instance in $\boldsymbol{X}$. For each of the three values, we see that the proportion of instances assessed by DCF is less than 2%, an impressive result given the quality of the clusterings returned. This highlights the ability of the peak-finding criterion to determine modes in the data efficiently.

4.5.4 Real-World Experiments

In this section, we compare the performance of DCF with the two method most similar to it in the literature, namely QuickShift++ and the density peaks clustering algorithm of Chapter 3. DCF is motivated in two ways (1) that algorithms which estimate cluster cores are more robust than those that seek point modes in the data, and (2) that the peak-finding criterion provides a more efficient method for searching for cluster cores in the data, than does the search in descending order of the density used in QuickShift++. To quantify the clustering performance and algorithmic efficiency improvements, we here evaluate all three methods on a selection of real-world datasets. All experiments have been conducted on a PC running Debian 10 (Buster), consisting of 24 cores and 24GB of RAM. Our method is implemented in Python; its source code, and code to reproduce the below experiments is available online.[1]

[1] https://github.com/tobinjo96/Thesis-Experiments/

Experimental Set-Up

DCF is assessed on five real-world datasets collected from the UCI Machine Learning Repository (Dua and Graff, 2019) and the Phonemes dataset (Hastie et al., 2009). Details of the datasets can be found in Table 4.1. Instances with missing values are removed. To assess the performance of DCF on these datasets, we apply the density peaks clustering method (DPC) with the k-NN density estimator implemented in Python and QuickShift++ (QSP) (Jiang et al., 2018) implemented in Python and C++.

To evaluate the clusterings produced by DCF, CPF, and the competitor methods, we adopt two widely used external indices: Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) and Adjusted Mutual Information (AMI) (Vinh et al., 2010), each comparing clusterings to ground truth labels available from the data. For both metrics, a larger value indicates a higher-quality clustering.

Results

The results of the clustering are presented in Table 4.2 and Table 4.3. DCF achieves the best clustering, for at least one of the metrics for every data set analyzed. The performance of QuickShift++ is comparable to that of DCF, reflecting the similarities in the methodologies. The density peaks clustering method performs well for the Dermatology and Phonemes datasets. The performance is not consistent however, with poor clustering returned for the Ecoli and Letter Recognition datasets, in particular. It can be concluded that cluster cores effectively model high density regions in the data, and prove more robust to the noise present in real world samples compared to point estimates of the modes.

The average run time, in seconds, for the three methods is presented in Table 4.3. For the small datasets used in the analysis, namely the Dermatology, Ecoli and Glass datasets, QuickShift++ executes the fastest, with DPC and DCF marginally slower. It is likely that the implementation of QuickShift++ in C++ is better optimized than

Source	**Name**	n	p	m
Dua and Graff (2019)	Dermatology	358	34	6
Dua and Graff (2019)	Ecoli	336	7	8
Dua and Graff (2019)	Glass	214	9	6
Dua and Graff (2019)	Letter Recognition	20000	16	26
Dua and Graff (2019)	Page Blocks	5743	10	5
Hastie et al. (2009)	Phonemes	4509	256	5

Table 4.1: Characteristics of the real-world datasets.

the implementations of DCF and DPC used in the analysis. We see that for larger datasets, the methods that use the peak-finding criterion have the fastest run time. The difference is most pronounced for the Letter Recognition dataset, the dataset with the most points in our experiments. DCF achieves a roughly 40% speed up compared to QuickShift++ for this dataset, as the search is restricted to points with large values of the peak-finding criterion. To support the run time analysis, the proportion of instances visited by DCF is presented in Figure 4.6. This proportion is almost always less than 1% for Letter Recognition, Page Blocks and Phonemes, the three larger datasets in our analysis.

Further experimental analysis, as well as a thorough analysis of the parameter space for DCF is provided in Chapter 6.

4.6 Application

Face recognition has become a central problem in deep learning in recent years (Wang et al., 2018a). However, the majority of work relies on large labelled datasets for use in training. As the number and volume of datasets increases, effective unsupervised face recognition methods will be required. DCF is applied to two large image datasets to demonstrate its ability to perform unsupervised face recognition. We use a sample of the MS-Celeb-1M data (Guo et al., 2016) consisting of 17,146 identities, each with roughly 100 images. The YouTube Face dataset (YTB-Faces) (Wolf et al., 2011) is another benchmark image dataset. We use the sample of this dataset in Yang et al.

Dataset	Metric	DCF	QSP	DPC
Dermatology	ARI	**0.73**	0.70	0.72
	AMI	0.78	0.78	**0.84**
Ecoli	ARI	**0.73**	**0.73**	0.47
	AMI	**0.68**	**0.68**	0.57
Glass	ARI	**0.31**	**0.31**	0.24
	AMI	**0.41**	**0.41**	0.31
Letter Recognition	ARI	**0.20**	**0.20**	0.05
	AMI	**0.59**	**0.59**	0.25
Page Blocks	ARI	**0.48**	**0.48**	0.22
	AMI	**0.32**	**0.32**	0.21
Phonemes	ARI	**0.76**	**0.76**	0.75
	AMI	**0.83**	0.81	0.81

Table 4.2: Quality of clusterings for the real-world datasets.

Dataset	DCF	QSP	DPC
Derm.	0.10	**0.03**	0.10
Ecoli	0.09	**0.02**	0.08
Glass	0.08	**0.05**	0.07
Letter R.	11.13	19.21	**10.43**
Page B.	0.73	1.61	**0.66**
Phonemes	**8.34**	8.79	9.18

Table 4.3: Average run time of the assessed clustering methods for the real-world datasets.

(2019) of 155,282 frames with 1,595 identities. For both datasets, we apply DCF to numerical features extracted using a trained CNN.[2] Sample images from the datasets are presented in Figure 4.7 and details of the datasets can be found in Table 4.4.

To assess the performance of DCF on these datasets, we compare it directly with Quick-Shift++ as is scalable to large datasets and does not require the number of clusters to be provided as an input. As the number of clusters is not likely to be provided for such an application, scalable adaptations of density peaks clustering (e.g. the method introduced in Sieranoja and Fränti (2019)) are not suitable for comparison.

[2]https://github.com/yl-1993/learn-to-cluster (GitHub Repository)

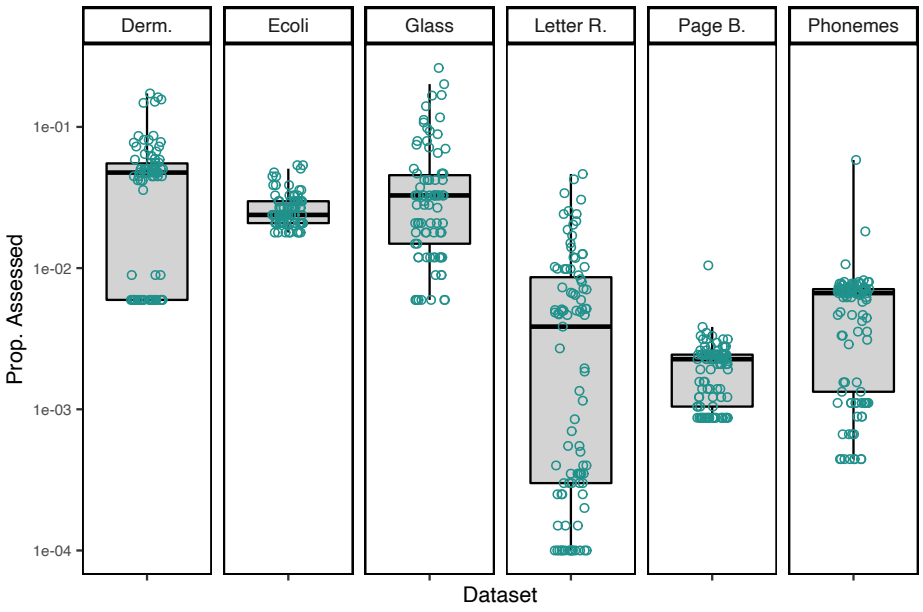


Figure 4.6: The proportion of instances assessed as modes by DCF for each of the six datasets and for all assessed parameter values. Note the log scaling of the y-axis.

Source	**Name**	n	p	m
Guo et al. (2016)	MS-Celeb-1M	1,160,507	256	17,146
Wolf et al. (2011)	YTB-Faces	155,282	256	1,595

Table 4.4: Characteristics of the face recognition datasets.

Dataset	**Metric**	**DCF**	**QSP**
MS-Celeb-1M	ARI	**0.90**	0.83
	AMI	**0.96**	0.92
YTB-Faces	ARI	**0.69**	0.52
	AMI	**0.91**	0.88

Table 4.5: Quality of clusterings for the face recognition datasets.

Dataset	**DCF**	**QSP**
MS-Celeb-1M	**835.14**	2787.77
YTB-Faces	**1419.02**	1593.89

Table 4.6: Average run time for the DCF and QuickShift++ methods for the two image datasets.

4.6.1 Results

The results of the experiments for both datasets are given in Table 4.6. DCF achieves exceptional results for both datasets. MS-Celeb-1M and YTB-Faces are challenging

(a) MS-Celeb-1M

(b) YTB-Faces

Figure 4.7: Three samples from two clusters present in each of the face recognition datasets.

datasets due to the large size and extremely large number of clusters. DCF achieves the highest quality clustering for both ARI and AMI. The impressive results of both DCF and QuickShift++ indicate that cluster cores are well suited to the problem of face detection. DCF shows slightly better performance. As clusters in face recognition data tend to be relatively well separated, the peak-finding criterion quickly traverses disconnected components in the mutual k-NN graph.

We report the average run time, in seconds, for each of the methods in Table 4.6. The peak-finding criterion leads to dramatically reduced run time for DCF compared to QuickShift++. For MS-Celeb-1M, DCF executes in less than 1/3 of the time of QuickShift++ and also executes faster for YTB-Faces.

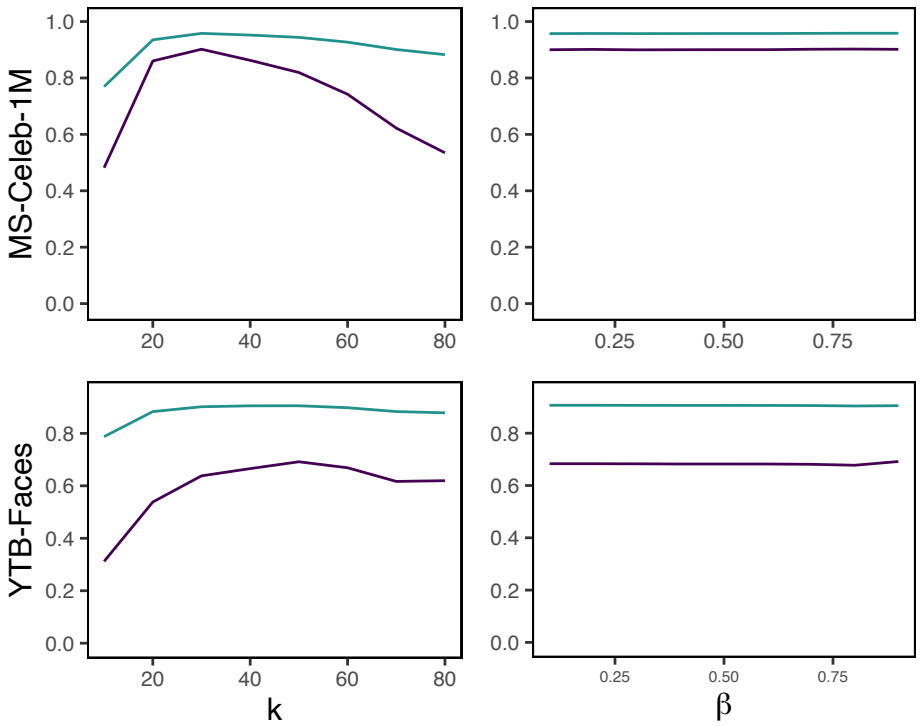


Figure 4.8: Analysis of effects of DCF parameters k and β for face detection datasets. The ARI is shown in purple and the AMI in green.

We provide an analysis of the parameter space for the face recognition datasets in Figure 4.8. Again we see that DCF is robust for a large range of values for k and β. As the size of the dataset increases, we observe that the optimal value for k is approximately $\log(n)$. Choosing $k \approx \log n$ instead of $\sqrt{n}$, has the added benefit of reducing the computation required to construct the k-NN graph.

4.7 Conclusion

This article introduced DCF, an improved algorithm for clustering using modal sets and density peaks. We showed that modelling high-density regions of the data using cluster cores achieves better results than using modes, and that the peak-finding criterion effectively locates cluster cores in the data. Theoretical guarantees on the performance of DCF were provided, supported by results from simulation studies. Experimental results further demonstrated that DCF has excellent performance. It achieves superior results over a range of benchmark datasets and indices when compared to the density

peaks clustering method, and achieves results similar to those of QuickShift++ with less computation. Finally, we showed the ability of DCF to perform unsupervised face recognition, a challenging contemporary clustering problem. DCF is both efficient and robust, gracefully scaling to big datasets and achieving exceptional performance over a broad range of parameter values.

5 Peak-Finding on Density-Level Sets

5.1 Summary

The density peaks clustering algorithm introduced in Chapter 3 and the density core finding method of Chapter 4 provide reliable estimates of the high density regions in the data. Moreover, as demonstrated previously the allocation method, which assigns instances to the same cluster as their nearest neighbor of higher estimated density is provably consistent as the sample size increases. However, for datasets of typical size, allocating instances using a sample-based analogy to gradient ascent of the density can fail. In this case, the clusterings returned by the density peaks clustering method, and its derivative methods, often contradict geometrical intuition about the structure of the clusters. Furthermore, as the method does not consider the changes in the density between instances, the allocation method can assign instances across regions of very low density. Aiming to remedy the issues with the allocation approach while retaining the excellent mode detection performance of the density peaks clustering method, we here develop a new scalable and automatic clustering algorithm combining the benefits of both level set and mode-seeking approaches to non-parametric clustering. The improvements are threefold: (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; (3) the computational complexity of the algorithm is near linear in the number of data points. We present extensive experimental results to verify that our algorithm works well in practice. Finally, a modified

version of our approach is presented, by integrating instance-level constraints on the clustering. We show that this modified version of our approach achieves excellent performance for an important problem in computer vision, multi-image matching.

5.2 Introduction

As discussed previously, non-parametric clustering methods conceptualise clusters as sets of objects spread over contiguous regions in the data space with consistently high density, separated from each other by contiguous regions of low density. These regions can be of arbitrary shape and size, and found deterministically for a given dataset using density-based approaches. Two contrasting methodologies dominate the literature in this area: level set approaches and mode-seeking methods.

Density-level set methods estimate clusters as the sets of points resulting from a cut through the probability density function at a certain density level. Such cuts induce connected regions in the feature space where the density is greater than the cut threshold. The points in each region form the clusters. As discussed in Section 2.2.2, the most prominent level set approach is DBSCAN (Ester et al., 1996). Furthermore, there exists an extensive literature describing how nearest neighbor (k-NN) graphs estimate density-level sets (Maier et al., 2009; Kpotufe and von Luxburg, 2011; Steinwart, 2011). Density-level sets quickly and reliably detect points of extremely low density and remove them as outliers. Moreover, they determine the number of clusters automatically. Such methods, however, are not well suited to data containing clusters of varying densities, where selecting the appropriate cut level can prove impossible. If no appropriate cut level recovers the true clusters, the clusters returned by level set methods are likely to under-segment the data.

Mode-seeking approaches, by contrast, take the approach of first aiming to estimate the regions of maximal density, i.e, the modes of the underlying probability distribution and subsequently exploit the shape of the density to assign instances to clusters based on the gradient of the density. The density peaks clustering algorithm, as shown in

Chapter 3 applies this methodology in a sample-based framework. The estimates of the modes are found using solely statistics of the observed sample, and the assignment method uses the observed data and the estimated density of each point to approximate the ascent of the density. The sample-based framework, while efficient and, as shown previously, provably consistent as the number of observations increases, is susceptible to errors resulting from estimating the density for the sample.

We introduce Component-wise Peak-Finding (CPF), a scalable and flexible density-based clustering method that combines the benefits of both density-level set and mode-seeking methods, first using connected components of a mutual k-NN graph to detect areas of the data separated by regions of very low density and remove outlying points. To each component set, we apply the density peaks clustering method to propose potential cluster centers. True centers and the resulting clusters are automatically extracted from the data using a criterion based on density-level sets. The complexity of our algorithm is of the order $O(nk \log(n))$, near linear in k and n. The benefits offered by combining density-level set and mode-seeking formulations include the ability to detect outliers, clusters of varying density and overlapping clusters, as well as robustness against spurious maxima of the density. These benefits are illustrated in Figure 5.1.

To demonstrate the adaptability of the CPF framework, we present a modified version of the method, CPF-Match, for multi-image matching. Multi-image matching methods aim to find correspondences between points in two or more images. The general framework can be applied to the task of matching object instances, shape matching or understanding the 3-D structure of an object from two-dimensional image sequences (Tron et al., 2017; Bernard et al., 2019; Wang et al., 2018b; Yan et al., 2015). A key feature of the multi-image matching problem is that no point can be matched in the clustering with points from the same image. CPF-Match modifies the assignment procedure of CPF to incorporate these constraints.

In Section 5.3 we provide existing guarantees on level set estimation methods; in Section

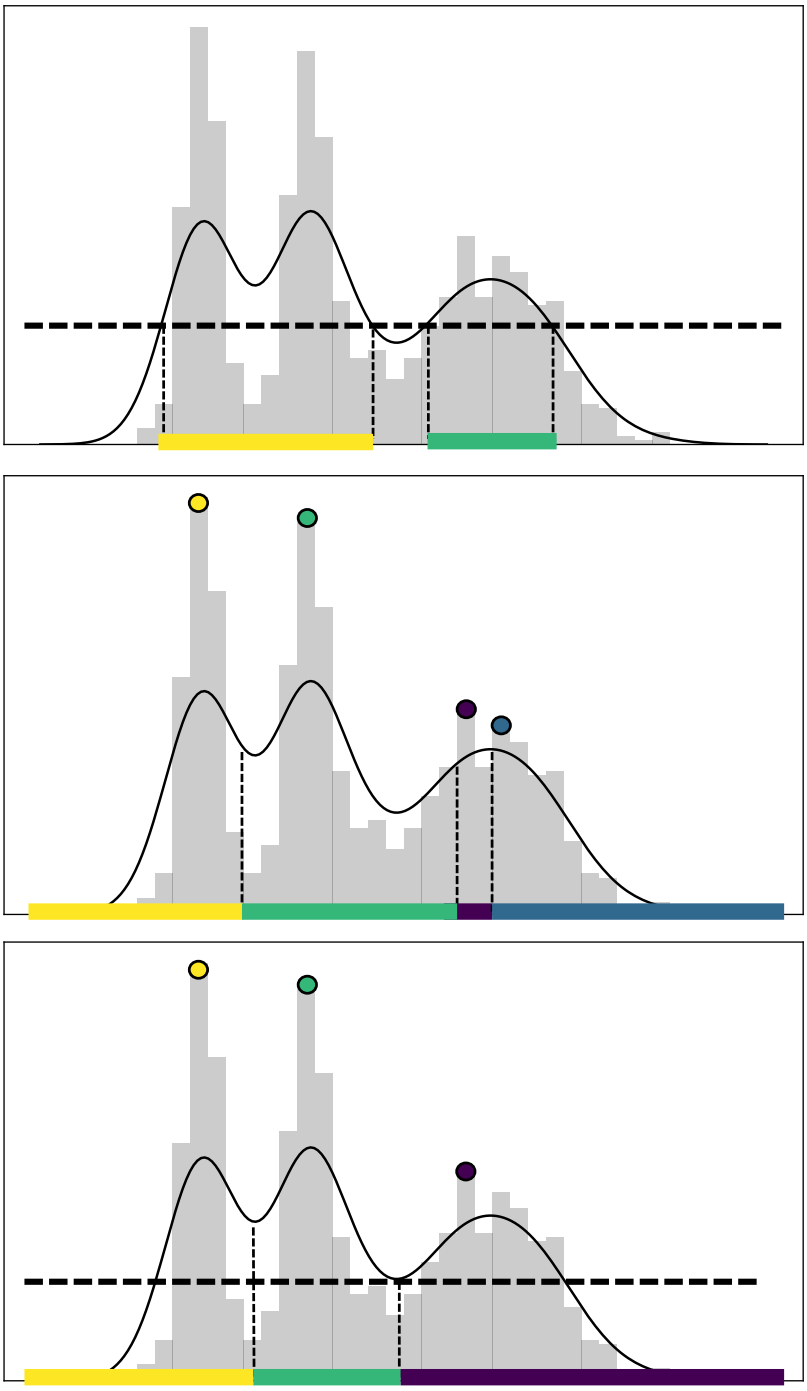


Figure 5.1: An illustration containing three clusters, two of high density and one of lower density. Clustering results are represented by colored solid lines. Top: Level set methods under-segment the data, as there is no level of the density at which a cut, represented by the dashed black line, will recover the true clusters. Middle: Mode-seeking approaches are susceptible to spurious local maxima in the data, leading to incorrect allocation, particularly of lower density clusters. Bottom: Combining the benefits of both approaches leads to successful clusterings.

5.4 we describe CPF in detail; in Section 5.5 we provide a brief analysis, demonstrating that CPF achieves the same theoretical guarantees on mode recovery as DCF and perform an ablation study analyzing the constituent methods of the CPF algorithm experimentally; in Section 5.6 we present and validate the CPF-Match method.

5.3 Related Work

The approach developed in this chapter adapts density-level set approaches to enhance the density peaks clustering algorithm.

Level set methods detect clusters as connected components of the level sets $\{\boldsymbol{x} : f(\boldsymbol{x}) \geq \lambda\}$ where f is the density and λ is the cutting threshold. The density f is unknown, and hence the level sets are required to be estimated from the data. Neighbor graphs have been widely used for this purpose (Jiang, 2017a; Chaudhuri and Dasgupta, 2010; Kpotufe and von Luxburg, 2011; Maier et al., 2009). Taking the instances to be the vertices of a graph, the process of defining edges to be added to the graph is regularly formulated in two different, but related, ways: ϵ-neighborhood graphs add an edge between two vertices if the distances between them is below a threshold value ϵ; k-NN graphs add edges between vertices according to the distance from a vertex to its k-th nearest neighbor. Symmetric k-NN graphs add an edge between vertices $\boldsymbol{x}$ and $\boldsymbol{x}'$ if $\boldsymbol{x}$ is in the k nearest neighbors of $\boldsymbol{x}'$ or vice versa. Mutual k-NN graphs add an edge between $\boldsymbol{x}$ and $\boldsymbol{x}'$ if both $\boldsymbol{x}$ and $\boldsymbol{x}'$ are within the k nearest neighbors of each other.

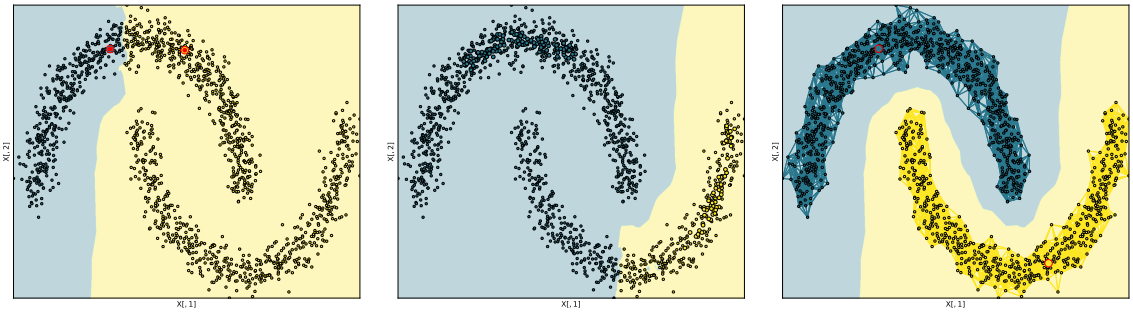
It has been shown that any density-level set of a given dataset can be approximated by the connected components of the mutual k-NN graph (Maier et al., 2009; Kpotufe and von Luxburg, 2011) and further work has aimed to develop an understanding of the optimal choice of k (Maier et al., 2009; Chaudhuri and Dasgupta, 2010; Steinwart, 2011). Procedures to improve the quality of the clusters recovered at a certain density level involve assessing whether the same clusters would be recovered at a slightly lower density level (Kpotufe and von Luxburg, 2011; Chaudhuri et al., 2014). As introduced

in Chapter 4, this process, termed pruning, reconnects clusters separated at high-density levels.

It has similarly been shown that ϵ-neighborhood graphs can recover density-level sets. A general method is developed in Chaudhuri and Dasgupta (2010). There, the connected components of the graph are tracked as ϵ increases from 0 to ∞. When vertices with few edges are removed at each level of ϵ, the process is shown to consistently estimate density-level sets at any density level. Furthermore, ϵ-neighborhood graphs provide the foundation of DBSCAN (Ester et al., 1996). In DBSCAN, edges are added between vertices if they lie within a threshold distance of each other. Vertices are removed from the graph if they have fewer edges than a threshold. For fixed values of its parameters, DBSCAN is shown to estimate the level sets at a given level of the density (Jiang, 2017a).

We initially partition the data into contiguous regions separated from each other by regions of very low density by building a mutual k-NN graph from the data. Kpotufe and von Luxburg (2011) show that level sets are higher density levels are estimated by removing edges from the graph in decreasing order of their length. To estimate the partition at a low level of the density, we retain all edges of the graph. As such, components that are separated in the mutual k-NN graph approximate regions separated by very low density in the sample space. Constraining the allocation mechanism of the density peaks clustering method to assign instances only to neighbors within the same region retains the speed and intuition of the sample-based allocation approach while significantly reducing the chances of low quality cluster assignments.

The existing literature developing methods to enhance the assignment strategy of density peaks clustering is scattered and poorly grounded in theoretical work. Xie et al. (2016) develop a strategy that assigns the k nearest neighbors of each mode estimate to a cluster, and subsequently assigns the remaining instances if they are within a threshold distance of the previously assigned points. The approach of Liu et al. (2018) is based on an inconsistent density estimate and convoluted computation in place of



(a) DPC (b) DCF (c) CPF

Figure 5.2: Comparison of CPF with (a) density peaks clustering and (b) density core finding clustering algorithms on the noisy moons dataset. Clusters and attraction regions are indicated by different colors. The mode and modal set estimates are the highlighted points in each plot. For the CPF method, the edges of the mutual k-NN graph are also shown.

the distance to nearest neighbor of higher estimated density. The work of Jiang et al. (2019) and Yu et al. (2019) also use inconsistent estimators in place of the density and develop assignment strategies based on distance thresholds that are challenging to implement in practice.

The approach taken here is simple and effective. Density-level sets have clear benefits for the density peaks clustering method, increasing the robustness of the allocation mechanism while being well-justified theoretically. Furthermore, combining the density peaks clustering method with k-NN graph level set estimators leads to a demonstrable improvement in performance, with no extra parameter tuning required.

5.4 Our Method

5.4.1 Motivation

The motivation for combining density-level set and mode-seeking formulations is illustrated in Figure 5.2. The figure depicts the noisy moons dataset, introduced in Section 3.4. The results of our method, CPF, are presented along with the results of density peaks clustering method as introduced in Chapter 3 and the density core finding method of Chapter 4.

As previously discussed, the density peaks clustering method is prone to selecting multiple centers from high density clusters due to noise in the density estimate. The negative impact of this is again clear from the example in the left panel of Figure 5.2. Two mode estimates are selected from the top cluster and none from the bottom cluster. This leads to all of the points in the bottom cluster being incorrectly assigned.

The density core finding method introduced in the previous chapter aims to remedy the issues with mode estimation by instead searching for high density regions in the dataset. The impact of directing the peak-finding criterion to detect cluster cores is seen in the middle panel of Figure 5.2. Here, the density core finding method first selects a cluster core from the top cluster. However, in contrast to the density peaks clustering method, the points assigned to the cluster core include many of the high density points in the first cluster. As a result, the second cluster core is correctly detected in the bottom cluster. The sample-based assignment method, common to both algorithms, is seen to fail in this instance. Instead of correctly recovering both clusters, the density core finding method incorrectly assigns many of the bottom cluster points to the higher density cluster. The assignment path for the incorrectly assigned instances must cross through regions of very low density in the data. This contravenes the understanding of a population cluster in the non-parametric formulation and is the issue we seek to remedy with the CPF approach.

The CPF procedure can be seen in the right panel of Figure 5.2. Initially partitioning the data into disjoint sets, termed component sets, using a mutual k-NN graph detects well-separated components with high density regions mutually separated from each other by low density regions. Subsequent application of a peak-finding based method detects only one cluster in each component, thus correctly recovering the clusters from the data. The clustering method applied to each component is similar to that of density core finding, allowing appropriate representation of high density regions in each component. Furthermore, it allows the procedure to operate automatically, without users being forced to select modes manually.

It is clear from the analysis of the datasets that both DCF and CPF offer improvements over the density peaks clustering method, while the CPF approach further offers a more robust allocation mechanism.

5.4.2 Notation and Definitions

In this section, we explain the component set notation and the peak-finding criterion. As before, we denote the mutual k-NN graph $G(\boldsymbol{X}, E)$.

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 $\mathcal{S} = \{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_{n_\mathcal{S}}\}$, where $n_\mathcal{S} = |\mathcal{S}|$ is the number of component sets, and $\boldsymbol{S}_1 \cup \cdots \cup \boldsymbol{S}_{n_\mathcal{S}} = \boldsymbol{X}$. Intuitively, two data points belonging to two different component sets are highly likely to belong to different clusters.

We now explain the mode selection mechanism. The definitions for the peak-finding technique used are the same as those given in Section 3.2. We next specify the mode estimation procedure used in CPF. 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 first instance with maximal value of the peak-finding criterion is automatically selected as a true cluster center. To decide whether or not to select the subsequent instances as candidate cluster centres, we here utilize an idea similar to that of cluster cores introduced in Chapter 4. A candidate cluster center $\boldsymbol{x}^*$ is accepted only when it is well separated from the others.

Definition 11. *Let $0 < \rho < 1$. For an instance $\boldsymbol{x}^* \in \boldsymbol{S}$, define a graph $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E_{\boldsymbol{x}^*})$ with*

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

and

$$E_{\boldsymbol{x}^*} = \{\{\boldsymbol{x}_i, \boldsymbol{x}_j\} \in E(\boldsymbol{S}) : \boldsymbol{x}_i \in \boldsymbol{V}_{\boldsymbol{x}^*}, \boldsymbol{x}_j \in \boldsymbol{V}_{\boldsymbol{x}^*}, \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq r_k(\boldsymbol{x}^*)\}.$$

We accept $\boldsymbol{x}^*$ as a cluster center if the connected component of the graph $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E_{\boldsymbol{x}^*})$ containing the vertex $\boldsymbol{x}^*$ does not contain any previously selected cluster centers.

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 $\boldsymbol{X}$. For the edge set $E(\boldsymbol{S})$, only edges with length less than $r_k(\boldsymbol{x}^*)$ are retained in the subgraph $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E_{\boldsymbol{x}^*})$. The component sets obtained from the graph $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E_{\boldsymbol{x}^*})$ are assessed, and if the component set containing $\boldsymbol{x}^*$ does not contain previously selected candidate centers, then $\boldsymbol{x}^*$ is accepted. The point of difference between the CPF and DCF center selection methods is that the edges of the graph are only by CPF if their distance is below the threshold value, $r_k(\boldsymbol{x}^*)$. This approach allows the graph to better reflect the scale of the data contained in the the component set. This edge removal is similar to methods used to recover level sets using k-NN graphs (Chaudhuri and Dasgupta, 2010; Chaudhuri et al., 2014).

Varying the parameter ρ determines the number of clusters for each component set $\boldsymbol{S}$. For low values of ρ, fewer vertices will be removed. As a result, it is less likely that a proposed center will be disconnected from existing centers. For larger values of ρ, more vertices and their edges will be removed from the graph. The probability of the proposed center being disconnected from previously detected cluster centers will increase. It is not required to have different ρ values for different component sets, because the two cutting thresholds, $\rho^{-\frac{1}{p}} r_k(\boldsymbol{x}^*)$ for vertex and $r_k(\boldsymbol{x}^*)$ for edge, adapt naturally to the density level of the component set being assessed.

5.4.3 The CPF Algorithm

The CPF 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 no edges are marked as outliers and removed. The remaining data

Algorithm 3: The Component-wise Peak-Finding Algorithm

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

Compute $G(\boldsymbol{X}, E)$, the mutual k-nearest neighbor graph.
Extract $\mathcal{S}$, the set of component sets from $G(\boldsymbol{X}, E)$.
for each $\boldsymbol{S} \in \mathcal{S}$ **do**
    Sort the $\boldsymbol{x}$'s according to their γ values.
    Let $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \boldsymbol{S}} \gamma(\boldsymbol{x})$.
    Initialise $\widehat{\mathcal{M}} = \{\boldsymbol{x}^*\}$, the set of true centers in $\boldsymbol{S}$.
    loop
        Let $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \boldsymbol{S}} \{\gamma(\boldsymbol{x}) : \boldsymbol{x} \notin \widehat{\mathcal{M}}\}$.
        Let $\boldsymbol{V}_{\boldsymbol{x}^*} = \{\boldsymbol{x} \in \boldsymbol{S} : r_k(\boldsymbol{x}) < \frac{r_k(\boldsymbol{x}^*)}{\rho^{1/p}}\}$.
        Let $E_{\boldsymbol{x}^*} = \{\{\boldsymbol{x}_i, \boldsymbol{x}_j\} \in E(\boldsymbol{S}) : \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq r_k(\boldsymbol{x}^*)\}$.
        Let $\boldsymbol{S}_\rho(\boldsymbol{x}^*) \subseteq \boldsymbol{S}$ be the component set of the graph $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E_{\boldsymbol{x}^*})$ containing $\boldsymbol{x}^*$.
        if $\boldsymbol{S}_\rho(\boldsymbol{x}^*) \cap \widehat{\mathcal{M}} = \varnothing$ **then**
            Add $\boldsymbol{x}^*$ to $\widehat{\mathcal{M}}$.
        end if
    end loop
    Initialise $\vec{G}(\boldsymbol{S}, \vec{E})$, a directed graph with $\boldsymbol{S}$ as vertices and no edges, $\vec{E} = \varnothing$.
    for each $\boldsymbol{x}$ in $\boldsymbol{S} \backslash \widehat{\mathcal{M}}$ **do**
        Add a directed edge from $\boldsymbol{x}$ to $b(\boldsymbol{x})$.
    end for
    for each cluster center $\boldsymbol{x} \in \widehat{\mathcal{M}}$ **do**
        Let $\boldsymbol{C}$ be the collection of the points connected by any directed path in $\vec{G}(\boldsymbol{S}, \vec{E})$ that terminates at $\boldsymbol{x}$.
        Add $\boldsymbol{C} \cup \boldsymbol{x}$ to $\widehat{\mathcal{C}}$.
    end for
end for
return $\widehat{\mathcal{C}}$

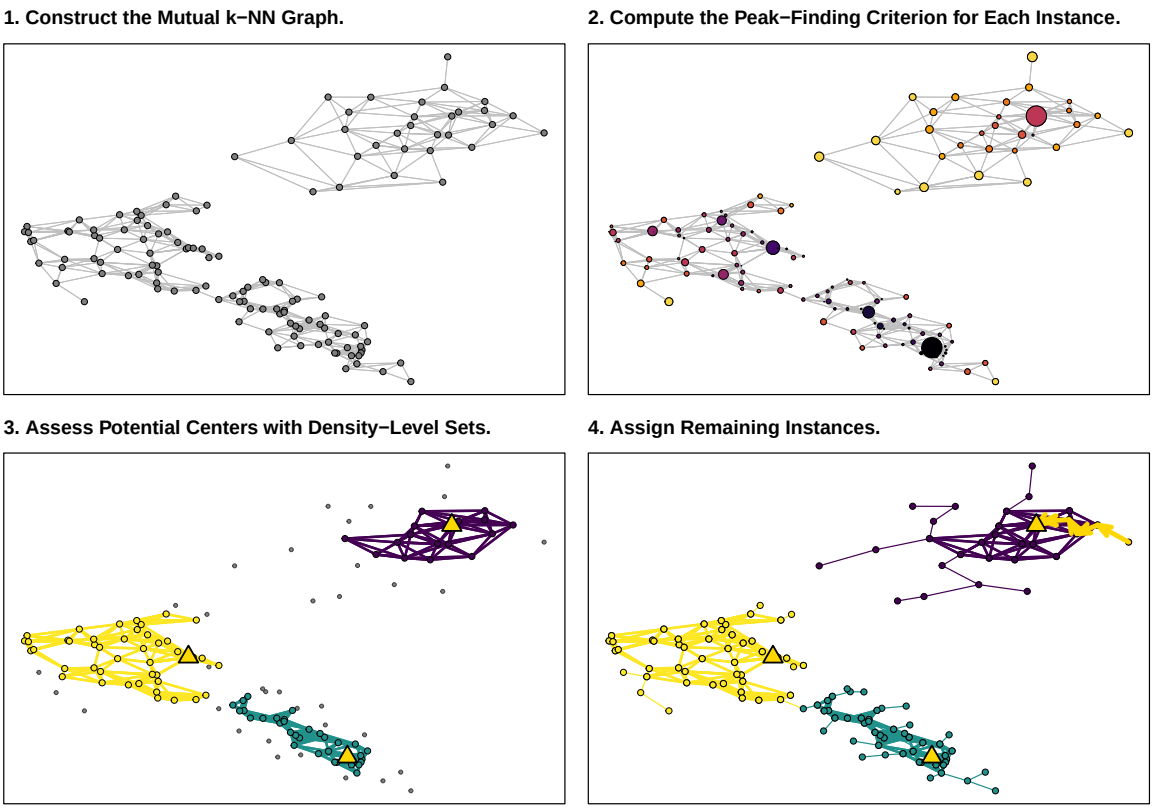


Figure 5.3: Illustration of the proposed Component-wise Peak-Finding algorithm. 1. The mutual k-NN graph is constructed, from which we extract two component sets. 2. Densities are computed as the inverse of the distance from an instance to its k-th nearest neighbor (darker color represents higher density). The distance from each instance to its nearest neighbor of higher density is found (larger point represents larger distance to point of higher density). The peak-finding criterion is the product of these two quantities. 3. For each connected component, density-level sets are used to assess potential cluster centers, shown in yellow. 4. Non-center instances are assigned to the same cluster as their nearest neighbor of higher local density. A sample assignment path is shown in gold for the purple cluster.

is partitioned into disjoint component sets according to the graph $G(\boldsymbol{X}, E)$ yielding $\mathcal{S} = \{\boldsymbol{S}_1, \ldots, \boldsymbol{S}_{n_S}\}$ (Lines 1-3).

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. The point $\boldsymbol{x}^*$ is automatically accepted as a cluster center, and the set of true centers for the component set $\boldsymbol{S}$ is initialised as $\widehat{\mathcal{M}} = \{\boldsymbol{x}^*\}$ (Lines 5-7).

Next, the instance with maximal value of the peak-finding criterion yet to be assessed is selected and denoted by $\boldsymbol{x}^*$. The subgraph $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E_{\boldsymbol{x}^*})$ is extracted, and the com-

ponent set of $G(\boldsymbol{V}_{\boldsymbol{x}^*}, E_{\boldsymbol{x}^*})$ containing $\boldsymbol{x}^*$ is denoted by $\boldsymbol{S}_\rho(\boldsymbol{x}^*)$ (Lines 9-12). If $\boldsymbol{S}_\rho(\boldsymbol{x}^*)$ is disjoint from all selected cluster centers in $\widehat{\mathcal{M}}$, then $\boldsymbol{x}^*$ is added to $\widehat{\mathcal{M}}$ (Lines 13-15).

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 19-22). 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 23-26). The process is repeated for each component set to return the final set of clusters $\widehat{\mathcal{C}}$. The method is further explained in Figure 5.3.

5.5 Analysis of CPF

The CPF algorithm combines density-level set estimation with k-NN graphs, and a density peaks-based clustering algorithm which assesses estimated modes using a pruning procedure similar to that of DCF. As such, the theoretical results relevant to CPF have been covered previously in this thesis and, thus, are not repeated. For results demonstrating the consistency of the density-level set estimates recovered by k-NN graph, consider those given in Maier et al. (2009) and Kpotufe and von Luxburg (2011) and further work aimed at understanding of the optimal choice of k, particularly the analysis of Maier et al. (2009) and Steinwart (2011). The results regarding the mode selection procedure are equivalent to those given in Section 4.5, applied instead to each component set with the scaling parameter $\rho = 1 - \beta$. The result proving the quality of the cluster assignment of Section 3.3 can also be applied to each component set, with suitable adjustments made to the number of observations in each component.

5.5.1 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

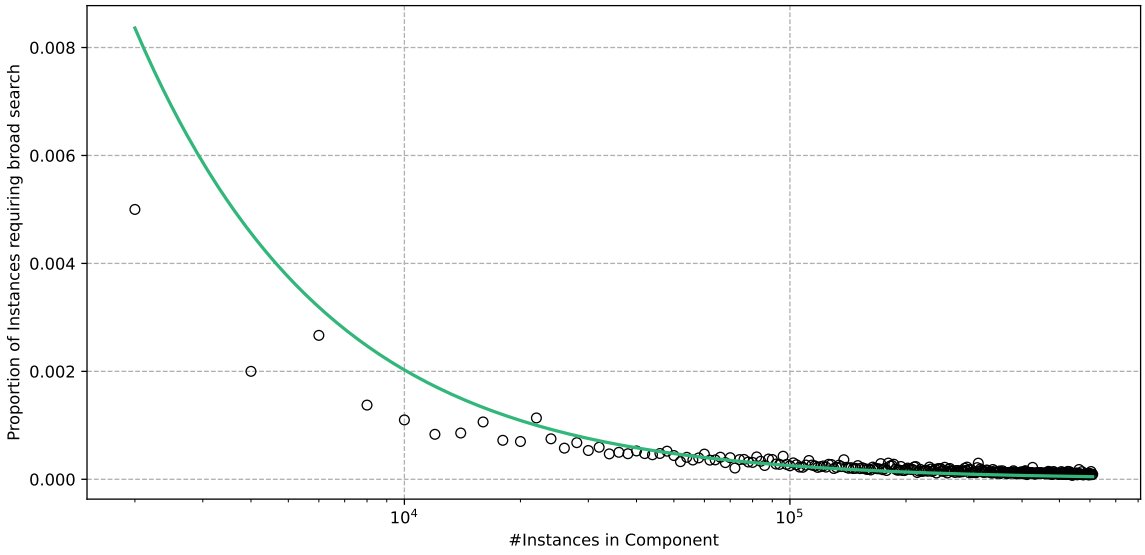


Figure 5.4: Analysis of the proportion of instances that do not have a point of higher density in their k nearest neighbors. Data are generated from Gaussian components according to the process in Section 4.5. The points in black are $(|\boldsymbol{S}|, p)$ for a given component with the green line showing the function $2.2\log(|\boldsymbol{S}|)/|\boldsymbol{S}|$.

$O(n)$ 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 a component set. Experimental results for the proportion of instances without a point of higher density in their neighbors are presented in Figure 5.4. The line in the figure is $2.2\log(|\boldsymbol{S}|)/|\boldsymbol{S}|$. 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.

5.5.2 Simulated Experiments

Considering again the example of Figure 5.2, while CPF correctly recovers the two clusters in the data, the same result could be obtained using only the mutual k-NN graph. The peak-finding element of the CPF algorithm does not contribute to the clustering results. In this section, we provide a brief simulated analysis to demonstrate

that the performance of CPF is enhanced by both the mutual k-NN graph and the peak-finding method. To do this, we assess the clustering performance in a brief ablation study via simulated data.

As in Section 4.5, we generate multiple synthetic datasets with different levels of separation and density to assess their impact on the clustering results. Gaussian components are sampled so that their means satisfy the following inequality:

$$\forall_{i \neq j} : \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\| \geq c \operatorname{tr}(\boldsymbol{I}),$$

where c characterizes the degree of separation of the clusters, and $\boldsymbol{I}$ is the identity matrix. For the purposes of this analysis, all component covariances are set to the identity matrix, $\boldsymbol{\Sigma} = \boldsymbol{I}$.

We generate mixtures of five components with size $n = 3000$ and dimension $p = 2$. To assess the impact of separation, eight degrees of separation are assessed, ranging from $c = 0$ (low) to $c = 4.0$ (high) in increments of 0.5. As the covariances of the Gaussian components are identical, to generate clusters of varying density, three prior distributions are used: (1) common prior $\boldsymbol{\pi} = \{0.2, 0.2, 0.2, 0.2, 0.2\}$; (2) medium prior variation $\boldsymbol{\pi} = \{0.4, 0.2, 0.2, 0.1, 0.1\}$; and (3) high prior variation $\boldsymbol{\pi} = \{0.6, 0.1, 0.1, 0.1, 0.1\}$. For each configuration, we generate 10 datasets.

We here assess the clustering performance offered by the constituent parts of the CPF method: (1) level set clustering using mutual k-NN graphs; and (2) mode-seeking clustering, using the peak-finding criterion and the CPF center selection method outlined above. We compare the performance of the level set and mode-seeking steps with the performance of CPF. The first comparison method, henceforth referred to as mutual k-NN clustering, consists of Lines 1-3 of Algorithm 3. Clusters are taken to be the component sets extracted from the graph $G(\boldsymbol{X}, E)$. The second comparison method applies the method developed in Lines 4-28 of Algorithm 3, however rather than applying the method to a component set $\boldsymbol{S}$, the method is applied to the entire dataset $\boldsymbol{X}$.

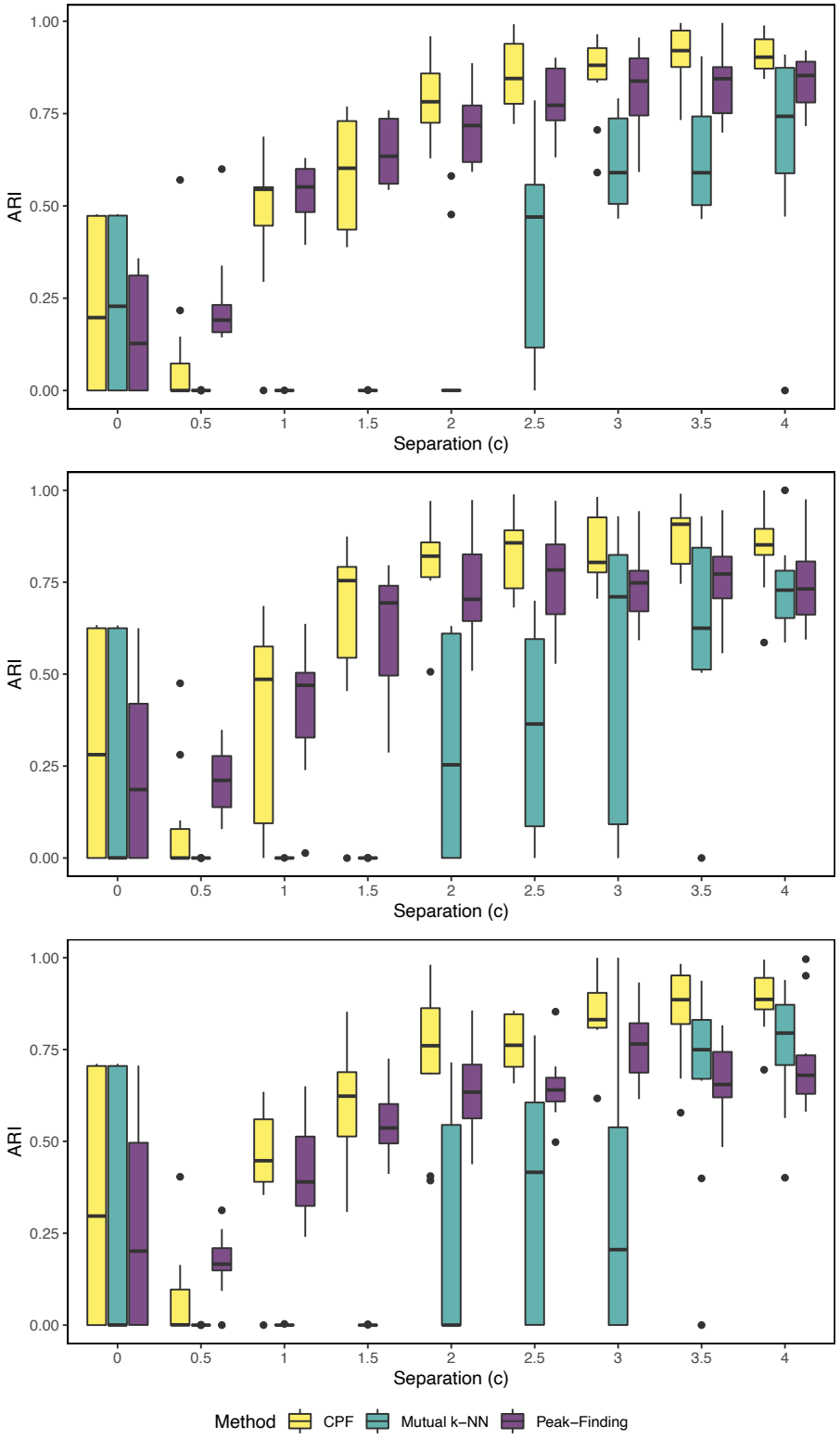


Figure 5.5: ARI values of the clustering results returned by CPF and constituent methods on synthetic datasets. Top: Datasets with prior $\boldsymbol{\pi} = \{0.2, 0.2, 0.2, 0.2, 0.2\}$. Middle: Datasets with prior $\boldsymbol{\pi} = \{0.4, 0.2, 0.2, 0.1, 0.1\}$. Bottom: Datasets with prior $\boldsymbol{\pi} = \{0.6, 0.1, 0.1, 0.1, 0.1\}$.

In Figure 5.5, the results of CPF and the constituent methods are shown. We assess the clustering performance using the ARI and the AMI introduced previously. The three figures correspond to the three levels of density variation. The box plots are grouped by the value of the separation parameter c. It is clear that CPF consistently outperforms both of its constituent methods. This effect is enhanced as the clusters become further separated and is observed across different formulations of the density variation. Mutual k-NN clustering performs poorly when clusters are not well separated and outperforms the peak-finding clustering as the separation increases, particularly for the highly separated data and data with varying densities. This analysis indicates that level set and mode-seeking non-parametric clustering methods can be mutually complementary. The performance of CPF is amplified through effective deployment of each formulation at different stages of the clustering procedure.

5.5.3 Real-World Experiments

In this section, we extend the comparison of CPF with its constituent methods to the real-world datasets introduced in Section 4.5. CPF is motivated by the potential improvements to clustering performance to be gained from combining density-level set and mode-seeking methods. By applying the peak-finding method within each level set of the density, CPF aims to reinforce the assignment mechanism of the density peaks clustering algorithm. To quantify the clustering performance and algorithmic efficiency of CPF, it is evaluated against the Mutual k-NN method and the peak-finding method detailed above. As before, all experiments have been conducted on a PC running Debian 10 (Buster), consisting of 24 cores and 24GB of RAM. Our method is implemented in Python; its source code, and code to reproduce the below experiments is available online.[1]

[1] https://github.com/tobinjo96/Thesis-Experiments/

Dataset	Metric	CPF	MKN	PKF
Dermatology	ARI	**0.80**	0.66	0.77
	AMI	**0.83**	0.76	**0.83**
Ecoli	ARI	**0.70**	0.12	**0.70**
	AMI	**0.66**	0.34	**0.66**
Glass	ARI	**0.29**	**0.29**	0.18
	AMI	**0.41**	**0.41**	0.27
Letter Recognition	ARI	**0.19**	0.00	0.16
	AMI	**0.56**	0.18	0.49
Page Blocks	ARI	**0.48**	0.00	**0.48**
	AMI	**0.32**	0.00	**0.32**
Phonemes	ARI	**0.75**	0.00	0.67
	AMI	**0.81**	0.00	0.79

Table 5.1: Quality of clusterings for the real-world datasets.

Dataset	CPF	MKN	PKF
Derm.	0.19	**0.02**	0.16
Ecoli	0.16	**0.01**	0.09
Glass	0.08	**0.01**	0.05
Letter R.	24.00	**9.85**	24.58
Page B.	1.44	**0.91**	1.48
Phonemes	21.13	**19.01**	31.86

Table 5.2: Average run time of the assessed clustering methods for the real-world datasets.

Experimental Set-Up

CPF is assessed on the same datasets used in the analysis of DCF in the previous chapter. Details of these datasets can be found in Table 4.1. Instances with missing values are removed. To assess the performance of CPF on these datasets, we apply the mutual k-NN clustering method (MKN) and the peak-finding clustering methods (PKF), both implemented in Python.

Results

The results of the clustering are presented in Table 5.1. As in the simulated analysis, CPF clearly outperforms the two constituent methods in terms of both metrics. CPF achieves the highest score for both metrics for every dataset analyzed. The benefits in performance offered by the CPF method are clear. Interpreting the results of the constituent methods allows a deeper understanding of the quality of the results of CPF. For the Glass dataset, the clustering returned by the CPF method is equal to that returned by Mutual k-NN. The center selection method of CPF does not detect multiple cluster centers in any component, and hence returns the level set clustering. In this case, it significantly outperforms the clustering of the peak-finding method. For the Page Blocks and Phonemes datasets, the level set method locates only one connected component of the density. This leads to the clusterings for the CPF and peak-finding methods being similar, with CPF outperforming the peak-finding method for the Phonemes dataset due to the removal of a small number of outlying points. For the Dermatology and Letter Recognition datasets, the combination of the level set and peak-finding approaches leads to clustering results that outperform either constituent method, reaffirming the benefits available through combination of these two methodologies.

The average run time, in seconds, for the three methods in Table 5.2. CPF is slow than the competitor approaches for the small datasets, understandable as it involves the execution of both alternate approaches. The difference in execution time is unlikely to be a major impediment to the use of CPF in applications, however. For the larger datasets, CPF executes faster than the peak-finding method. This is the result of the broad search, discussed in the complexity analysis in Section 5.5.1, taking place over a component set, rather than the entire dataset as for the peak-finding method. This indicates that CPF will scale well for large scale applications.

Further experimental analysis, as well as a thorough analysis of the parameter space for CPF is provided in Chapter 6.

5.6 Application

In this section, we introduce an adapted version of CPF for multi-image matching, demonstrating its adaptability and performance for non-standard clustering problems. Multi-image matching is an important application in modern computer vision, notably in the reconstruction of 3-D scenes form 2-D images. We can consider the multi-image matching problem as a special case of constrained clustering. Constrained clustering extends clustering from an unsupervised method to a semi-supervised method. For a set of instances, the clustering algorithm is supervised by a set of pairwise constraints indicating pairs of instances which must or cannot be clustered together. 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 (Tron et al., 2017) 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. QuickMatch achieves exceptional speed for the multi-image matching problem. Other clustering approaches have been applied to this problem, including a novel context-specific iterative algorithm (Yan et al., 2016); however, high execution time has inhibited their application. Methods that do not formulate multi-image matching as a clustering problem are outside the scope of this work, and a review can be found in Ma et al. (2021).

We adapt the CPF method introduced in Algorithm 3 to accommodate this 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 3, substituting lines 19-22 with Algorithm 4. CPF-Match modifies the allocation procedure of CPF, while component sets and cluster centers are selected in the same way.

Algorithm 4: CPF-Match

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

Again, a directed graph $\vec{G}(\boldsymbol{S}, \vec{E})$ is initialized as before (Line 19). Next, CPF-Match sorts the non-center points of $\boldsymbol{S}$ according to the distance $\|\boldsymbol{x} - b(\boldsymbol{x})\|$, from smallest to largest (Line 20). 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 21-25).

To demonstrate the ability of CPF-Match to perform multi-image matching, we apply it to the Graffiti dataset.[2] 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 (Lowe, 1999). To each set of features, we apply the CPF-Match algorithm. The results of CPF-Match for a pair of images from each of the six image groups are presented in Figure 5.6.

For evaluation, we apply the same approach as in Tron et al. (2017). 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 then 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 image group, we use all detected feature points in the first image as test points and evaluate the matches from the first image to the other five images. The performance curves for CPF-Match and

[2] https://cvssp.org/featurespace/web/related_papers/graffiti.html

Figure 5.6: One pair of images from each of the six image groups (bark, bikes, boat, graffiti, Leuven, and UBC). Lines between each pair of images indicate a match detected by CPF-Match.

QuickMatch for each of the six datasets are presented in Figure 5.7.

The ability of CPF-Match to detect common points between multiple images is demonstrated clearly. 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. Furthermore, it is clear that CPF is an adaptable clustering framework.

5.7 Conclusion and Future Work

This chapter introduced CPF, a clustering algorithm that combines the benefits of both density-level set and mode-seeking density-based clustering methods. We showed, in a simulated analysis, that both formulations are complementary when combined correctly, and better handle key features of contemporary datasets than either approach alone. These results were reinforced in an experimental analysis of 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.

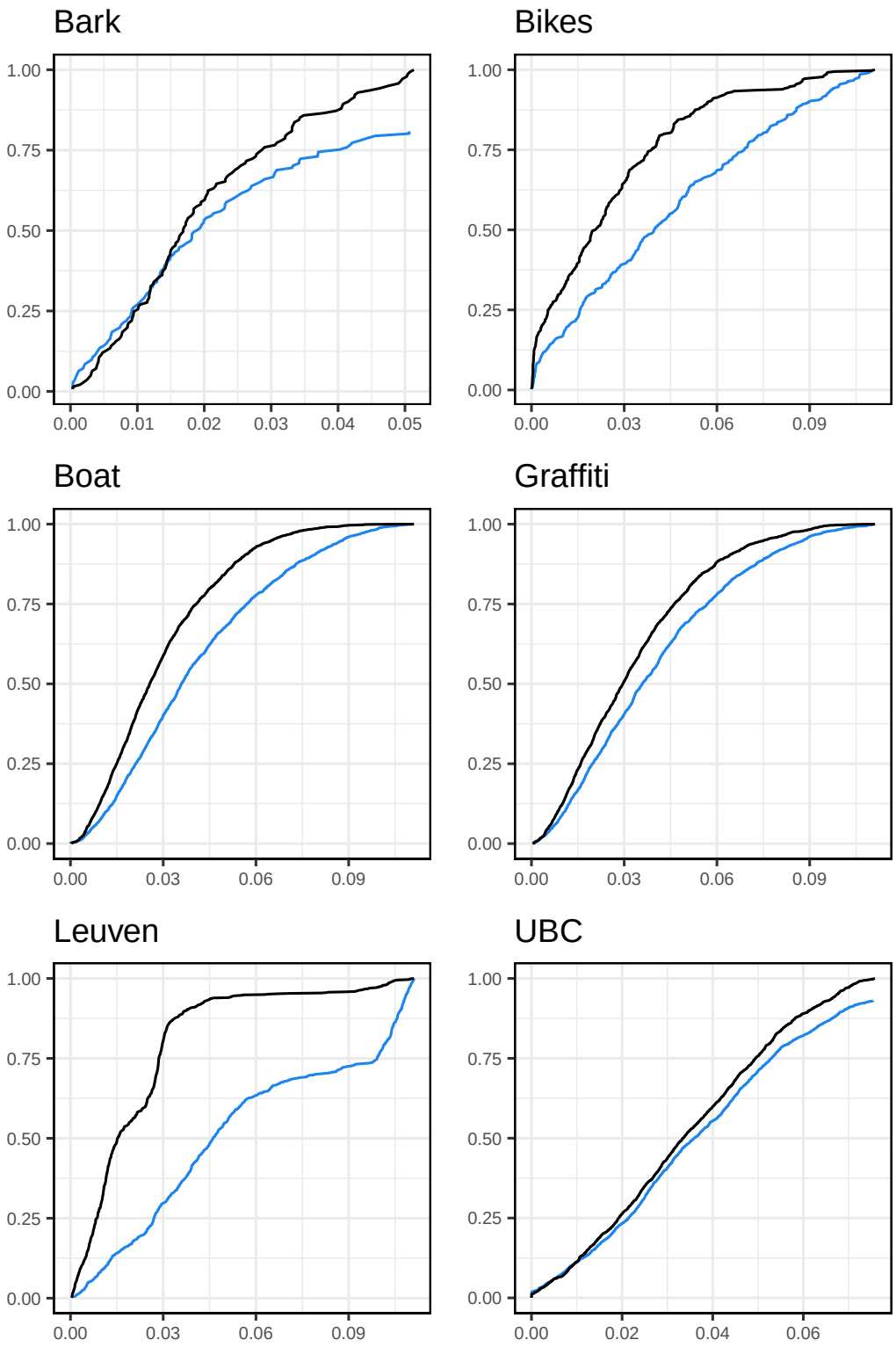


Figure 5.7: The performance curves for the CPF-Match (black) and QuickMatch (blue) multi-image matching methods on the Graffiti datasets. The y-axis shows the percentage of correct matches and the x-axis shows the distance threshold divided by the image width (in pixels). For all datasets, $k = 10$, $\rho = 0.5$ and the threshold parameter of QuickMatch is set to 4.

6 Experimental Comparison

6.1 Summary

In this chapter, an extensive experimental evaluation of the DCF and CPF methods is completed. The two novel approaches are compared to a broad range of prominent density-based clustering methods. DCF, CPF, and the competitor methods are assessed on ten real-world clustering datasets. The quality of the clusterings is reported, measured using two widely used validation indices, as well as the execution time of each method. As well as comparing the optimal clustering results for each method, an analysis of the effect of hyper-parameter selection on the clustering results is undertaken.

6.2 Introduction

In the previous two sections, novel methods aimed at improving and extending the density peaks clustering algorithm have been proposed. The DCF method of Chapter 4 models high density regions of the data using cluster cores rather than point modes, thus reducing the likelihood of incorrectly detecting point modes in the data due to noise in the density estimate. DCF was shown experimentally to outperform the density peaks clustering algorithm with the k-NN density estimator in terms of clustering performance, and achieve similar results to a related method, QuickShift++, with more efficient computation. In Chapter 5, the CPF algorithm was introduced. CPF combines density-level set approaches with the a peak-finding cluster algorithm

with the aim of removing the situations where the sample-based allocation mechanism of density peaks methods incorrectly assigns instances across regions of very low density in the data. This approach was shown to be superior to clustering using with density-level set or peak-finding clustering alone.

While the analyses in the previous two chapters indicate that the methods developed achieve the goal of improving the density peaks clustering algorithm, their superiority over a broad range of non-parametric density-based clustering methods remains to be proved. As such, in this section, we provide a detailed experimental analysis using well-known competitor methods over a broad range of real-world datasets. Furthermore, the parameters used by each algorithm are introduced and the sensitivity of the clustering results to the choice of parameters is assessed. It is demonstrated that both DCF and CPF, as well as achieving superb results for the best clusterings, return high quality outputs over a broad range of parameters.

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

6.3 Experimental Set-Up

We assess DCF and CPF on an expanded pool of ten real-world datasets. Details of the datasets can be found in Table 6.1. As before, instances with missing values are removed.

To evaluate the clusterings produced we again use the ARI and the AMI. For both metrics, a larger value indicates a higher-quality clustering.

To assess the performance of the methods proposed in this thesis on the ten real-world datasets, we apply the following competitor non-parametric density-based clustering algorithms:

- Density Peaks Clustering (DPC) method with k-NN density estimator introduced in Chapter 3 and implemented in Python. This method takes the true number

of clusters, m as an input. The instances with the top m values of the peak-finding criterion are selected. This approach has one parameter, k, the number of neighbors used in the density estimate.

- The original Density Peaks Clustering (ODP) method of Rodriguez and Laio (2014) implemented in R in the densityClust library. This method takes the true number of clusters, m as an input. The instances with the top m values of the peak-finding criterion are selected. This approach has one parameter, d_c, the threshold distance used in the density estimate.

- Adaptive Density Peaks Clustering (ADP) (Wang and Xu, 2017) implemented in R in the ADPclust library. This method take the true number of clusters as an input, and uses quantiles of the density estimates and distance to a neighbor of higher density to determine the cluster centers. The approach has one parameter, h, used to tune a KDE.

- Comparative Density Peaks Clustering (CDP) (Li and Tang, 2018) implemented in Matlab.[1] This method take the true number of clusters as an input, and uses a modified version of the peak-finding criterion that accounts for low density points to determine the cluster centers. The approach has one parameter, d_c, the threshold distance used in the density estimate.

- DBSCAN (DBS) (Ester et al., 1996) implemented in Python and C++ in the SciKit library (Pedregosa et al., 2011). This approach determines the number of clusters automatically. DBSCAN has two parameters, (1) a primary parameter, *eps*, capturing the radius used to construct the neighborhood graph and (2) a secondary parameter, $minPts$, that controls the minimum size of a cluster.

- HDBSCAN (HDB) (Campello et al., 2013) implemented in Python and C++ in the hdbscan library. This approach determines the number of clusters automatically. HDBSCAN has one parameter, $minPts$, that controls the minimum size of a cluster.

[1] https://github.com/ZejianLi/ComparativeDensityPeaks

Source	Name	n	p	m
Dua and Graff (2019)	Dermatology	358	34	6
Dua and Graff (2019)	Ecoli	336	7	8
Dua and Graff (2019)	Glass	214	9	6
Dua and Graff (2019)	Letter Recognition	20000	16	26
Dua and Graff (2019)	Optdigits	5620	64	10
Dua and Graff (2019)	Page Blocks	5743	10	5
Dua and Graff (2019)	Pendigits	10992	16	10
Hastie et al. (2009)	Phonemes	4509	256	5
Dua and Graff (2019)	Seeds	210	7	3
Dua and Graff (2019)	Vertebral	310	6	3

Table 6.1: Characteristics of the real-world datasets.

- Mean Shift (MNS) (Cheng, 1995; Comaniciu and Peter, 2002) implemented in Python and C++ in the SciKit library (Pedregosa et al., 2011). Mean shift returns the number of clusters automatically. Mean shift has one parameter, h, the bandwidth of the KDE.

- Quick Shift (QKS) (Vedaldi and Soatto, 2008) implemented in Python.[2] Quick shift determines the number of clusters automatically. There is one parameter, τ, the threshold distance used to estimate modes.

Of these algorithms, 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.

6.3.1 Results

Results for clustering ten real-world datasets are presented in Table 6.2. For each method we present the clustering with the highest average value of ARI and AMI. DCF achieves the best clustering, in terms of the ARI, for five of the datasets assessed, and the best clustering, in terms of the AMI, for six of the datasets assessed. CPF achieves the best clustering, in terms of the ARI, for three of the datasets assessed,

[2]https://github.com/Nick-Ol/MedoidShift-and-QuickShift

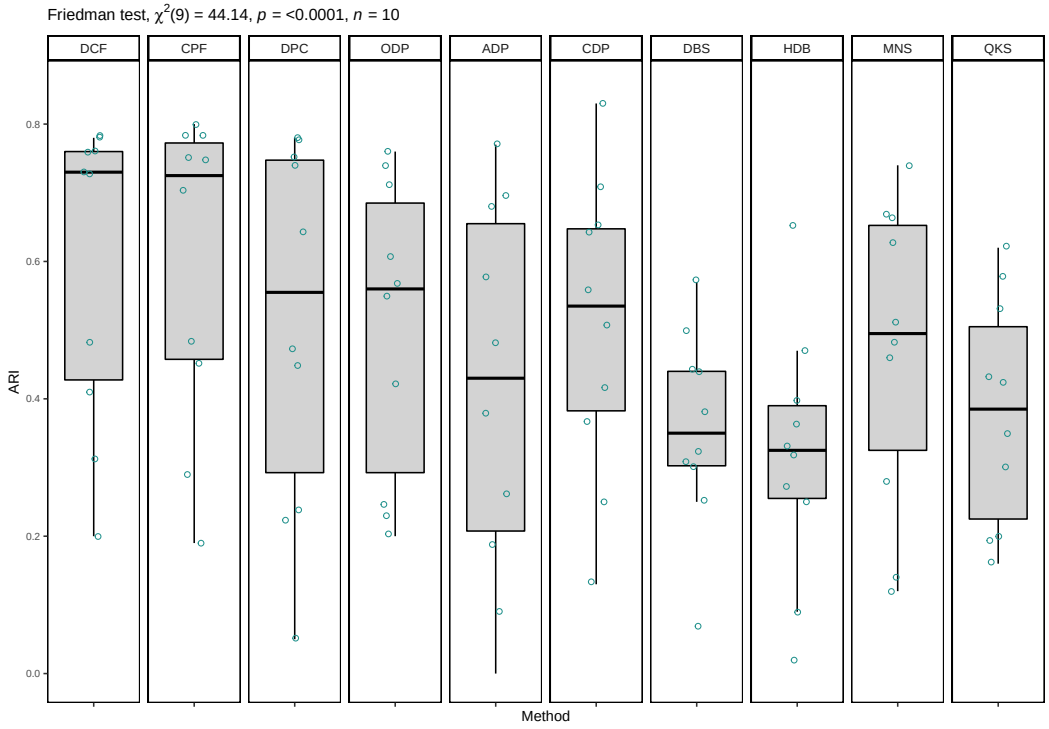
and the best clustering, in terms of the AMI, for four of the datasets assessed. Both methods significantly outperform all of the competitor methods, with only the original density peaks clustering method achieving an optimal score for either metric on more than two datasets. Also presented in Table 6.2 are the mean rankings for the quality of the clusterings returned by each of the methods for both metrics. Here, DCF is seen to have the best performance overall, with a mean ranking for both metrics. CPF achieves similar ranking scores, indicating that the clustering results are both generally of high quality. The clusterings returned by the DCF algorithm are one of the top three as measured with the ARI for nine of the ten datasets, and for all ten of the datasets when the AMI is used to assess their quality. For CPF, the clusterings returned are one of the top three for each of the datasets when the ARI is used for assessment, and for nine of the ten datasets, when the AMI is used to assess the clustering quality.

In terms of the ARI, the methods with the joint next highest rank is the original density peaks clustering algorithm and the comparative density peaks clustering method. In terms of the AMI, the density peaks clustering algorithm, as formulated in Section 3.2 is the best performing approach. Taken together, this makes a strong case for the ability of the peak-finding criterion to detect meaningful clusters in the data. It should be noted that the performance of the density peaks based methods is achieved with the true number of clusters specified as an input, a feature that is unlikely to be the case in many applications.

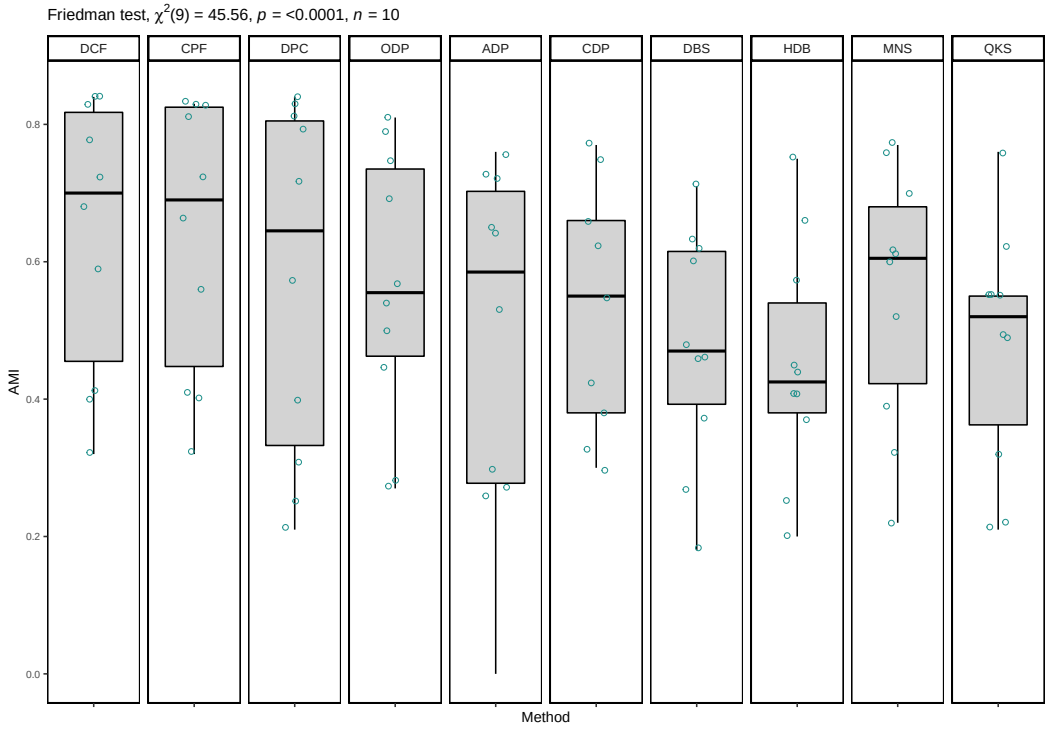
Considering the competitor approaches that determine the number of clusters automatically, the performance is significantly worse than DCF and CPF. The level set methods DBSCAN and HDBSCAN perform poorly. The poor performance in both metrics indicates that these methods fail to capture the classes present in the data. Mean shift achieves the optimal clustering for two datasets, Ecoli and Page Blocks, but does not consistently return high quality clusterings. The results for the Vertebral dataset is the worst of all methods assessed. Quick shift also does not return high quality clusterings, particularly as assessed using ARI. As ARI significantly penalizes false positive clusters, it can again be concluded that quick shift is not adequately de-

Dataset	Metric	DCF	CPF	DPC	ODP	ADP	CDP	DBS	HDB	MNS	QKS
Dermatology	ARI	0.73	**0.80**	0.74	0.25	0.58	0.71	0.44	0.47	0.66	0.35
	AMI	0.78	**0.83**	**0.83**	0.45	0.73	0.75	0.63	0.66	0.77	0.49
Ecoli	ARI	0.73	0.70	0.47	0.55	0.68	0.51	0.50	0.40	**0.74**	0.42
	AMI	0.68	0.66	0.57	0.50	0.65	0.55	0.48	0.41	**0.70**	0.49
Glass	ARI	**0.31**	0.29	0.24	0.20	0.26	0.25	0.25	0.25	0.28	0.20
	AMI	**0.41**	**0.41**	0.31	0.27	0.27	0.38	0.37	0.37	0.39	0.32
Letter Recognition	ARI	0.20	0.19	0.05	**0.23**	0.09	0.13	0.07	0.02	0.14	0.16
	AMI	**0.59**	0.56	0.25	0.54	0.53	0.42	0.46	0.45	0.52	0.55
Optdigits	ARI	0.78	0.78	0.78	0.74	0.00	**0.83**	0.30	0.09	0.51	0.53
	AMI	0.84	0.83	0.84	0.79	0.00	**0.86**	0.60	0.44	0.61	0.55
Page Blocks	ARI	**0.48**	**0.48**	0.22	0.42	0.38	0.42	0.32	0.33	**0.48**	0.30
	AMI	**0.32**	**0.32**	0.21	0.28	0.26	0.30	0.18	0.20	**0.32**	0.22
Pendigits	ARI	**0.76**	0.75	0.64	0.61	0.48	0.64	0.57	0.65	0.67	0.58
	AMI	**0.84**	0.83	0.79	0.75	0.64	0.77	0.71	0.75	0.76	0.76
Phonemes	ARI	**0.76**	0.75	0.75	**0.76**	0.70	0.56	0.44	0.36	0.46	0.43
	AMI	**0.83**	0.81	0.81	0.81	0.76	0.66	0.62	0.57	0.60	0.55
Seeds	ARI	**0.78**	**0.78**	**0.78**	0.71	0.77	0.65	0.38	0.32	0.63	0.62
	AMI	**0.72**	**0.72**	**0.72**	0.69	**0.72**	0.62	0.46	0.41	0.62	0.62
Vertebral	ARI	0.41	0.45	0.45	**0.57**	0.19	0.37	0.31	0.27	0.12	0.19
	AMI	0.40	0.40	0.40	**0.57**	0.30	0.33	0.27	0.25	0.22	0.21
Mean Ranking	**ARI**	**1.8**	**2.0**	**5.0**	**4.8**	**6.4**	**4.8**	**7.6**	**7.9**	**4.9**	**7.9**
	AMI	**1.5**	**1.9**	**4.2**	**5.5**	**6.2**	**5.0**	**7.8**	**8.2**	**4.9**	**7.3**

Table 6.2: The quality of the clusterings for the real-world datasets. The best results are highlighted in bold. The mean ranks summarize the overall performance of the algorithms.



(a) ARI



(b) AMI

Figure 6.1: Boxplots of the observed (a) ARI and (b) AMI for each of the methods assessed. The results of a Friedman test to detect differences across multiple clustering attempts are presented at the top of each table.

	DCF	**CPF**	**DPC**	**ODP**	**ADP**	**CDP**	**DBS**	**HDB**	**MNS**	**QKS**
DCF	1.000	-	-	-	-	-	-	-	-	-
CPF	0.975	1.000	-	-	-	-	-	-	-	-
DPC	0.128	**0.085**	1.000	-	-	-	-	-	-	-
ODP	0.162	0.145	0.825	1.000	-	-	-	-	-	-
ADP	**0.011***	**0.011***	0.521	0.548	1.000	-	-	-	-	-
CDP	**0.046***	**0.044***	1.000	0.548	0.555	1.000	-	-	-	-
DBS	**0.011***	**0.011***	0.169	0.110	0.527	0.110	1.000	-	-	-
HDB	**0.011***	**0.011***	0.113	0.825	0.413	**0.045***	0.241	1.000	-	-
MNS	**0.045***	**0.053**	0.617	0.714	0.615	0.714	**0.072**	**0.072**	1.000	-
QKS	**0.011***	**0.011***	0.126	**0.082**	0.548	**0.020***	0.981	0.763	0.145	1.000

Table 6.3: P-values for Wilcoxon signed-rank tests, comparing the ARI values for each of the methods assessed on each dataset. The values are adjusted using the Benjamini-Hochberg adjustment for multiple comparisons. Significance at the $\alpha = 10\%$ level is denoted in bold and significance at the $\alpha = 5\%$ level is denoted with an asterisk.

	DCF	**CPF**	**DPC**	**ODP**	**ADP**	**CDP**	**DBS**	**HDB**	**MNS**	**QKS**
DCF	1.000	-	-	-	-	-	-	-	-	-
CPF	0.562	1.000	-	-	-	-	-	-	-	-
DPC	0.141	0.140	1.000	-	-	-	-	-	-	-
ODP	0.132	0.175	0.965	1.000	-	-	-	-	-	-
ADP	**0.037***	**0.037***	0.498	0.677	1.000	-	-	-	-	-
CDP	**0.037***	**0.037***	0.866	0.981	0.809	1.000	-	-	-	-
DBS	**0.037***	**0.022***	0.158	0.158	0.498	**0.022***	1.000	-	-	-
HDB	**0.037***	**0.022***	0.158	0.210	0.677	**0.037***	0.331	1.000	-	-
MNS	**0.053**	**0.053**	0.866	0.929	0.677	1.000	**0.096**	**0.037***	1.000	-
QKS	**0.022***	**0.022***	0.210	0.246	0.651	0.158	0.965	0.651	**0.084**	1.000

Table 6.4: P-values for Wilcoxon signed-rank tests, comparing the AMI values for each of the methods assessed on each dataset. The values are adjusted using the Benjamini-Hochberg adjustment for multiple comparisons. Significance at the $\alpha = 10\%$ level is denoted in bold and significance at the $\alpha = 5\%$ level is denoted with an asterisk.

tecting the true number of clusters in the data. Considering the significant similarities between the methodology of quick shift and that of the density peaks clustering methods, the poor results are likely the result of difficulty in finding the optimal value of the parameter h. The impact of hyper-parameter selection for all methods is analyzed in greater detail in subsequent sections.

Following the guidance given in Demšar (2006) and Garcia and Herrera (2008), the results are also subjected to a statistical analysis using non-parametric tests. Firstly, a Friedman test is applied (Friedman, 1937). A Friedman test is non-parametric equivalent of a repeated-measures ANOVA, testing the null hypothesis that the methods assessed are equivalent. The results for the Friedman tests for (a) the ARI values and (b) the AMI values are shown in Figure 6.1. For both tests the p-value is less than 1% indicating a high level of significance. Secondly, we apply the Wilcoxon signed-rank test for pairwise comparisons, using the Benjamini-Hochberg correction to control the false-discovery rate (Wilcoxon, 1945; Benjamini and Hochberg, 1995). The p-values for the associated comparison are show for the ARI values in Table 6.3 and for the AMI values in Table 6.4. The results indicate a strong level of statistical significance for the improved clustering quality for the DCF and CPF method. The novel approaches introduced in this work significantly outperform all bar one of the methods assessed and are not outperformed by any of the competitor approaches.

The average run time, in seconds, for each method is presented in Table 6.5. The methods can be categorised into three groups; (1) methods that have consistently high run time, namely the original density peaks clustering method, and the adaptive density peaks method; (2) methods that achieve fast run time for small datasets but whose run time increases super-linearly with the number of data points, namely the comparative density peaks method and mean shift, and (3) methods that achieve consistently fast run times, namely DBSCAN, HDBSCAN, quick shift, and the methods introduced in this thesis namely the density peaks method using a k-NN density estimator, DCF and CPF. The slow approaches both use density estimators with quadratic complexity. This impedes their application to larger datasets. For small datasets, DBSCAN

Dataset	**DCF**	**CPF**	**DPC**	**ODP**	**ADP**	**CDP**	**DBS**	**HDB**	**MNS**	**QKS**
Dermatology	0.10	0.19	0.10	4.65	2.5	0.32	**0.01**	0.02	1.11	0.05
Ecoli	0.09	0.16	0.08	2.54	1.67	0.33	**0.00**	0.02	0.93	0.03
Glass	0.08	0.08	0.08	0.61	0.24	0.11	**0.00**	**0.00**	0.67	0.03
Letter Recognition	11.13	24.00	**10.44**	2430.84	1002.42	372.14	19.94	25.53	1128.16	35.41
Optdigits	1.67	4.45	1.53	126.83	2404.59	4.80	2.03	**1.64**	25.11	1.88
Page Blocks	0.73	1.44	**0.67**	123.27	43.26	14.59	1.23	0.68	21.77	12.78
Pendigits	1.54	3.33	**1.32**	320.98	126.74	15.25	2.72	1.49	30.45	5.89
Phonemes	8.34	21.13	9.18	1627.81	57.33	43.22	16.26	11.42	24.12	**4.80**
Seeds	0.07	0.08	0.09	0.51	7.78	0.03	**0.00**	0.00	1.07	0.01
Vertebral	0.08	0.05	0.10	0.84	15.28	0.11	**0.00**	**0.00**	0.40	0.02

Table 6.5: The average run time for the real-world datasets.

and HDBSCAN achieve the fastest run time, however the magnitude of difference with DCF and CPF is unlikely to hinder their use in applications. For larger datasets, DCF and CPF remain competitive with the fastest methods and achieve near the fastest run time for Letter Recognition, the dataset with the largest number of instances assessed. It is concluded that DCF and CPF, as well as achieving high quality clustering without specification of the true number of clusters, do so efficiently, with computation that gracefully scales to larger datasets.

6.3.2 Analysis of the Parameter Space

DCF and CPF achieve superb results across the datasets when optimal values for the parameters are applied. The consistency of the performance of the two approaches is now demonstrated for a wide range of parameter values. DCF has two parameters: (1) k, the number of neighbors computer for each point when computing the k-NN density estimator and (2) β, the amount of variation in the density used to determine cluster cores. CPF also 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 6.3. In Figure 6.2 we present the clustering quality in terms of the ARI and the AMI over a broad range of parameter values, for each dataset. In each subfigure, the first row shows how the ARI and the AMI varies with changes in k and β, and k and ρ respectively. The remaining rows show changes in the ARI and AMI with respect to changes in the primary parameter of each competitor method according to guidance provided by the authors.

It is clear that the performance of DCF and CPF is robust to the choice of the hyper-parameters. We see immediately that the performance of DCF is robust to choices of k and β. The results remain consistent as k is increased. For all datasets except Letter Recognition, choosing $k \approx \sqrt{n}$ returns high quality results. The quality of the clusterings remain also remarkably consistent as the variation permitted within the

cluster core increases from $\beta = 0.1$ to $\beta = 0.9$. In applications, setting $\beta = 0.4$ initially is advised, as it is observed to achieve near optimal results for all datasets bar Page Blocks.

Similarly, 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 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. Users can intuitively tune the parameter ρ for alternate clusterings, increasing ρ if more clusters are desired and decreasing ρ if fewer clusters are desired.

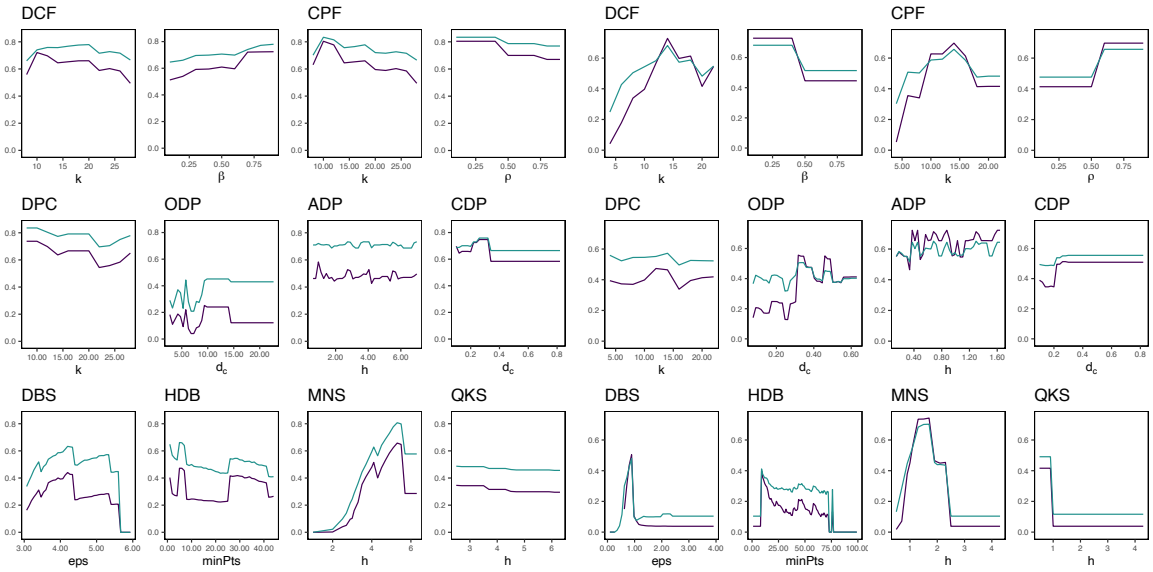
Considering the competitor methods, it is noted that ADP, CDP and quick shift also achieve consistent results as the values of their respective parameters increase. Each of these methods, as well as DCF and CPF, allocate instances to the same cluster as their nearest neighbor of higher local density.

An additional benefit of DCF and CPF over all competitor methods except HDBSCAN 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. By contrast, methods such as DBSCAN and MeanShift require an understanding of the average distances between instances in the dataset.

6.4 Conclusion

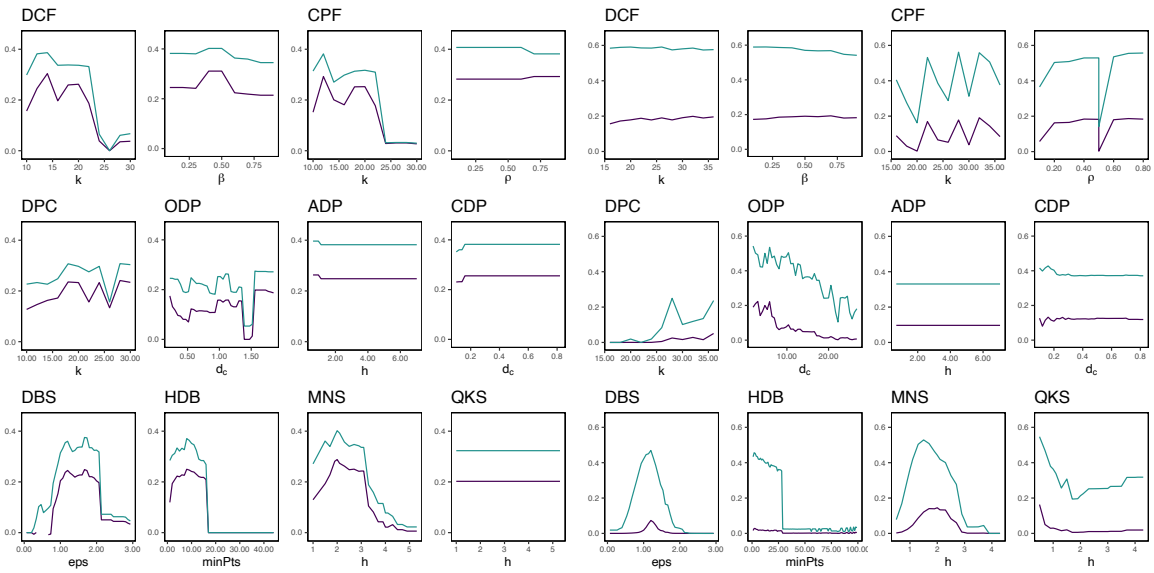
In this chapter, the performance of the methods introduced in this thesis were compared against a broad range of prominent non-parametric density-based clustering methods. Using ten real-world datasets, it was shown that DCF and CPF consistently outperform the competitor methods. They achieve superior clusterings over almost all the datasets assessed measured using two prominent metrics, the ARI and the AMI. In the few cases where DCF and CPF do not return the best clustering, the performance never

drops significantly. Furthermore, these results are achieved without requiring the true number of clusters as an input. As a result, both DCF and CPF can be recommended for use in the exploratory analysis of a broad range of datasets. The stability of the results was also proven, by showing the clustering results as the hyper-parameters input to each method is varied. DCF and CPF have interpretable parameters, allowing users to make high-quality selections for their values using only knowledge of the number of instances in the data.



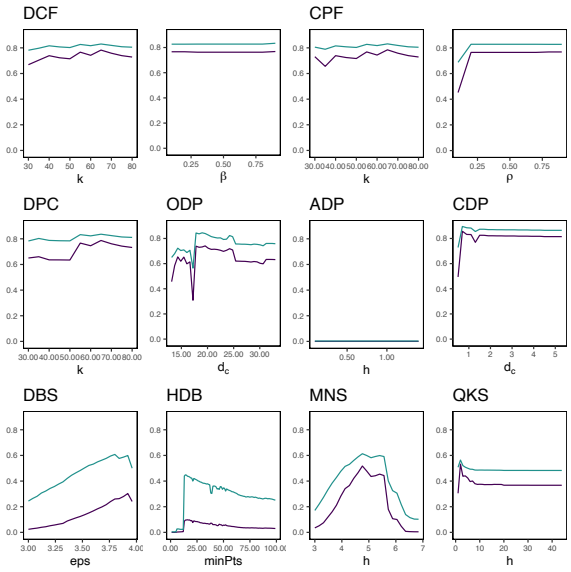
(a) Dermatology

(b) Ecoli

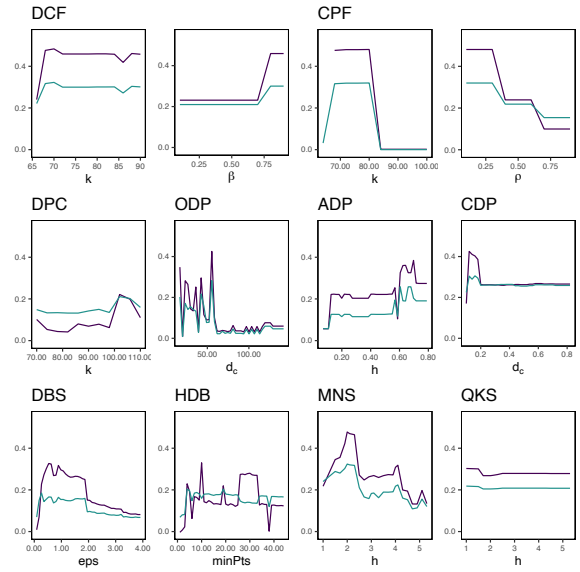


(c) Glass

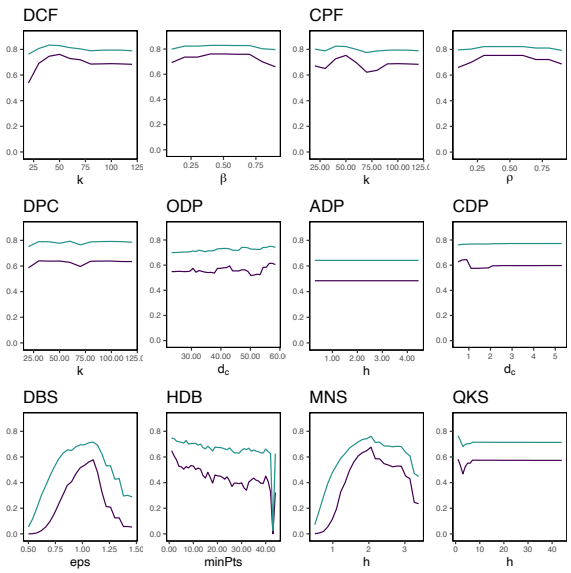
(d) Letter Recognition



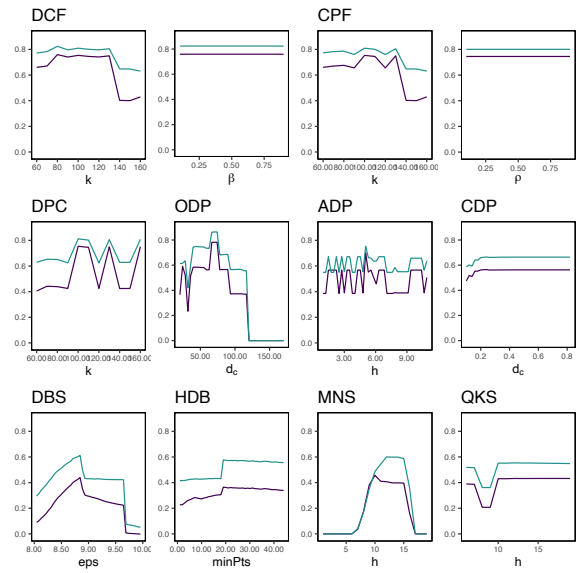
(e) Optdigits



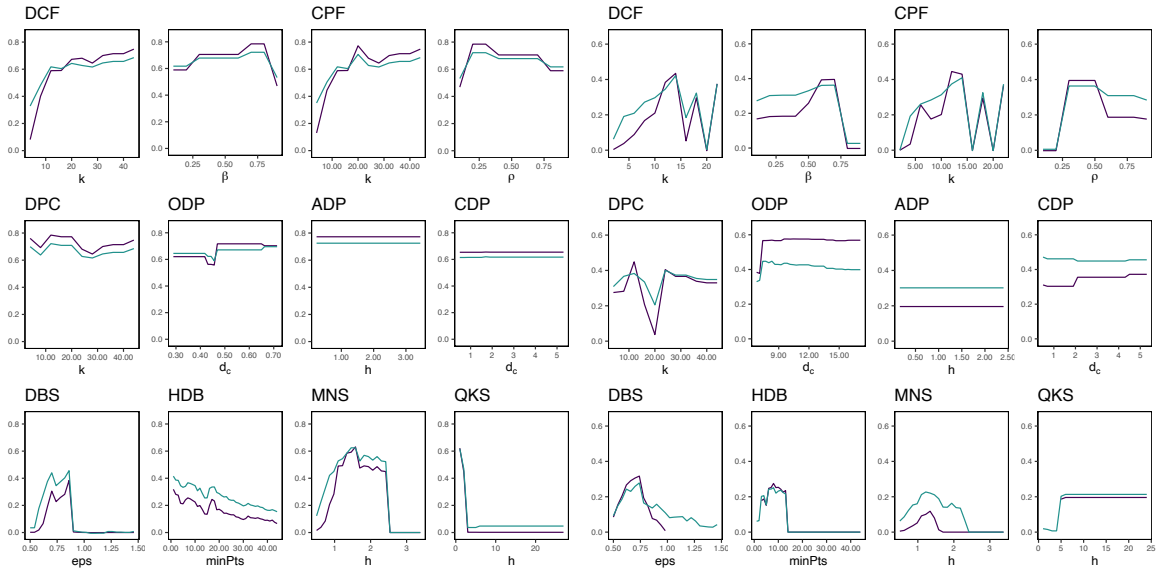
(f) Page Blocks



(g) Pendigits



(h) Phonemes



(i) Seeds (j) Vertebral

Figure 6.2: For each dataset and clustering algorithm, we show the clustering quality as a function of the input parameters. The ARI is shown in purple, and the AMI in green. Note that for DCF, we present the clustering quality as a function of both k and β and for CPF, we present the clustering quality as a function of both k and ρ. For the DCF plots assessing the k parameter, β is set to the optimal value for each dataset (in the order of appearance, $\beta = \{0.7, 0.4, 0.1, 0.4, 0.1, 0.8, 0.4, 0.4, 0.4, 0.1\}$). For the DCF plots assessing the β parameter, k is set to the optimal value for each dataset (in the order of appearance, $k = \{10, 14, 12, 18, 55, 74, 50, 80, 12, 8\}$). For the CPF plots assessing the k parameter, ρ is set to the optimal value for each dataset (in the order of appearance, $\rho = \{0.3, 0.6, 0.9, 0.6, 0.9, 0.2, 0.6, 0.6, 0.6, 0.9\}$). For the DCF plots assessing the ρ parameter, k is set to the optimal value for each dataset (in the order of appearance, $k = \{10, 14, 12, 28, 55, 80, 50, 110, 12, 8\}$).

7 Density Peaks for Parametric Clustering

7.1 Summary

Methods that employ the EM algorithm for parameter estimation typically face the notorious yet unsolved problem that the initialization input significantly impacts the algorithm output. We here develop a Reinforced Expectation Maximization (REM) algorithm for cluster analysis using Gaussian mixture models. The competence of REM is achieved by introducing two innovative strategies into the EM framework: (1) a mode-finding strategy for initialization that detects non-trivial modes in the data using the peak-finding technique, and (2) a mode-pruning strategy for detecting true modes/mixture components of the population. The pruning strategy is well-justified in the context of mixture modelling, and we present theoretical guarantees on the quality of the initialization. Extensive experimental studies on both synthetic and real datasets show that our approach achieves better performance compared to state-of-the-art methods.

7.2 Introduction

Model-based clustering methods utilize mixture models to partition a collection of objects (Fraley and Raftery, 2002; Baudry et al., 2010a). As defined by McNicholas (2016b), a cluster in the parametric formulation should be a "unimodal component

within an appropriate finite mixture model". The clustering is done by assigning each object to the mixture component (i.e., cluster) to which it is most likely to belong a posteriori. The most widely used mixture model is the Gaussian mixture model (GMM), and the Expectation-Maximization (EM) algorithm is the most popular algorithm for parameter estimation. As discussed in Chapter 2, the EM algorithm for GMMs has several drawbacks (Bishop, 2006, Chapter 9): it may converge to a singularity at which the likelihood is infinite, leading to meaningless estimates; it is sensitive to initialization, because the log-likelihood function is not unimodal, and the resulting solution is a local optimum in the neighborhood of the initial guess. Melnykov and Maitra (Melnykov and Maitra, 2010) partitioned existing initialization methods for the EM algorithm into the stochastic category and the deterministic category. The simplest stochastic strategy is random initialization. As discussed previously, Jin et al. (2016) proved that, with high probability, the EM algorithm with random initialization will converge to bad local maxima, whose log-likelihood could be arbitrarily worse than that of the global maximum. A prominent deterministic method, implemented in the R package *mclust* (Scrucca et al., 2016), initializes the EM algorithm with the solution of model-based Gaussian hierarchical clustering.

The challenges initializing the EM algorithm are compounded by the fact that, generally, the true number clusters is unknown. A common practice is to run the EM algorithm with an initialization method for different cluster numbers, and decide the optimal cluster number via a model selection criterion such as those introduced in Chapter 2 namely the AIC, the BIC, and the ICL. Stochastic initialization methods struggle to provide consistent initializations for multiple runs with different cluster numbers, making model selection more difficult. The deterministic initialization method in *mclust* also provides inconsistent results when the true number of clusters is not specified.

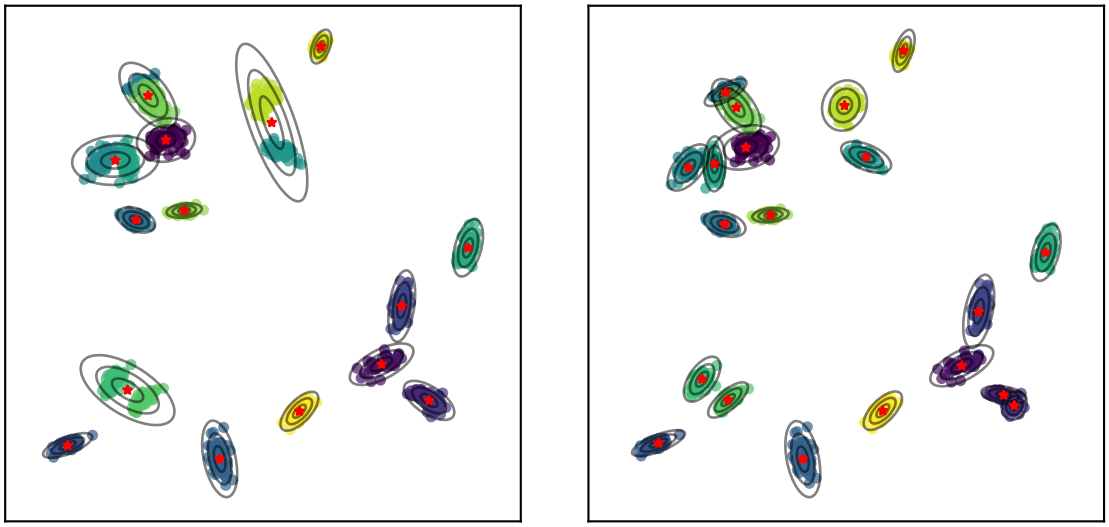
This leads to the key driving force of the present work: to tackle the initialization problem of the EM algorithm for GMMs, utilizing the notion of exemplars/medoids. Following the notion of a model-based cluster given by McNicholas, the modes of a

Gaussian mixture density are seen to be symptomatic of the underlying population structure and can guide the initialization procedure. In particular, if the Gaussian components in a GMM are well separated, then the modes exactly match the Gaussian means; if the components overlap a bit, then the modes would include but not be limited to the Gaussian means (Améndola et al., 2020). This motivates us to use a KDE with the Gaussian kernel to approximately locate the modes of the mixture density, within which to pinpoint the Gaussian means. As demonstrated in this thesis, exemplars from the data can provide a fast and intuitive way to recover high-quality estimates of the density modes, henceforth referred to as exemplars. Therefore, we apply the peak-finding technique to detect the exemplars in the data. By restricting the initial values for the Gaussian means to the set of exemplars, the initialization is robust to outliers and adjusts to the location of the centers.

From an inclusive pool of initial exemplars, we produce a hierarchy of clusterings by iteratively pruning superfluous clusters through the optimization of a convex objective function regularized by an adaptive cardinality penalty. We prune exemplars one at a time, automatically generating a nested sequence of clustering results from which the preferred clustering can be selected.

The initialization-pruning framework is called the reinforced EM (REM) algorithm. By selecting the initial Gaussian means from the exemplar set only, the REM algorithm never allows components collapse into one point at which the likelihood is infinite. Furthermore, the objective function for exemplar pruning is well-justified in the context of mixture modelling; it is solved analytically, leading to efficient run time for large datasets. A comparison of the REM method to *mclust* on a synthetic dataset is presented in Figure 7.1.

The remainder of the chapter is organized as follows: in Section 7.3, we provide a summary of the EM algorithm for GMMs and overview of existing approaches; in Section 7.4, we introduce the peak-finding method used by REM for mean initialization; in Section 7.5, the REM algorithm is described; Section 7.6 presents an experimental



(a) *mclust* (b) *REM*

Figure 7.1: Comparison between mclust and REM, on a synthetic dataset containing 20 clusters. Optimal clusterings were selected using BIC Schwarz (1978).

evaluation, and Section 7.7 concludes.

7.3 Background

A GMM density has the form $f(\boldsymbol{x}) = \sum_{j=1}^{m} \pi_j \phi(\boldsymbol{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$, with mixing proportions π_j ($\pi_j > 0$ and $\sum_{j=1}^{m} \pi_j = 1$), and each Gaussian density $\phi(\cdot; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ has a mean $\boldsymbol{\mu}_j$ and a covariance matrix $\boldsymbol{\Sigma}_j \succ 0$. Let $\boldsymbol{\pi}$ denote the vector of mixing proportions: $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_m)^T$. The log-likelihood function is

$$\ell(\boldsymbol{\pi}, \{\boldsymbol{\mu}_j\}_{j=1}^{m}, \{\boldsymbol{\Sigma}_j\}_{j=1}^{m}; \boldsymbol{X}) = \sum_{i=1}^{n} \log(\sum_{j=1}^{m} \pi_j \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)).$$

The classical method for computing maximum-likelihood estimates for GMM parameters is the EM algorithm. The EM algorithm consists of the following steps:

1. Initialize the parameters: $\{\pi_1, \ldots, \pi_m\}$, $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m\}$ and $\{\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_m\}$.

2. Compute the responsibilities:

$$r_{ij} = \frac{\pi_j \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{v=1}^{m} \pi_v \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v)},$$

for $1 \leq i \leq n$ and $1 \leq j \leq m$.

3. Update the estimates:

$$\pi_j = \frac{\sum_{i=1}^{n} r_{ij}}{n}, \ \boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n} r_{ij}\boldsymbol{x}_i}{\sum_{i=1}^{n} r_{ij}}, \text{ and } \boldsymbol{\Sigma}_j = \frac{\sum_{i=1}^{n} r_{ij}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T}{\sum_{i=1}^{n} r_{ij}},$$

for $1 \leq j \leq m$.

4. Iterate steps 2 and 3 until convergence.

The hill-climbing nature of the EM algorithm, coupled with the multi-modal surface of the log-likelihood function, lends crucial importance to the quality of the initialization. The simplest initialization strategy draws initial values at random from the parameter space or from the data pool. Jin et al. (2016) proved that, with high probability, the EM algorithm with random initialization will converge to bad local maxima, whose log-likelihood could be arbitrarily worse than that of the global maximum. Another intuitive idea is to run EM with multiple random starts for each value of m. The *emEM* algorithm (Biernacki et al., 2003) consists of several short runs of EM, initialized with random starts, until a loose convergence criterion is satisfied. The solution with the highest log-likelihood is used to initialize a long run of EM with strict converge criteria. A related approach, called *Rnd-EM* (Maitra, 2009), computes the log-likelihood of several random starts without running any EM iterations. The best is used as the initializer for the long EM stage. The k-means algorithm is also frequently used to provide an initial partition of the data, from which initial parameter estimates can be computed. However, the k-means algorithm itself requires a good initialization, typically achieved with the k-means++ method (Vassilvitskii and Arthur, 2006). None of these methods make efforts to ensure the similarity of initializations for different cluster numbers. This leads to unstable clusterings and hinders comparison of clusterings using model selection criteria.

The popular *mclust* package in R provides hard partitions of the data using an agglommerative model-based clustering technique. Initializations for different numbers

of clusters are found from partitions extracted using dendrogram derived from the hierarchical clustering. This approach provides a hard partition of the data, leading to improper splitting of true components when the estimated cluster number is greater than the true number of components.

7.4 Exemplar Selection

We initialize the mean vectors for the EM algorithm using the exemplars (i.e., medoids) in the data. The peak-finding method in Section 3.2 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 apply the Gaussian kernel with bandwidth $h > 0$ for density estimation:

$$\hat{f}_h(\boldsymbol{x}) = \frac{1}{n \cdot h^p} \sum_{i=1}^{n} K\left(\frac{\boldsymbol{x} - \boldsymbol{x}_i}{h}\right),$$

where $K(\cdot)$ is the Gaussian kernel. Note that our density estimate is different from the k-NN density estimator primarily used in this thesis. We here use a KDE with Gaussian kernel as (1) it is itself a form of mixture based estimator, centering a spherical Gaussian component at each observation, and (2) as demonstrated in Section 3.3, the same theoretical guarantees regarding the consistency of the estimator are available. For the distance input, as before we define

$$b(\boldsymbol{x}) = \underset{\boldsymbol{x}' \in \boldsymbol{X}}{\arg\min} \left\{ \|\boldsymbol{x} - \boldsymbol{x}'\| : \hat{f}_h(\boldsymbol{x}) < \hat{f}_h(\boldsymbol{x}') \right\},$$

i.e. the nearest neighbor of $\boldsymbol{x}$ with a higher density. Then the distance from $\boldsymbol{x}$ to its nearest neighbor of higher density is simply $\omega(\boldsymbol{x}) = \|\boldsymbol{x} - b(\boldsymbol{x})\|$. For the point with the highest density estimate $\boldsymbol{x} = \arg\max_{\boldsymbol{x}' \in \boldsymbol{X}} \hat{f}_h(\boldsymbol{x}')$, the distance is defined as $\omega(\boldsymbol{x}) = \max_{\boldsymbol{x}' \in \boldsymbol{X}} \|\boldsymbol{x} - \boldsymbol{x}'\|$.

Intuitively, $\omega(\boldsymbol{x})$ will be large if $\boldsymbol{x}$ has a locally or globally maximal density, or if $\boldsymbol{x}$ is an outlier. Therefore, a data point $\boldsymbol{x}$ will be selected as an exemplar by the peak-finding method, only if both $\hat{f}_h(\boldsymbol{x})$ and $\omega(\boldsymbol{x})$ are large. To generate an initial set of exemplars

$\widehat{\mathcal{M}}_0 = \{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_\kappa^*\}$, threshold values for the density $\hat{f}_h(\boldsymbol{x})$ and the distance $\omega(\boldsymbol{x})$ need to be set: the exemplars are the data points with the two metric values both above the thresholds, i.e. $\widehat{\mathcal{M}}_0 = \{\boldsymbol{x} \in \boldsymbol{X} : \hat{f}_h(\boldsymbol{x}) \geq l, \omega(\boldsymbol{x}) \geq \tau\}$. As in the method of Section 3.2, we provide users with a decision plot, a scatter plot of $\{(\hat{f}_h(\boldsymbol{x}), \omega(\boldsymbol{x})) : \boldsymbol{x} \in \boldsymbol{X}\}$, to provide intuition regarding the threshold values. An example of the decision graph and the selected exemplars is provided in Figure 7.2.

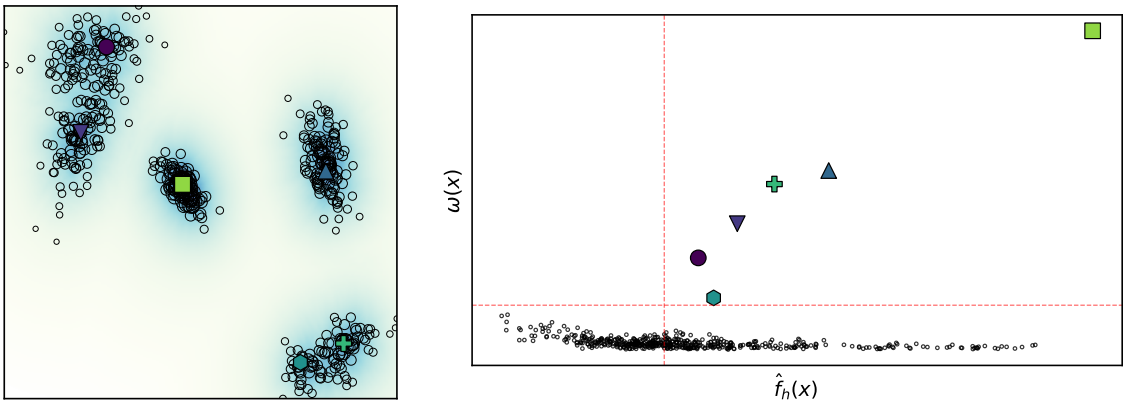


Figure 7.2: An example with five components. The left plot shows the estimated density of the data, with darker regions having higher density. Also shown are the locations of the exemplars, with colour and marker type corresponding to the instances in the right plot. The right plot is the decision plot, where the red lines indicate the threshold values. The peak-finding method selects six initial exemplars, with each true component well represented.

Following the theoretical guarantees provided in Chapter 3, it is claimed that the set of exemplars obtained by the peak-finding method contains unique and consistent estimates of all modes in the data. Specification of the threshold values l and τ allows for modes recovered at different densities and resolutions. Figure 7.2 gives an example, showing that the peak-finding method can select multiple exemplars close to the true center of each cluster.

For high-dimensional data, we recommend to project the data into a space of lower dimension, using the scaled SVD transformation, to compute more robust density estimates. Once exemplars are selected, the REM algorithm can be run on the original data.

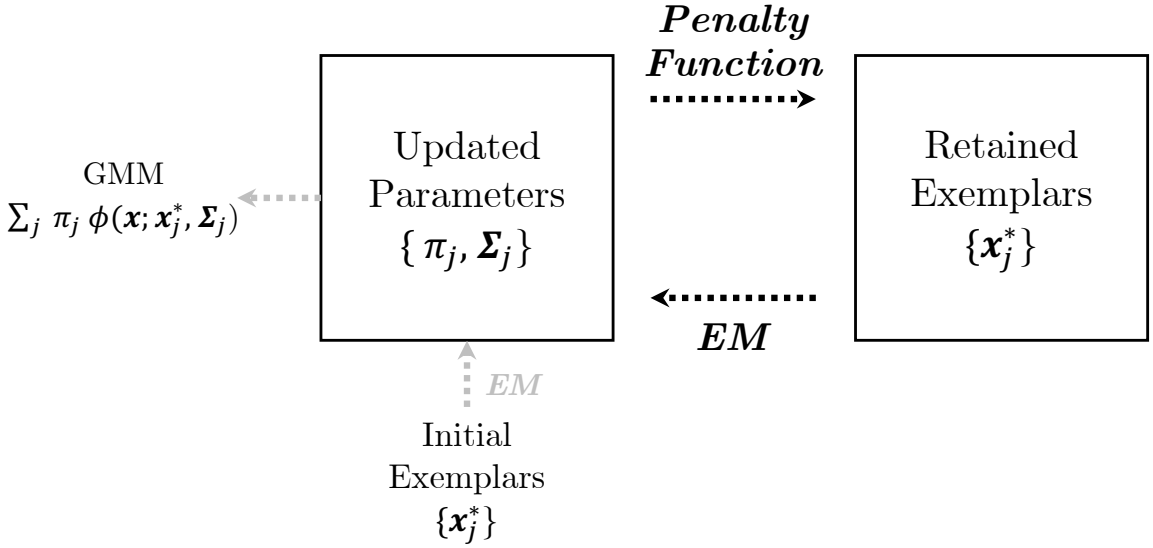


Figure 7.3: The iterative pruning procedure. Given any exemplar set $\{\boldsymbol{x}_j^*\}$, we estimate the parameters $\{\pi_j, \boldsymbol{\Sigma}_j\}$ by an EM-type algorithm. At convergence, we obtain a GMM $\sum_j \pi_j \phi(\boldsymbol{x}; \boldsymbol{x}_j^*, \Sigma_j)$. Then we optimize a regularized objective function to force certain π_j's to be 0. The relevant exemplars are removed from the exemplar set.

7.5 The Iterative Pruning Procedure

The REM algorithm has two blocks: the EM block and the pruning block; see Figure 7.3. Given the initial exemplar set $\widehat{\mathcal{M}}_0 = \{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_\kappa^*\}$, our iterative procedure will produce a sequence of nested clustering results, respectively with $\kappa - 1, \kappa - 2, \ldots, 2$ mixture components. The final optimal clustering is determined by a model selection criterion of the user's choice. In the following, if a data point is in the exemplar set, it is excluded from the data pool, and once a data point is pruned from the exemplar set, it will go back to the data pool. In other words, if $\boldsymbol{x}_j^*$ is pruned in the pruning block, then we update $\widehat{\mathcal{M}} = \widehat{\mathcal{M}}/\{\boldsymbol{x}_j^*\}$, $\boldsymbol{X} = \boldsymbol{X} \cup \{\boldsymbol{x}_j^*\}$ and $n = n + 1$, before running the EM block.

7.5.1 EM Block

Given the updated data pool (the original data without the retained exemplars), the EM block operates as follows.

1. Input: The retained exemplars $\{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_m^*\}$ and the responsibilities $\{r_{ij} : i =$

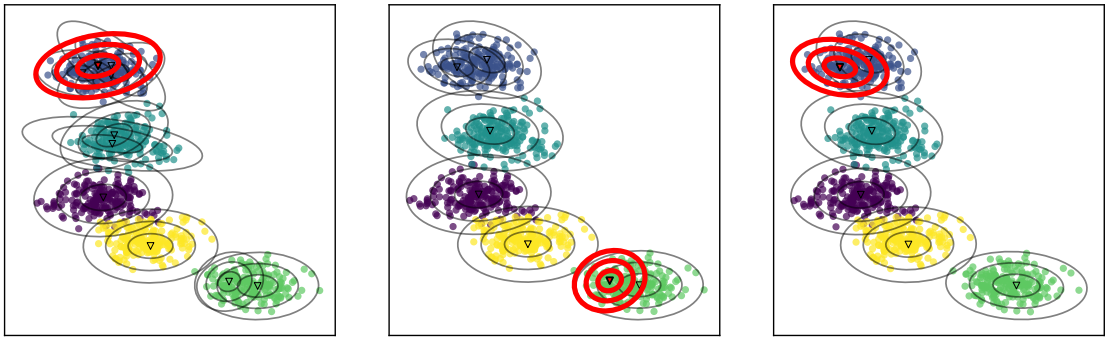


Figure 7.4: A worked example with five clusters. The colors represent the clusters; the contours represent the covariance matrices, with red contours indicating the component to be pruned in the next iteration.

$1, \ldots, n, j = 1, \ldots, m\}$.

2. Update the estimates: for $j = 1, \ldots, m$,

$$\pi_j = \frac{\sum_{i=1}^{n} r_{ij}}{n}, \boldsymbol{\Sigma}_j = \frac{\sum_{i=1}^{n} r_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j^*)(\boldsymbol{x}_i - \boldsymbol{x}_j^*)^T}{\sum_{i=1}^{n} r_{ij}}.$$

3. Compute the responsibilities: for $i = 1, \ldots, n$ and $j = 1, \ldots, m$,

$$r_{ij} = \frac{\pi_j \phi(\boldsymbol{x}_i; \boldsymbol{x}_j^*, \boldsymbol{\Sigma}_j)}{\sum_{v=1}^{m} \pi_v \phi(\boldsymbol{x}_i; \boldsymbol{x}_v^*, \boldsymbol{\Sigma}_v)}.$$

4. Iterate steps 2 and 3 until convergence.

It is immediately clear that, in the EM block, the Gaussian means are fixed at the given exemplars, and only the mixing proportions and covariance matrices are estimated. Since the exemplar set is disjoint from the dataset, the mean vectors will always differ from the data point, and therefore the iteration will never converge to a degenerate solution with a zero covariance matrix. The GMM density at convergence is

$$f(\boldsymbol{x}) = \sum_{j=1}^{m} \pi_j \phi(\boldsymbol{x}; \boldsymbol{x}_j^*, \boldsymbol{\Sigma}_j).$$

7.5.2 Pruning Block

While the original exemplar set contains consistent estimates of all the density peaks of the mixture model, the number of density peaks can be significantly larger than the number of mixture components (Améndola et al., 2020). Hence, the exemplars $\{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_\kappa^*\}$ need to be further filtered to obtain the true mean vectors. In the pruning block, we prune exemplars by inducing sparsity in the mixing proportion vector $\boldsymbol{\pi}$ such that, if $\pi_j = 0$, then the exemplar $\boldsymbol{x}_j^*$ will be removed from the exemplar set and returned to the data pool. Let $\mathbf{1}_n$ denote the vector of 1's of dimension n; let Δ denote the probabilistic simplex of the appropriate dimension: if $\boldsymbol{\pi} \in \Delta$, then $\boldsymbol{\pi} \geq 0$ and $\|\boldsymbol{\pi}\|_1 = 1$. Given the exemplar set $\widehat{\mathcal{M}}_0$ and covariance-matrix estimates $\{\boldsymbol{\Sigma}_j\}_{j=1}^\kappa$, let $\boldsymbol{D} = [d_{ij}]_{n\times\kappa}$ denote the distance matrix, where $d_{ij}^2 = (\boldsymbol{x}_i - \boldsymbol{x}_j^*)^T \boldsymbol{\Sigma}_j^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_j^*)$.

The Objective Function

Given the exemplars, the log-likelihood function is $\sum_{i=1}^n \log\left(\sum_{j=1}^\kappa \pi_j \phi(\boldsymbol{x}_i; \boldsymbol{x}_j^*, \boldsymbol{\Sigma}_j)\right)$. We have by Jensen's inequality that

$$
\begin{aligned}
-\log\left(\sum_{j=1}^{\kappa} \pi_j \phi(\boldsymbol{x}_i; \boldsymbol{x}_j^*, \boldsymbol{\Sigma}_j)\right) &= \left(\sum_{j=1}^{\kappa} r_{ij}\right) \log\left(\frac{\sum_{j=1}^\kappa r_{ij}}{\sum_{j=1}^\kappa \pi_j \phi(\boldsymbol{x}_i; \boldsymbol{x}_j^*, \boldsymbol{\Sigma}_j)}\right) \\
&\leq \sum_{j=1}^{\kappa} r_{ij} \log\left(\frac{r_{ij}}{\pi_j \phi(\boldsymbol{x}_i; \boldsymbol{x}_j^*, \boldsymbol{\Sigma}_j)}\right),
\end{aligned}
$$

where the upper bound is achievable when $r_{ij} \in \{0, 1\}$. Then the negative log-likelihood can be formulated as an optimization problem:

$$
-\sum_{i=1}^{n} \log\left(\sum_{j=1}^{\kappa} \pi_j \phi(\boldsymbol{x}_i; \boldsymbol{x}_j^*, \boldsymbol{\Sigma}_j)\right) = \min_{\{\boldsymbol{R}_{i\cdot} \in \Delta\}_{i=1}^n} \sum_{i=1}^{n} \sum_{j=1}^{\kappa} r_{ij} \log\left(\frac{r_{ij}}{\pi_j \phi(\boldsymbol{x}_i; \boldsymbol{x}_j^*, \boldsymbol{\Sigma}_j)}\right),
$$

where $\boldsymbol{R} = [r_{ij}]_{n\times\kappa}$ and $i\cdot$ indicates the ith row. Then maximizing the log-likelihood is equivalent to

$$
\min_{\{\boldsymbol{x}_j^* \in \mathbb{R}^p, \boldsymbol{\Sigma}_j \succ 0\}_{j=1}^\kappa} \min_{\{\boldsymbol{R}_{i\cdot} \in \Delta\}_{i=1}^n} \sum_{i=1}^{n} \sum_{j=1}^{\kappa} r_{ij} \log\left(\frac{r_{ij}}{\pi_j \phi(\boldsymbol{x}_i; \boldsymbol{x}_j^*, \boldsymbol{\Sigma}_j)}\right).
$$

In the pruning block, the $\boldsymbol{x}_j^*$'s and $\boldsymbol{\Sigma}_j$'s are fixed, and hence the optimization variables are the responsibilities only:

$$\min_{\{\boldsymbol{R}_{i\cdot}\in\Delta\}_{i=1}^n} \quad \sum_{i=1}^{n}\sum_{j=1}^{\kappa} r_{ij}\log\left(\frac{r_{ij}}{\pi_j\phi(\boldsymbol{x}_i;\boldsymbol{x}_j^*,\boldsymbol{\Sigma}_j)}\right).$$

The detailed formulation of the objective function is

$$\sum_{i=1}^{n}\sum_{j=1}^{\kappa} r_{ij}\log\left(\frac{r_{ij}}{\pi_j\phi(\boldsymbol{x}_i;\boldsymbol{x}_j^*,\boldsymbol{\Sigma}_j)}\right) = \sum_{i=1}^{n}\sum_{j=1}^{\kappa} r_{ij}\times\left[\log\left(\frac{r_{ij}}{\pi_j}\right)+\frac{1}{2}\log\left(|\boldsymbol{\Sigma}_j|\right)+\frac{1}{2}(\boldsymbol{x}_i-\boldsymbol{x}_j^*)^T\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i-\boldsymbol{x}_j^*)\right].$$

To shrink the proportion vector $\boldsymbol{\pi}$, we take out the first term in the brackets; otherwise, numerical algorithms will behave erratically when $\pi_j \to 0$. Further discussion of the motivation for and impact of removing this term is given in Section 7.5.2

This yields the optimization problem

$$\min_{\{\boldsymbol{R}_{i\cdot}\in\Delta\}_{i=1}^n} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{\kappa} r_{ij}(\boldsymbol{x}_i-\boldsymbol{x}_j^*)^T\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i-\boldsymbol{x}_j^*)+\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{\kappa} r_{ij}\log\left(|\boldsymbol{\Sigma}_j|\right).$$

In matrix-vector notation, the optimization problem is

$$\min_{\{\boldsymbol{R}_{i\cdot}\in\Delta\}_{i=1}^n} \quad \frac{1}{2}\sum_{i=1}^{n}\boldsymbol{R}_{i\cdot}^T(\boldsymbol{D}_{i\cdot}+\boldsymbol{\xi}), \tag{7.1}$$

where $\boldsymbol{\xi} = (\log(|\boldsymbol{\Sigma}_1|),\ldots,\log(|\boldsymbol{\Sigma}_\kappa|))^T$.

The Penalty

Problem (7.1) is not amenable to the classical ℓ_1-norm penalty on $\boldsymbol{\pi}$, since $\|\cdot\|_1 = 1$ is constant on a simplex. This motivates us to develop a penalty in the form of $\|\boldsymbol{\delta}\circ\boldsymbol{\pi}\|_1 = \boldsymbol{\delta}^T\boldsymbol{\pi}$, where $\circ$ is the element-wise multiplication operator. The weight vector $\boldsymbol{\delta} = (\delta_1,\ldots,\delta_\kappa)^T$ should be data-driven and has the desirable property that gives more penalty to closer exemplars. Therefore, we define δ_i as the probability

that an instance from the ith mixture component is misclassified into the jth mixture component:

$$\delta_i = \max_{j=1,\ldots,\kappa} \Pr\left(\pi_i \phi(\boldsymbol{x}; \boldsymbol{x}_i^*, \boldsymbol{\Sigma}_i) < \pi_j \phi(\boldsymbol{x}; \boldsymbol{x}_j^*, \boldsymbol{\Sigma}_j) | \boldsymbol{x} \sim N(\boldsymbol{x}_i^*, \boldsymbol{\Sigma}_i)\right).$$

This definition was introduced by Maitra and Melnykov (2010) to measure the degree of overlap between two Gaussian distributions. The weight δ_i reflects the likelihood of the exemplar $\boldsymbol{x}_i^*$ belonging to the group of another exemplar. Exemplars favoured by this penalty are in keeping with the peak-finding conception of cluster centers as instances with high density, and relatively large distance to other points of higher density.

For Gaussian distributions with homogeneous covariance matrices, the computation of the probability is straightforward and related to the Mahalanobis distance between the exemplars. For general covariance matrices, the computation involves evaluating the cumulative distribution function of linear combinations of independent non-central chi-squared and normal random variables. A method using the algorithm AS 155 of Davies (1980) has been implemented in C as part of the *MixSim* package in R (Melnykov et al., 2012). Example values of the δ_i's for four typical scenarios of two-component mixtures are provided in Figure 7.5.

Objective Minimization

Our penalized objective function is

$$\min_{\{\boldsymbol{R}_{i\cdot} \in \Delta\}_{i=1}^n} \quad \frac{1}{2} \sum_{i=1}^n \boldsymbol{R}_{i\cdot}^T (\boldsymbol{D}_{i\cdot} + \boldsymbol{\xi}) + \theta \boldsymbol{\delta}^T \boldsymbol{R}^T \mathbf{1}_n, \tag{7.2}$$

where the regularization term will force certain columns of $\boldsymbol{R}$ to be exactly zero.

The objective function (7.2) is linear and hence can be simplified to be

$$\min_{\{\boldsymbol{R}_{i\cdot} \in \Delta\}_{i=1}^n} \sum_{i=1}^n \boldsymbol{R}_{i\cdot}^T \boldsymbol{b}_i = \sum_{i=1}^n \min_{\boldsymbol{R}_{i\cdot} \in \Delta} \boldsymbol{R}_{i\cdot}^T \boldsymbol{b}_i,$$

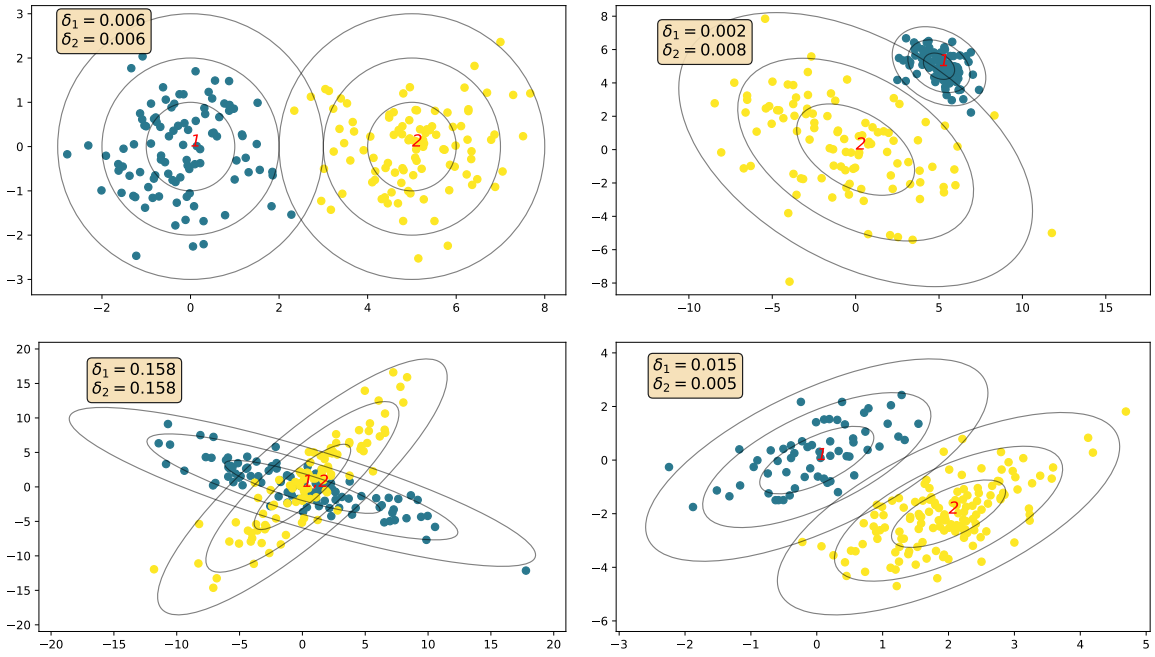


Figure 7.5: The overlap penalty for typical scenarios with two components: (1) equal weights and homogeneous spherical covariance matrices, (2) equal weights and heterogeneous covariance matrices, (3) equal weights, common mean and heterogeneous covariance matrices, (4) unequal weights and homogeneous covariance matrices.

where $\boldsymbol{b}_i = \frac{1}{2}\boldsymbol{D}_{i\cdot} + \frac{1}{2}\boldsymbol{\xi} + \theta\boldsymbol{\delta}$.

The parameter θ controls the amount of shrinkage on $\boldsymbol{\pi}(= \frac{1}{n}\boldsymbol{R}^T\mathbf{1}_n)$. In fact, the solution path is piecewise linear in θ, and the whole trajectory of $\boldsymbol{\pi}$, as a function of θ, can be easily computed by piecewise-linear homotopy methods. We now provide details on how the trajectory of $\boldsymbol{\pi}$, as a function of θ, is computed. The objective function (7.2) is equivalent to

$$\min_{\{\boldsymbol{R}_{i\cdot}\in\Delta\}_{i=1}^n} \sum_{i=1}^n \boldsymbol{R}_{i\cdot}^T[\boldsymbol{D}_{i\cdot} + \boldsymbol{\xi} + \theta\boldsymbol{\delta}] = \sum_{i=1}^n \min_{\boldsymbol{R}_{i\cdot}\in\Delta} \boldsymbol{R}_{i\cdot}^T\boldsymbol{b}_i,$$

where $\boldsymbol{b}_i = \boldsymbol{D}_{i\cdot} + \boldsymbol{\xi} + \theta\boldsymbol{\delta}$, and we have incorporated the factor $\frac{1}{2}$ into the parameter θ. The solution to each individual problem is:

$$r_{ij}^* = \begin{cases} 1, & \text{if } j = \arg\min_{1\le v\le\kappa}\{b_v\}; \\ 0, & \text{otherwise.} \end{cases}$$

If the first exemplar $\boldsymbol{x}_1^*$ were to be pruned, we would require a θ value such that:

$$\begin{aligned} b_{11} &\geq \min\{b_{12}, b_{13}, \ldots, b_{1\kappa}\} \\ b_{21} &\geq \min\{b_{22}, b_{23}, \ldots, b_{2\kappa}\} \\ &\vdots \\ b_{n1} &\geq \min\{b_{n2}, b_{n3}, \ldots, b_{n\kappa}\} \end{aligned}$$

Therefore, the θ value should satisfy that

$$\begin{aligned} d_{11} + \xi_1 + \theta\delta_1 &\geq \min\{d_{12} + \xi_2 + \theta\delta_2, d_{13} + \xi_3 + \theta\delta_3, \ldots\} \\ d_{21} + \xi_1 + \theta\delta_1 &\geq \min\{d_{22} + \xi_2 + \theta\delta_2, d_{23} + \xi_3 + \theta\delta_3, \ldots\} \\ &\vdots \\ d_{n1} + \xi_1 + \theta\delta_1 &\geq \min\{d_{n2} + \xi_2 + \theta\delta_2, d_{n3} + \xi_3 + \theta\delta_3, \ldots\} \end{aligned}$$

To simplify notation, define $p_{ij}^1 = (d_{ij} + \xi_j) - (d_{i1} + \xi_1)$, for $j = 2, \ldots, \kappa$. We assume w.l.o.g. that $\delta_1 < \delta_2$ and $\delta_1 > \delta_j$ for $j \geq 3$. Then we can find the possible range of solutions for θ by considering the following set of intervals:

$$\theta \in \bigcap_{i=1}^{n} \left[\left(-\infty, \frac{p_{i2}}{\delta_1 - \delta_2} \right] \bigcup \left(\bigcup_{j=3}^{\kappa} \left[\frac{p_{ij}}{\delta_1 - \delta_j}, \infty \right) \right) \right] \tag{7.3}$$

If the interval solutions overlap, then the θ value that prunes the first exemplar is the lower bound of the overlapping interval. If the interval solutions do not overlap, resulting in $\theta \in \varnothing$, then the first exemplar will not be pruned.

By analogy, we can obtain the critical θ value for each exemplar, denoted by $\{\theta_1, \ldots, \theta_m\}$. Then with θ in (7.2) gradually increasing from 0, all the proportions in $\boldsymbol{\pi}$ also change. When θ reaches to $\min\{\theta_1, \ldots, \theta_m\}$, the exemplar with the minimal critical θ value will be pruned first, and its proportion value reduces to 0.

Given the solution $\boldsymbol{R}^*$ to problem (7.2) with one zero-column, the refined exemplar set is $\widehat{\mathcal{M}}_1 = \{\boldsymbol{x}_j^* \in \widehat{\mathcal{M}}_0 : \pi_j^* \neq 0\}$. Figure 7.6 gives an example, where we prune four

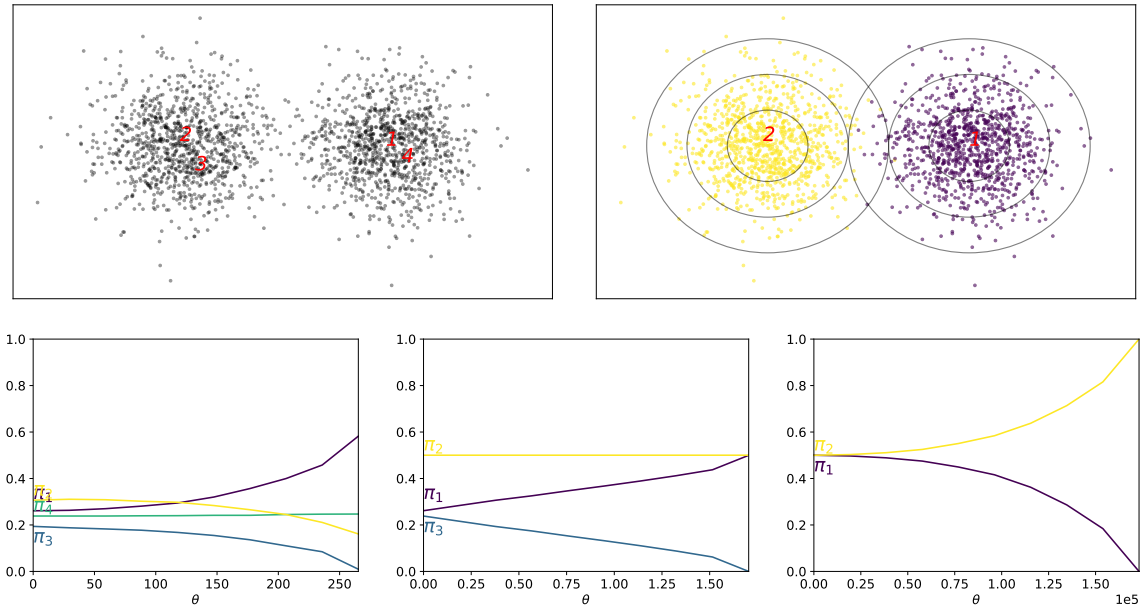


Figure 7.6: A toy example shows the piecewise-linear trajectory of $\boldsymbol{\pi}$. Top Left: The data and the four selected exemplars, labelled in decreasing order of $\hat{f}_h(\boldsymbol{x}) \times \omega(\boldsymbol{x})$. Top Right: The final clustering obtained by REM. Bottom: The whole trajectory of $\boldsymbol{\pi}$, as a function of θ, in each REM iteration. After the first iteration, exemplar $\boldsymbol{x}_3^*$ is pruned; after the second iteration, exemplar $\boldsymbol{x}_4^*$ is pruned. The bottom right panel shows that θ needs to be very large to merge two true cluster centers.

exemplars into two and plot the trajectory of $\boldsymbol{\pi}$ in each iteration.

Before continuing, we provide a motivation for the removal of the first term from the objective discussed in Section 7.5.2. Consider again the original objective function before removing the first term

$$\min_{\{\boldsymbol{R}_{i\cdot} \in \Delta\}_{i=1}^n} \quad \sum_{i=1}^{n} \sum_{j=1}^{\kappa} r_{ij} \left[\log \left(\frac{r_{ij}}{\pi_j} \right) + b_{ij} \right], \tag{7.4}$$

where $b_{ij} = \frac{1}{2} d_{ij} + \frac{1}{2} \xi_j + \frac{1}{\kappa} \theta \delta_j$ is thus a constant determined by the data and the penalty. Further expansion yields

$$\min_{\{\boldsymbol{R}_{i\cdot} \in \Delta\}_{i=1}^n} \quad \left[\sum_{i=1}^{n} \sum_{j=1}^{\kappa} r_{ij} \log \left(\frac{r_{ij}}{\pi_j} \right) + \sum_{i=1}^{n} \sum_{j=1}^{\kappa} r_{ij} b_{ij} \right].$$

Now, we have that the first term

$$\sum_{i=1}^{n}\sum_{j=1}^{\kappa} r_{ij}\log\left(\frac{r_{ij}}{\pi_j}\right),$$

is a bounded function over the product of the n simplex spaces $\Delta \times \Delta \times \ldots \times \Delta$. Moreover,

$$\lim_{r_{ij}\to 0} r_{ij}\log\left(\frac{r_{ij}}{\pi_j}\right) = 0$$
$$\lim_{\pi_j\to 0}\sum_{i=1}^{n} r_{ij}\log\left(\frac{r_{ij}}{\pi_j}\right) = 0.$$

Therefore, the optimal solution, for the term $\sum_{i=1}^{n}\sum_{j=1}^{\kappa} r_{ij}\log\left(\frac{r_{ij}}{\pi_j}\right)$, must lie in a region where no $\pi_j = 0$ (in fact in a region where no $r_{ij} = 0$). As the arguments of the function are completely exchangeable and have equal importance, were say $\pi_1 = 0$, then it must be the case that $\pi_2 = 0$. The solution would thus reduce to $\pi_1 = \ldots = \pi_{\kappa-1} = 0$ and $\pi_\kappa = 1$.

The second term can be decomposed into n independent optimization problems. Moreover, the solution to each individual problem $\boldsymbol{R}^*_{i\cdot} = \min_{\boldsymbol{R}_{i\cdot}\in\Delta} \boldsymbol{R}^T_{i\cdot}\boldsymbol{b}_i$ is trivial:

$$r^*_{ij} = \begin{cases} 1, & \text{if } j = \arg\min_{1\le v\le\kappa}\{b_v\}; \\ 0, & \text{otherwise.} \end{cases}$$

In summary, the solutions do not cohere and to obtain a sparse solution to the optimization problem of (7.4) will require a value of θ large enough such that the second term completely dominates the first. The term is removed from the objective to remove the need for extensive computation.

7.5.3 Model Selection

Model selection for clustering with Gaussian mixture models usually involves the use of an information criterion, such as the AIC, BIC or the ICL as discussed in Section

2.1.2. The AIC and the BIC offer approximations to the log integrated likelihood using maximum likelihood estimates of the parameters. These criteria may overestimate the number of clusters in the data, however, if the clusters are cohesive and well separated, but without their distribution being Gaussian. If the clusters are better approximated by a mixture of Gaussian components, the number of components selected by the AIC and the BIC may be larger than the number of clusters (Fruhwirth-Schnatter et al., 2019). The ICL of Biernacki et al. (2003) aims to estimate the number of clusters directly by a BIC-like approximation to the integrated complete likelihood. By contrast, the ICL can lead to an underestimation of the number of clusters in the data, particularly if the data is arising from a mixture of poorly separated components.

Applying information criteria to select the number of components for the optimal clustering produced by REM is further complicated by fixing the mean parameters in the mixture at the exemplars. While the exemplars are consistent estimates of the high density regions of the data, and allow REM to avoid converging to solutions at which the likelihood is infinite, the theoretical guarantees surrounding the application of the AIC, BIC and ICL are not satisfied as the mean parameters are no longer maximum likelihood estimates.

REM provides one suggested clustering for each value of κ from which the user can choose based on substantive grounds. For a more automated procedure, several possibilities are available. A potential approach involves running a supplementary clustering procedure, taking the parameter values produced by REM at each iteration as the initial parameters for a run of the full EM algorithm, allowing the mean vectors to vary. Proceeding for the values of κ required, the solution selected among the clusterings produced by REM would be the clustering with the same number of components as the clustering chosen from the supplementary run using the AIC, BIC or ICL criteria. An alternative approach is to use an elbow rule on the plot displaying the value of the overlap for the pruned clusters at each iteration of the REM method against the number of clusters. This approach is similar to the method introduced in Baudry et al. (2010b), where an elbow rule is used to detect changes in the value of the penalty used

to merge cluster components. A third approach applies the information criteria to the clusterings produced by REM without adaptation. While lacking theoretical rigour, this strategy is shown below to perform well experimentally.

7.6 Evaluation

The performance of the REM algorithm is demonstrated on a range of synthetic and real-world classification datasets. We compare REM with popular and state-of-the-art initialization methods for mixture modeling with EM. The following methods are used for comparison:

Random Initialization (riEM) Random sampling from the data pool, and the EM algorithm is run to convergence for each. Implemented in the Scikit-Learn library (Pedregosa et al., 2011).

***k*-means++ Initialization (kmEM)** The k-means++ algorithm is used to provide initial partitions of the data. Implemented in the Scikit-Learn library (Pedregosa et al., 2011).

emEM (Biernacki et al., 2003) Truncated runs of EM with random initializations provide initial parameter values. Implemented as a wrapper for the Scikit-Learn library (Pedregosa et al., 2011).

rndEM (Maitra, 2009) Similar to emEM with truncated runs lasting only one iteration. Implemented as a wrapper for the Scikit-Learn library (Pedregosa et al., 2011).

mclust (Scrucca et al., 2016) Model-based hierarchical clustering provides initial partitions of the data. Implemented in R and C as part of the *mclust* package.

REM is implemented in Python and can be invoked similar to scikit-learn mixture methods. The code for REM, and code to reproduce the below experiments, is available online.[1]

[1] https://github.com/tobinjo96/Thesis-Experiments/

We report the execution time for each method to produce the optimal clustering and the number of clusters detected. We adopt the widely used Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) and Adjusted Mutual Information (AMI) (Vinh et al., 2010) for performance evaluation.

7.6.1 Experimental Setup

The bandwidth value in the Gaussian kernel is set to be the average of the distances from instances to their k-th nearest neighbor, where $k = \min(\sqrt{n}, 30)$. The impact of the bandwidth parameter on REM performance is further discussed in Section 7.6.5 below. For all the benchmark methods, the range for the number of clusters is required as an input. If κ exemplars are selected for a given dataset by REM, then for competitor methods, the cluster number m will range from 1 to $\kappa + 2$. For each value of m, riEM is initialized 25 times, kmEM is initialized 25 times, rndEM is initialized 200 times, emEM is initialized 50 times (with the maximum number of iterations of the truncated EM set to 50 and the tolerance for convergence of the log-likelihood set to 1×10^{-3}). Model selection for REM is completed using three information criteria, the AIC, BIC, and ICL computed using the parameter values provided by REM. A comparison of the other model selection methods discussed in Section 7.5.3 is also given. For riEM, kmEM, emEM, and rndEM, the AIC and BIC are used. For mclust, the authors note a preference for the BIC. To allow fair comparison between methods, we allow the covariance matrices to be free to take any particular form. The tolerance for convergence of the log-likelihood for EM is set to 1×10^{-5}, and the maximum number of iterations is set to 100 for each method. All experiments were conducted on a PC running Debian 10 (Buster), consisting of 24 cores and 24GB of RAM.

7.6.2 Simulated Datasets

We first examine the performance of the peak-finding method when no cluster structure is present in the data. Following Scrucca (2016a), we generate 40 datasets, 20 of dimension $p = 2$ and $p = 50$ respectively, from independent χ^2 distributions with

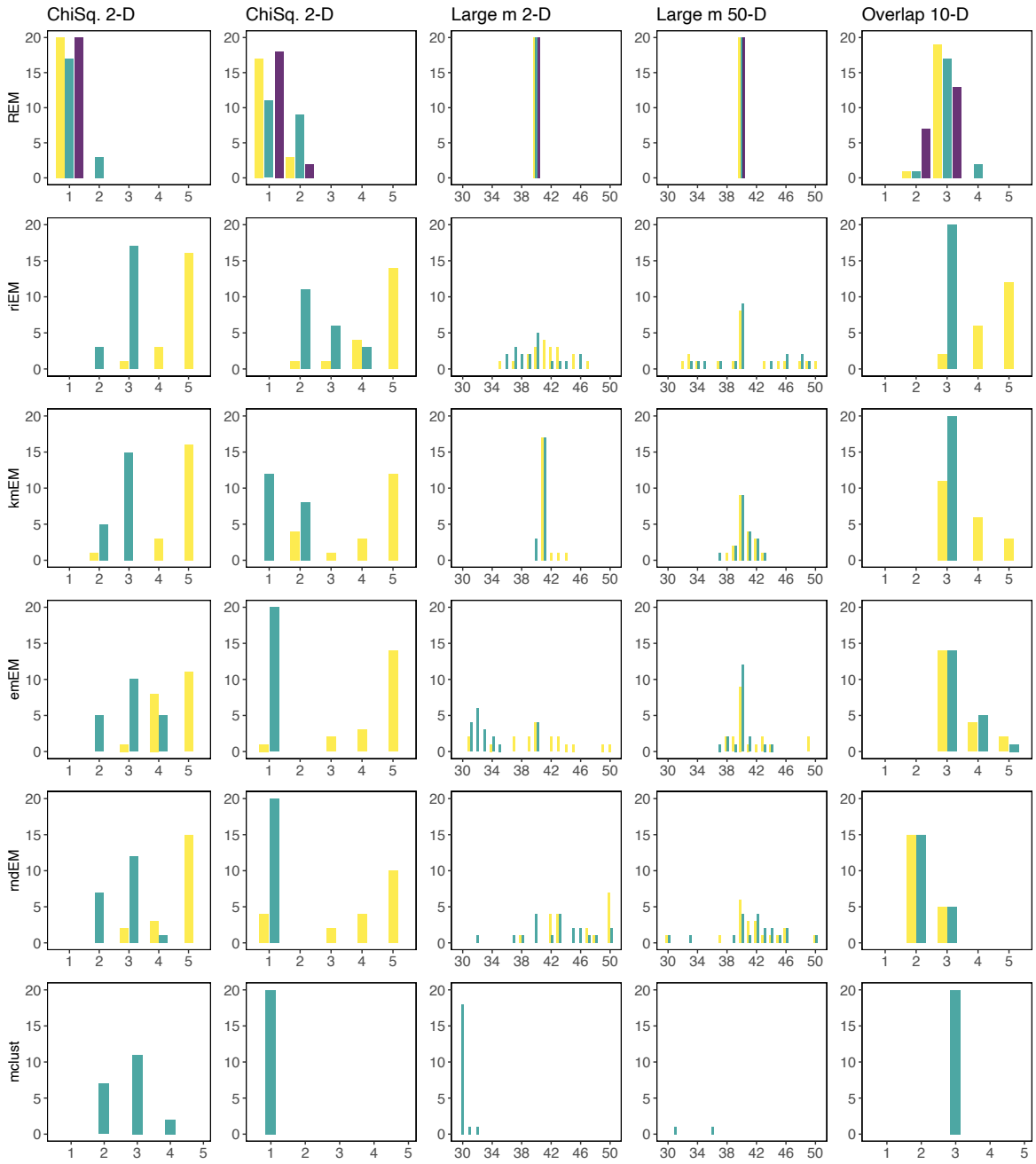


Figure 7.7: The number of clusters returned by each algorithm using the AIC (yellow), BIC (green), and ICL (purple) model selection criteria.

10 degrees of freedom. The competitor methods are initialized with $m \in [1, 5]$. The number of clusters returned by the competitor methods and REM for each of the datasets is shown in Figure 7.7 (the Chi-squared columns). The peak-finding method is adept at correctly choosing only one potential exemplar in the data. Since the data are skewed, the competitor methods detect more than one component in the data. This shows the efficiency of the peak-finding method for initialization.

We next consider datasets with a large number of well-separated clusters. Again, 40 datasets are generated, 20 each of dimension $p = 2$ and $p = 50$ respectively. The *MixSim* package in R is used to generate 40 clusters with no overlap. The competitor methods were initialized with $m \in [30, 50]$. The number of clusters returned by the competitor methods and REM for each of the datasets is shown in Figure 7.7 (the Large m columns). The advantages of deterministic initialization methods are clear in this case. In Figure 7.7, kmEM and REM achieve the correct number of components for $p = 2$, while the remaining stochastic initialization methods struggle to place initial mean vectors in every cluster. Due to the high number of parameters in these models, mclust returns a solution for only two of the 50-dimensional datasets. REM achieves the correct number of clusters for each model selection criterion.

Finally, we consider 20 simulations of the mixture provided in Experiment 3 of Melnykov and Michael (2020). This mixture consists of three overlapping clusters in 10 dimensions, two with adjacent mean vectors. The competitor methods are initialized with $m \in [1, 5]$. The number of clusters returned by the competitor methods and REM for each of the datasets is shown in Figure 7.7 (the Overlap column). In this case, REM achieves the correct number of components for the AIC and BIC model-selection criteria for each dataset. The ICL criterion merges the components with adjacent mean vectors, in keeping with its preference for sparser models relative to the other two criteria. For the competitor methods, mclust achieves the correct number of clusters for each dataset and the BIC is seen to generally outperform the AIC.

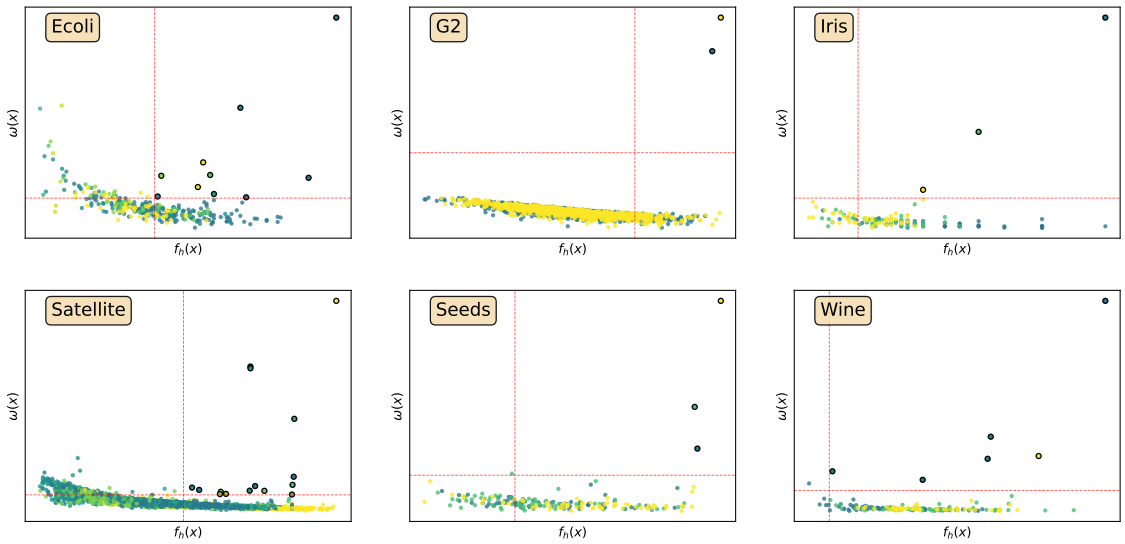


Figure 7.8: The decision plots for the six real datasets with color representing the true class label. The dashed lines show the values of τ and l that decide the initial exemplars.

7.6.3 Real Datasets

Six datasets are used to compare the clustering performance of REM with the competitor methods. Details of the datasets can be found in Table 7.1. Instances with missing values were removed. For each dataset, the initial exemplars for the REM algorithm are indicated in the decision plot in Figure 7.8. For the G2, Iris and Seeds datasets, the number of components is immediately obvious from the decision plot. For the other three datasets, we set the two threshold values to include all promising exemplars.

Source	**Name**	n	p	m
Dua and Graff (2019)	Ecoli	336	7	8
Mariescu-Istodor and Zhong (2016)	G2	2048	128	2
Dua and Graff (2019)	Iris	150	4	3
Dua and Graff (2019)	Satellite	4435	36	6
Dua and Graff (2019)	Seeds	210	7	3
Dua and Graff (2019)	Wine	178	14	3

Table 7.1: The characteristics of the evaluated datasets.

The results for clustering the six datasets are presented in Table 7.2. For each method, the ARI and AMI values are calculated from the clustering decided by the relevant

		REM			**riEM**		**kmEM**		**emEM**		**rndEM**		**Mclust**
Dataset	**Metric**	AIC	BIC	ICL	AIC	BIC	AIC	BIC	AIC	BIC	AIC	BIC	BIC
Ecoli	ARI	**0.623**	0.575	0.575	0.600	0.282	0.619	0.277	0.608	0.233	0.607	0.294	0.560
	AMI	**0.621**	0.572	0.572	0.601	0.382	0.598	0.370	0.596	0.275	0.574	0.385	0.551
G2	ARI	**1.000**	**1.000**	**1.000**	0.148	0.000	0.699	0.003	0.748	0.000	**1.000**	0.000	0.000
	AMI	**1.000**	**1.000**	**1.000**	0.185	0.004	0.723	0.051	0.764	0.000	**1.000**	0.000	0.000
Iris	ARI	**0.904**	**0.904**	**0.904**	0.742	0.443	0.693	0.693	0.856	0.568	**0.904**	0.568	0.562
	AMI	**0.900**	**0.900**	**0.900**	0.791	0.649	0.769	0.769	0.857	0.734	**0.900**	0.734	0.761
Satellite	ARI	**0.524**	0.429	0.429	0.443	0.419	0.444	0.465	0.423	0.439	0.465	0.476	0.467
	AMI	**0.578**	0.507	0.507	0.501	0.490	0.565	0.556	0.553	0.521	0.549	0.556	0.563
Seeds	ARI	0.766	0.766	0.766	0.594	0.663	0.576	0.621	0.624	0.624	0.644	0.663	**0.788**
	AMI	**0.744**	**0.744**	**0.744**	0.643	0.621	0.646	0.663	0.663	0.663	0.669	0.621	**0.744**
Wine	ARI	**0.534**	0.520	0.520	0.480	0.008	0.502	0.510	0.507	0.510	0.476	0.510	0.498
	AMI	0.526	**0.621**	**0.621**	0.559	0.072	0.508	0.608	0.584	0.608	0.520	0.601	0.613

Table 7.2: Clustering results on the real datasets by different methods. The best results are highlighted in bold.

Dataset	REM	riEM	kmEM	emEM	rndEM	Mclust
Ecoli	**1.42**	214.7	71.6	19.6	181.9	2.4
G2	**32.0**	2025.9	52.1	181.1	2433.0	1153.9
Iris	**0.3**	90.9	4.7	2.26	26.3	0.7
Satellite	**295.0**	7302.2	1161.9	1020.4	7242.5	437.7
Seeds	**0.9**	158.2	2.3	4.01	41.2	1.2
Wine	3.9	142.8	4.1	4.69	57.6	**0.4**

Table 7.3: Execution time (seconds) for the evaluated datasets.

model-selection criterion. REM achieves the best clustering, in terms of AMI, for every dataset examined and, in terms of ARI, the best for five of the six datasets. Moreover, the clustering results from REM are consistent across the different model-selection criteria. This is due to the fact that, in REM, the mean vectors are fixed at the exemplars. Mclust, the other deterministic initialization method assessed, outperforms REM for the Seeds dataset, and achieves comparable performance for the Satellite and Wine datasets. Unlike REM, which achieves the perfect clustering on the G2 dataset, Mclust is not able to detect the two clusters, as the BIC criterion merges the two components in search of a sparse model. The stochastic initialization approaches are capable of achieving excellent results, for example the rndEM on the Iris dataset. However, the performance is not consistent between different runs, and the results are not robust to the choice of the model-selection criterion.

The run time, in seconds, for each of the methods is presented in Table 7.3. For small datasets, we see that REM is competitive with Mclust, and both are much faster than the other methods. REM runs faster for the low-dimensional datasets, whereas Mclust is faster for the Wine dataset. The magnitude of difference is negligible and unlikely to hinder the use of REM in applications. We see that, for larger datasets, REM has the fastest run time. The difference is most pronounced for the G2 dataset and the Satellite dataset. The execution times of the stochastic methods are significantly higher than Mclust and REM. For riEM and rndEM, this is caused by the slow convergence of the EM algorithm as a result of naive initializations. For emEM and kmEM, providing the initializations requires significant computation, slowing down execution significantly.

7.6.4 Model Selection Methods

To assess the performance of the various model selection methods introduced in Section 7.5.3, we compare the quality of the clusterings chosen by each method for the six real-world datasets. We compare seven model selection methods: the three used thus far which input the clusterings produced by REM directly into the AIC, BIC and ICL respectively and are denoted as such; three that select the optimal clustering from those produced by REM as having the same number of clusters as that selected by the AIC, BIC, and ICL when computed using maximum likelihood parameter estimates taken from a supplementary full run of EM with the REM clusterings used as initializations, denoted AICsup, BICsup, ICLsup; and a heuristic elbow-rule approach that assesses the value of the cluster overlap for the pruned component at each iteration of the REM algorithm denoted Elbow. The elbow-rule plots are shown in Figure 7.9 and the results are given in Table 7.4.

The AIC exhibits the best performance of the model selection methods assessed, with the BIC and ICL method also performing well. These results indicate that the information criteria remain useful guides for detecting high-quality clusterings, even in the absence of maximum likelihood estimates for the mean parameters. For the Ecoli, G2, Iris and Seeds datasets, the results are relatively consistent across different model selection methods, with high-quality clusterings returned for each. It is for the Wine dataset that the greatest variation between the approaches is seen. There, the BICsup and ICLsup do not detect any cluster structure in the data while the remaining methods detect similar clusterings. Each method is able to return the the best or joint-best clustering for at least one of the datasets, and the choice of model selection method can be made by the user, balancing the practical reliability of the method, theoretical guarantees for the approach, and the degree of automation desired.

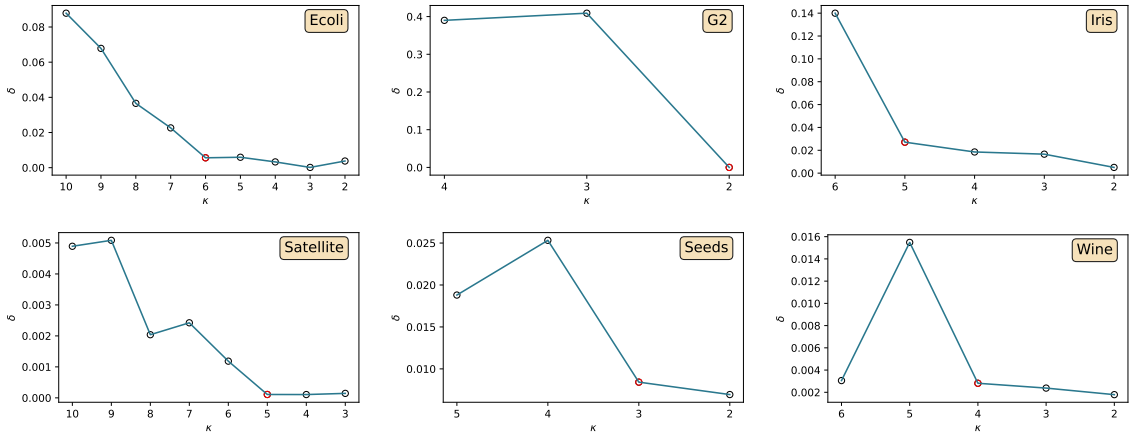


Figure 7.9: Elbow-rule plots for the clustering produced by REM for the six real-world datasets. The number of components chosen is highlighted in red.

	Metric	Ecoli	G2	Iris	Sat.	Seeds	Wine
AIC	ARI	**0.623**	**1.000**	**0.904**	**0.524**	**0.766**	**0.534**
	AMI	**0.621**	**1.000**	**0.900**	**0.578**	**0.744**	0.526
BIC	ARI	0.575	**1.000**	**0.904**	0.429	**0.766**	0.520
	AMI	0.572	**1.000**	**0.900**	0.507	**0.744**	**0.621**
ICL	ARI	0.575	**1.000**	**0.904**	0.429	**0.766**	0.520
	AMI	0.572	**1.000**	**0.900**	0.507	**0.744**	**0.621**
AICsup	ARI	0.575	**1.000**	0.775	0.524	0.600	**0.534**
	AMI	0.572	**1.000**	0.770	0.524	0.653	0.526
BICsup	ARI	0.575	0.659	**0.904**	0.429	0.659	0.000
	AMI	0.572	0.659	**0.900**	0.429	0.674	0.000
ICLsup	ARI	0.575	0.659	**0.904**	0.429	0.659	0.000
	AMI	0.572	0.659	**0.900**	0.429	0.674	0.000
Elbow	ARI	0.575	**1.000**	0.775	0.330	0.659	**0.534**
	AMI	0.575	**1.000**	0.770	0.447	0.674	0.526

Table 7.4: Clustering results on the real datasets for the model selection methods. The best results are highlighted in bold.

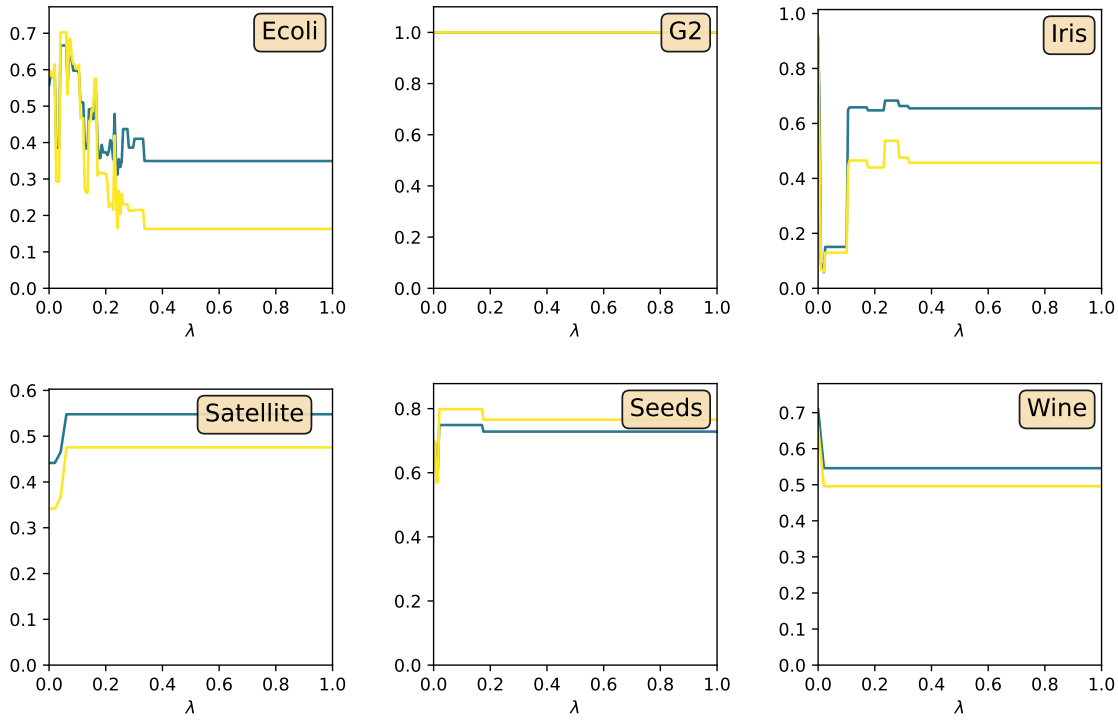


Figure 7.10: The performance of the REM algorithm evaluated by the ARI (yellow) and AMI (green) as the parameter h changes.

7.6.5 Ablation Study

REM has two tuning parameters: (1) h, the bandwidth of the Gaussian kernel and (2) κ, the number of exemplars selected from the decision graph. To examine the impact of h and κ on the REM algorithm, we assess the quality of the clusterings produced by REM for a broad range of parameter values.

To assess the impact of the bandwidth parameter, we compute the minimum (d_{min}) and maximum (d_{max}) pairwise distances in the data, and set $h = d_{min} + \lambda(d_{max} - d_{min})$. We increase λ from 0 to 1 in increments of 0.005 and run REM for each increment. We can see from Figure 7.10 that, for the Iris, Satellite and Seeds datasets, the results are robust to the choice of h, with REM achieving high quality clusterings over a very broad range of λ values. While the Ecoli dataset is seen to be relatively sensitive to the parameter choice, the range of high-quality values allows for λ to be selected between 0 and 0.3, a wide range for users to exploit. The other two datasets exhibit perfect consistency for all values of λ, emphasising the robustness of the proposed

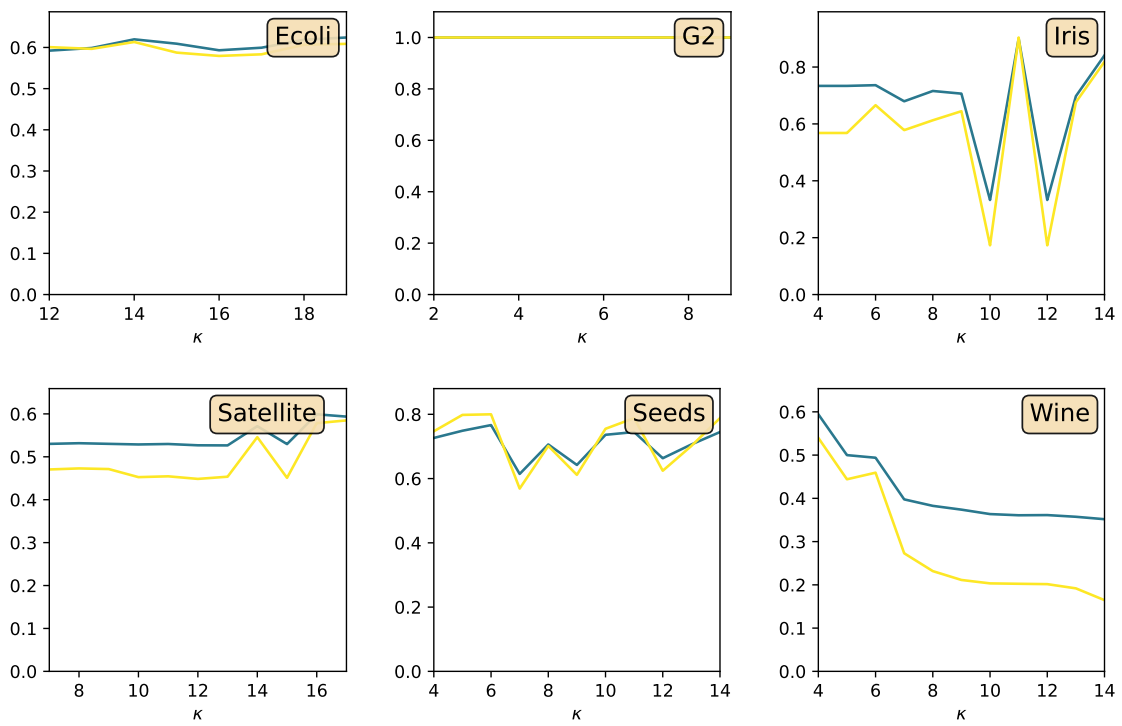


Figure 7.11: The performance of the REM algorithm evaluated by the ARI (yellow) and AMI (green) as the number of selected exemplars changes.

approach.

To assess the impact of κ on the clustering quality, we run REM on the datasets ten times, initialized with $\kappa \in [m, m + 10]$ centers. The selected exemplars are the κ instances with highest values of $\hat{f}_h(\boldsymbol{x}) \times \omega(\boldsymbol{x})$. The results, shown in Figure 7.11, demonstrate that the REM algorithm is robust to the number of initial exemplars. This confirms the intuition that the penalty introduced in Section 7.5.2 correctly prunes spurious exemplars. The Ecoli and G2 datasets are also robust to the choice of κ. While for the Wine dataset, the quality is seen to degrade as κ increases, it should be noted that the decision graph included in Figure 7.8 points users to values that return high-quality clusterings.

The importance of the main features of REM are demonstrated by individually substituting (1) the peak-finding exemplar selection with random selection, (2) the iterative pruning algorithm with an ordering of exemplars based on the magnitude of $f_h(\boldsymbol{x}) \times \omega(\boldsymbol{x})$, and (3) the EM-type algorithm with fixed means with the full EM al-

	Peaks	Prune	Fixed	ARI	AMI
M1		✓		0.237	0.283
M2	✓			0.571	0.541
M3	✓		✓	0.417	0.563
M4	✓	✓		0.601	0.598
REM	✓	✓	✓	**0.623**	**0.621**

Table 7.5: Clustering results on the Ecoli dataset by the ablation methods.

	Metric	G2	Iris	Sat.	Seeds	Wine
M1	ARI	0.431	0.795	0.516	0.475	0.383
	AMI	0.522	0.808	0.572	0.555	0.471
M2	ARI	**1.000**	0.899	0.463	0.640	0.463
	AMI	**1.000**	0.879	0.559	0.670	0.501
M3	ARI	0.000	0.664	0.492	0.641	0.456
	AMI	0.000	0.730	0.597	0.671	0.501
M4	ARI	**1.000**	0.870	**0.524**	0.715	0.000
	AMI	**1.000**	0.883	**0.578**	0.737	0.000
REM	ARI	**1.000**	**0.904**	**0.524**	**0.766**	**0.534**
	AMI	**1.000**	**0.900**	**0.578**	**0.744**	**0.536**

Table 7.6: Clustering results on the real datasets by ablation methods. The best results are highlighted in bold.

gorithm detailed in Section 7.4. The results for the Ecoli dataset are shown in Table 7.5 and the remaining datasets are given in Table 7.6. Random initialization leads to significantly poorer results than the peak-finding method as seen when comparing methods M1 and M2 in the Table 7.5. The impact of the pruning approach is also clear when comparing methods M2 and M4. Finally, it is noted that REM achieves the best result for each metric on each of the datasets assessed, highlighting the mutually beneficial impact of each of the constituent parts.

The results for the ablation study for the G2, Iris, Satellite, Seeds and Wine datasets are shown in Table 7.5. The strength of the REM method is again demonstrated. The peak-finding initializations are seen to significantly outperform the random initialization used in M1. Furthermore, the peak-finding initializations are seen to complement the pruning method introduced in this work.

7.7 Conclusion and Future Work

This chapter introduced REM, an algorithmic tool for model-based clustering, which extricates the EM algorithm from the initialization problem. We showed that the peak-finding method is an effective tool for quickly determining high-quality exemplars. For exemplar pruning, we developed a novel objective function that originates from the log-likelihood function, integrates a data-driven penalty, admits analytic solutions, and allows distributed computing. Through iterative pruning of the exemplars, our algorithm generates a sequence of nested clusterings, from which the preferred partition can be selected. Experimental results demonstrated that our method has excellent performance. It achieves perfect clusterings for datasets containing well-separated clusters and outperforms prominent benchmark methods on a broad range of simulated and real-world datasets. The hyper-parameters of our model are conventional and can be handily tuned by users. We showed that our algorithm achieves consistent results over a broad range of hyper-parameter values. In future, we envisage incorporating structured covariance matrices into our method to allow for even faster computation.

8 Conclusions

8.1 Summary

This thesis has analyzed the density peaks clustering algorithm, provided theoretical guarantees regarding its consistency, and developed three well-justified derivative methods which ameliorate deficiencies of the original algorithm while extending the range of its potential applications. This work has both broadened and deepened the body of existing literature on these topics.

8.1.1 Non-Parametric Clustering

The first analysis of the consistency of the estimates returned by the density peaks clustering algorithm has been produced. Density peaks clustering with a consistent density estimator returns consistent estimates of the modes of the underlying density with high probability. Furthermore, it was shown that if the modes are estimated sufficiently well, then the assignment strategy of density peaks clustering correctly clusters the instances in the data. Despite these guarantees, several issues were illustrated with the operation of the density peaks clustering algorithm for commonly sized datasets. It was shown that the density peaks clustering algorithm can erroneously detect spurious modes in the data in the presence of a noisy density estimate. Moreover, the allocation mechanism can produce results which do not meet the notion of a cluster used in non-parametric density-based clustering.

Aiming at remedying these issues, two novel non-parametric density-based clustering

methods were developed.

The first, termed DCF, seeks to model high density regions of the data using modal sets rather than point modes. By doing do, the algorithm is robust to fluctuations in the density estimate. Theoretical results showed that the set of modal set estimates returned by the DCF method can be bijectively related to true modal sets of the underlying density. The superiority of the DCF method over competitor approaches was demonstrated in an experimental analysis. The DCF method was also shown to be capable at an important computer vision application, namely unsupervised face detection.

The second method developed in this thesis, termed CPF, aims to improve the allocation mechanism of the density peaks clustering method. To eradicate the possibility of assigning instances to clusters over regions of very low density in the data, the data is first partitioned into density-level sets. This partition separates regions between which there are areas of low density. To each level set, a peak-finding clustering approach is applied, utilizing a procedure to prune spurious modes which is similar to that used in DCF. The benefits of combining level set and mode-seeking density-based clustering methods was demonstrated using an experimental analysis on simulated and real-world datasets. Subsequently, a modified version of the CPF method was introduced, incorporating instance-level clustering constraints to allows its application for a modern computer vision task, multi-image matching.

Both methods were included in an extensive experimental comparison, assessing the performance of prominent non-parametric density-based clustering methods on an expanded pool of real-world datasets. DCF and CPF were shown to significantly outperform competitor methods in terms of clustering quality. The effect of hyper-parameters on the clustering results was also assessed, demonstrating that the quality of the clusterings produced by DCF and CPF is not degraded for a broad range of parameters. Guidance for parameter selection was also discussed.

8.1.2 Parametric Clustering

The density peaks clustering method is shown to achieve high quality estimates of the modes. Considering the notion of a cluster in a parametric clustering as a unimodal component within an appropriate finite mixture model, a novel approach adapting the density peaks clustering methodology for mixture modelling was presented. The method, termed REM, selects an intial set of mode estimates using the peak-finding approach. These estimates are used as initializations for runs of the EM algorithm. REM uses a novel pruning algorithm to iteratively remove redundant exemplars from the pool. The pruning algorithm is well-justified in the context of mixture modelling and uses convex optimization to determine the exemplars to be pruned. The relevant theoretical results regarding mode recovery are discussed and analytical solutions to the optimization problem were also provided. To confirm the suitability of REM for parametric clustering, an experimental analysis using simulated and real-world data was completed. The results confirm that REM outperforms popular competitor methods and consistently recovers high quality parametric clusterings from the data.

8.1.3 Research Aims

Recalling the primary research aims discussed in the introduction, each has been achieved. The density peaks clustering algorithm has been theoretically analyzed for the first time, deepening understanding of its place in the field of non-parametric density-based clustering. Subsequently, two novel approaches have been developed which adapt and extend the density peaks clustering method. By detecting cluster cores in the data, the vulnerability to fluctuations in the density estimate was remedied. The allocation mechanism of density peaks clustering was also improved, reducing the likelihood of incoherent clustering outputs. The improvements offered by these new approaches have been extensively validated in experimentation and their performance in modern applications has been demonstrated. Moreover, we have proposed a novel direction for mode detection using the density peaks methodology, namely providing

stable estimates for the mean parameters in Gaussian mixture models. The density peaks clustering framework has, in this thesis, been shown to be a consistent, adaptable, and effective non-parametric density-based clustering method with the potential for quality application across a broad spectrum of data analysis tasks.

8.2 Further Work

8.2.1 Parametric Density Peaks Clustering

The REM method uses non-parametric estimates of the modes, provided by the density peaks clustering method, as initialization for the mean vector of a Gaussian mixture model. That the combination of parametric and non-parametric clustering methods proved fruitful motivates a proposal for a method that uses a parametric density estimate, obtained from a Gaussian mixture model, as an input to a peak-finding algorithm.

Assuming a parametric model for the density has several benefits: (1) the estimation process for the mixture density is well understood, following the work on model-based clustering outlined in Chapter 2; (2) the quality of the density estimation can be quantified using information criteria; (3) the form of the mixture components can be readily controlled by the used; and (4) the parametric density estimation process is accepted to scale better as the dimension of the data increases.

In Chacón (2016), a mixture density is first fit to the data and then a non-parametric clustering algorithm based on mean shift is applied to cluster the data. Similarly Scrucca (2016b) fit a mixture model before applying a non-parametric clustering approach, extracting level sets from the mixture density. Considering the improved performance of the density peaks clustering methods, compared to competitor level set and mode-seeking methods, demonstrated in this thesis, it is natural to consider the application of density peaks clustering using mixture densities as the estimator.

8.2.2 Density Peaks Clustering for High Dimensional Data

The effect of high dimensional data on the quality of the density peaks clustering method is a second potential topic of future study. In high-dimensional spaces, the curse of dimensionality can lead to all points becoming near equidistant from one another as the space becomes increasingly sparse. While the methods introduced in this thesis outperform the competitor methods for datasets of high dimension (for example see the performance of DCF and CPF for the Phonemes dataset and the performance of REM on the G2 dataset), in such a case methods based on distance between instances can fail.

The task of developing adaptations of the density peaks clustering method that provide quality clusterings of high-dimensional data offers many potential avenues of research. One could investigate density estimators designed to provide accurate estimates in high dimensions, such as those reviewed in Wang and Scott (2019). An alternate route would investigate the ability of dimension reduction techniques to provide mappings to lower-dimensional spaces while preserving the cluster structure uncovered by the peak-finding method. Such work leads naturally to the estimation of clusters that lie on lower dimensional manifolds within the data space. A brief review of such methods is available in Wasserman (2018), but as of yet no work has used the density peaks clustering method for this task. A related field is the task of detecting subspace clusters present in the data. Here, rather than seeking to project the data to a single low-dimensional subspace, as is done with many dimension reduction techniques, it is assumed that the data points are drawn from multiple subspaces. The task is to simultaneously cluster the data into multiple subspaces and find a low-dimensional subsapce fitting each group of points. In this situation, point modes are unlikely to adequately represent the clusters, but there is potential for the cluster core method used in DCF to determine true subspace clusters present in the data.

Finally, this formulation could be extended to be applicable for clustering functional data, with density peaks clustering methods applied once a suitable dissimilarity metric

and density estimator has be chosen.

8.2.3 Model Selection Criteria for Non-Parametric Clustering

The non-parametric methods introduced in this thesis, namely DCF and CPF, are hard clustering methods where each observation is assigned to one and only one cluster. By contrast, the mixture model method REM attempts to capture the uncertainty in the cluster assignment by producing an m assignment vector for each instance that is a probability distribution over the clusters. As discussed in Chen et al. (2016), soft clustering methods provide insights into two types of cluster uncertainty, at the sample level and at the population level. The sample level uncertainty reflects the uncertainty in the density estimate $\hat{f}$, and the population uncertainty captures the relationship between an observation and the true modes of the underlying density. For example, if an instance is on the border between the attraction regions of two modes, the assignment vector should reflect this uncertainty. Capturing and quantifying this uncertainty using soft assignments allows for the creating of model selection criteria for non-parametric clustering.

The work of Chen et al. (2016) develops a soft clustering adaptation for mean shift and uses it to develop a model selection criterion based on the connectivity among clusters. A worthwhile avenue of future research would be to develop a comparable method for density peaks clustering. Such an approach would develop further the theoretical results for the sample-based assignment mechanism of density peaks clustering to understand the uncertainty associated with each observation, accounting for the uncertainty introduced by the density estimator.

8.2.4 Sparse Covariance Structures and Non-Gaussian Mixture Models for REM

A simple extension of the REM method will implement the sparse covariance structures outlined in Chapter 2. This extension will bring the REM Python package closer to the suite of mixture model tools available in the popular mclust package. Integrating sparse covariance structures will also further improve the execution efficiency of the REM method.

The REM framework also has the potential to be extended to mixtures of multivariate Gaussian components. As discussed in Chapter 2, the mixture model framework has been applied for mixtures of multivariate-t distributions, skew-normal distributions, skew-t distributions, among others. For each of these component distributions, the modes of the mixture density are symptomatic of the underlying population structure. The consistent estimates provided by the density peaks clustering method could provide high quality initializations as is done for Gaussian mixtures. It should be noted that the extension to mixture of non-Gaussian components will require the pruning methodology to be reformulated, as it is currently justified only in the context of Gaussian mixtures. Such a goal could reasonably achieved and would provide an interesting addition to the literature.

8.2.5 Riemannian Optimization for Gaussian Mixture Models

Hosseini and Sra (2020) develop an alternative estimation approach for Gaussian mixture models using Riemannian optimization. Using a clever reparameterization of the mean vectors and covariance matrices, the maximum likelihood estimates are found through manifold optimization on the product space of the m p-dimensional positive definite matrices, $\prod_{j=1}^{m} \mathbb{P}^p$, and the $(m-1)$-dimensional product vector, $\mathbb{R}^{(m-1)}$. This approach is shown to converge faster than EM and produce higher quality parameter

estimates. Unfortunately, as a result of the reparameterization, the sparse covariance structures used for Gaussian mixtures can not be integrated into their proposal.

It was shown in Chapter 7 that the density peaks clustering algorithm can provide high quality estimates for the mean vectors of a mixture model, leaving only the covariance matrices to be estimated. In this case, manifold optimization methods over the space of positive definite matrices could be used to produce higher quality results more quickly than the EM algorithm. Moreover, the sparse covariance structures for Gaussian mixture models can be easily integrated into this proposal. In fact, the manifold optimization becomes significantly easier as rather than optimizing over the product space of positive definite matrices, the optimization will be over a reduced space depending on the particular covariance constraints considered. For example, if no constraints are adopted for m covariance matrices of size $p \times p$, the optimization is over the space $\prod_{j=1}^{m} \mathbb{P}^p \times \mathbb{R}^{(m-1)}$. Instead, if the covariance matrices are restricted to be diagonal, the space becomes $\prod_{j=1}^{m} \mathbb{R}^p \times \mathbb{R}^{(m-1)}$, or if they are common across each component, the optimization is over $\mathbb{P}^p \times \mathbb{R}^{(m-1)}$. A key benefit of this approach is that simpler covariance structures are reflected in simpler and more efficient estimation procedures.

Bibliography

Abraham, C., Biau, G., and Cadre, B. (2004). On the asymptotic properties of a simple estimate of the mode. *ESAIM: Probability and Statistics*, 8:1–11.

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723. Conference Name: IEEE Transactions on Automatic Control.

Améndola, C., Engström, A., and Haase, C. (2020). Maximum Number of Modes of Gaussian Mixtures. *Information and Inference: A Journal of the IMA*, 9(3):587–600. arXiv: 1702.05066.

Arias-Castro, E., Mason, D., and Pelletier, B. (2016). On the Estimation of the Gradient Lines of a Density and the Consistency of the Mean-Shift Algorithm. *Journal of Machine Learning Research*, 17(43):1–28.

Ayvaci, A. and Soatto, S. (2009). Motion segmentation with occlusions on the superpixel graph. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 727–734. IEEE.

Azzalini, A. and Torelli, N. (2007). Clustering via nonparametric density estimation. *Statistics and Computing*, 17(1):71–80.

Banfield, J. D. and Raftery, A. E. (1993). Model-Based Gaussian and Non-Gaussian Clustering. *Biometrics*, 49(3):803–821. Publisher: [Wiley, International Biometric Society].

Baudry, J.-P., Raftery, A. E., Celeux, G., Lo, K., and Gottardo, R. (2010a). Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, 19(2):332–353.

Baudry, J.-P., Raftery, A. E., Celeux, G., Lo, K., and Gottardo, R. (2010b). Combining mixture components for clustering. *Journal of computational and graphical statistics*, 19(2):332–353.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300.

Bernard, F., Thunberg, J., Swoboda, P., and Theobalt, C. (2019). HiPPI: Higher-Order Projected Power Iterations for Scalable Multi-Matching. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10283–10292, Seoul, Korea (South). IEEE.

Biau, G., Chazal, F., Cohen-Steiner, D., Devroye, L., and Rodriguez, C. (2011). A weighted k-nearest neighbor density estimate for geometric inference. *Electronic Journal of Statistics*, 5:204–237.

Biernacki, C., Celeux, G., and Govaert, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3):561 – 575.

Bishop, C. (2006). *Pattern recognition and machine learning*. Information science and statistics. Springer-Verlag New York, 1 edition.

Bouveyron, C., Celeux, G., Murphy, T. B., and Raftery, A. E. (2019). *Model-based clustering and classification for data science: with applications in R*, volume 50. Cambridge University Press.

Campello, R. J., Kröger, P., Sander, J., and Zimek, A. (2020). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(2):e1343.

Campello, R. J. G. B., Moulavi, D., and Sander, J. (2013). Density-Based Clustering Based on Hierarchical Density Estimates. In Pei, J., Tseng, V. S., Cao, L., Motoda, H., and Xu, G., editors, *Advances in Knowledge Discovery and Data Mining*, Lecture Notes in Computer Science, pages 160–172, Berlin, Heidelberg. Springer.

Carlsson, G. and Mémoli, F. (2013). Classifying Clustering Schemes. *Foundations of Computational Mathematics*, 13(2):221–252.

Carmichael, J. W. and Julius, R. S. (1968). Finding Natural Clusters. *Systematic Biology*, 17(2):144–150.

Casa, A. (2019). Climbing modes and exploring mixtures: a journey in density-based clustering.

Celeux, G., Frühwirth-Schnatter, S., and Robert, C. P. (2019). Model selection for mixture models–perspectives and strategies. In *Handbook of mixture analysis*, pages 117–154. Chapman and Hall/CRC.

Celeux, G. and Govaert, G. (1995). Gaussian parsimonious clustering models. *Pattern Recognition*, 28(5):781–793.

Chacón, J. E. and Duong, T. (2018). *Multivariate kernel smoothing and its applications*. Chapman and Hall/CRC.

Chacón, J. E. (2015). A Population Background for Nonparametric Density-Based Clustering. *Statistical Science*, 30(4):518–532. Publisher: Institute of Mathematical Statistics.

Chacón, J. E. (2016). Mixture model modal clustering. *arXiv:1609.04721 [stat]*. arXiv: 1609.04721.

Chaudhuri, K. and Dasgupta, S. (2010). Rates of convergence for the cluster tree. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, pages 343–351, Red Hook, NY, USA. Curran Associates Inc.

Chaudhuri, K., Dasgupta, S., Kpotufe, S., and von Luxburg, U. (2014). Consistent Procedures for Cluster Tree Estimation and Pruning. *IEEE Transactions on Information Theory*, 60(12):7900–7912. Conference Name: IEEE Transactions on Information Theory.

Chen, B., Ting, K. M., Washio, T., and Zhu, Y. (2018). Local contrast as an effective means to robust clustering against varying densities. *Machine Learning*, 107(8):1621–1645.

Chen, Y.-C., Genovese, C. R., and Wasserman, L. (2016). A comprehensive approach to mode clustering. *Electronic Journal of Statistics*, 10(1):210–241. Publisher: Institute of Mathematical Statistics and Bernoulli Society.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799. Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence.

Comaniciu, D. and Peter, M. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:281–288.

Dasgupta, S. (1999). Learning Mixtures of Gaussians. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS '99, page 634, USA. IEEE Computer Society.

Dasgupta, S. and Kpotufe, S. (2014). Optimal rates for k-NN density and mode estimation. *Advances in Neural Information Processing Systems*, 27.

Davies, R. B. (1980). Algorithm AS 155: The Distribution of a Linear Combination of χ^2 Random Variables. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29(3):323–333. Publisher: [Wiley, Royal Statistical Society].

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38. Publisher: [Royal Statistical Society, Wiley].

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30.

Devroye, L. (1979). Recursive estimation of the mode of a multivariate density. *Canadian Journal of Statistics*, 7(2):159–167.

Ding, J., He, X., Yuan, J., and Jiang, B. (2018). Automatic clustering based on density peak detection using generalized extreme value distribution. *Soft Computing*, 22(9):2777–2796.

Ding, S., Du, M., Sun, T., Xu, X., and Xue, Y. (2017). An entropy-based density peaks clustering algorithm for mixed type data employing fuzzy neighborhood. *Knowledge-Based Systems*, 133:294–313.

Domany, E. (1999). Superparamagnetic clustering of data — the definitive solution of an ill-posed problem. *Physica A: Statistical Mechanics and its Applications*, 263(1):158–169. Proceedings of the 20th IUPAP International Conference on Statistical Physics.

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). 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*, KDD'96, pages 226–231, Portland, Oregon. AAAI Press.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, 32(200):675–701.

Fruhwirth-Schnatter, S., Celeux, G., and Robert, C. P. (2019). *Handbook of mixture analysis*. CRC press.

Fukunaga, K. and Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40. Conference Name: IEEE Transactions on Information Theory.

Fulkerson, B., Vedaldi, A., and Soatto, S. (2009). Class segmentation and object localization with superpixel neighborhoods. In *2009 IEEE 12th international conference on computer vision*, pages 670–677. IEEE.

Gan, J. and Tao, Y. (2015). DBSCAN Revisited: Mis-Claim, Un-Fixability, and Approximation. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 519–530.

Garcia, S. and Herrera, F. (2008). An extension on" statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of machine learning research*, 9(12).

Guo, Y., Zhang, L., Hu, Y., He, X., and Gao, J. (2016). MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition. *arXiv:1607.08221 [cs]*.

Hartigan, J. A. (1975). *Clustering Algorithms*. Wiley. Google-Books-ID: cDnvAAAAMAAJ.

Hartigan, J. A. (1981). Consistency of Single Linkage for High-Density Clusters. *Journal of the American Statistical Association*, 76(374):388–394. Publisher: [American Statistical Association, Taylor & Francis, Ltd.].

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Science & Business Media. Google-Books-ID: tVIjmNS3Ob8C.

Hosseini, R. and Sra, S. (2020). An alternative to EM for Gaussian mixture models: Batch and stochastic Riemannian optimization. *Mathematical Programming*, 181(1):187–223.

Hou, J. and Pelillo, M. (2016). A new density kernel in density peak based clustering. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 468–473.

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

Jiang, H. (2017a). Density Level Set Estimation on Manifolds with DBSCAN. In *International Conference on Machine Learning*, pages 1684–1693. PMLR. ISSN: 2640-3498.

Jiang, H. (2017b). On the Consistency of Quick Shift. *Advances in Neural Information Processing Systems*, 30.

Jiang, H. (2017c). Uniform Convergence Rates for Kernel Density Estimation. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1694–1703. PMLR. ISSN: 2640-3498.

Jiang, H., Jang, J., and Kpotufe, S. (2018). Quickshift++: Provably Good Initializations for Sample-Based Mean Shift. In *International Conference on Machine Learning*, pages 2294–2303. PMLR. ISSN: 2640-3498.

Jiang, H. and Kpotufe, S. (2017). Modal-set estimation with an application to clustering. In *Artificial Intelligence and Statistics*, pages 1197–1206. PMLR. ISSN: 2640-3498.

Jiang, J., Chen, Y., Meng, X., Wang, L., and Li, K. (2019). A novel density peaks clustering algorithm based on k nearest neighbors for improving assignment process. *Physica A: Statistical Mechanics and its Applications*, 523:702–713.

Jin, C., Zhang, Y., Balakrishnan, S., Wainwright, M. J., and Jordan, M. I. (2016). Local maxima in the likelihood of gaussian mixture models: Structural results and

algorithmic consequences. In *Advances in neural information processing systems 29*, pages 4116–4124.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of the american statistical association*, 90(430):773–795.

Kpotufe, S. and von Luxburg, U. (2011). Pruning nearest neighbor cluster trees. *arXiv:1105.0540 [cs, stat]*. arXiv: 1105.0540.

Li, X. and Wong, K.-C. (2018). Evolutionary multiobjective clustering and its applications to patient stratification. *IEEE transactions on cybernetics*, 49(5):1680–1693.

Li, Z. and Tang, Y. (2018). Comparative density peaks clustering. *Expert Systems with Applications*, 95:236–247.

Liu, R., Wang, H., and Yu, X. (2018). Shared-nearest-neighbor-based clustering by fast search and find of density peaks. *Information Sciences*, 450:200–226.

Liu, Y., Li, Z., Xiong, H., Gao, X., and Wu, J. (2010). Understanding of internal clustering validation measures. In *2010 IEEE international conference on data mining*, pages 911–916. IEEE.

Loftsgaarden, D. O. and Quesenberry, C. P. (1965). A Nonparametric Estimate of a Multivariate Density Function. *The Annals of Mathematical Statistics*, 36(3):1049–1051. Publisher: Institute of Mathematical Statistics.

Lowe, D. (1999). Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2.

Lu, K., Xia, S., and Xia, C. (2015). Clustering based road detection method. In *2015 34th Chinese Control Conference (CCC)*, pages 3874–3879. IEEE.

Lulli, A., Dell'Amico, M., Michiardi, P., and Ricci, L. (2016). NG-DBSCAN: scalable density-based clustering for arbitrary data. *Proceedings of the VLDB Endowment*, 10(3):157–168.

Ma, J., Jiang, X., Fan, A., Jiang, J., and Yan, J. (2021). Image Matching from Handcrafted to Deep Features: A Survey. *International Journal of Computer Vision*, 129(1):23–79.

Maier, M., Hein, M., and von Luxburg, U. (2009). Optimal construction of k-nearest-neighbor graphs for identifying noisy clusters. *Theoretical Computer Science*, 410(19):1749–1764.

Maitra, R. (2009). Initializing Partition-Optimization Algorithms. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(1):144–157. Conference Name: IEEE/ACM Transactions on Computational Biology and Bioinformatics.

Maitra, R. and Melnykov, V. (2010). Simulating Data to Study Performance of Finite Mixture Modeling and Clustering Algorithms. *Journal of Computational and Graphical Statistics*, 19(2):354–376. Publisher: Taylor & Francis _eprint: https://doi.org/10.1198/jcgs.2009.08054.

Mariescu-Istodor, P. F. R. and Zhong, C. (2016). Xnn graph. LNCS 10029:207–217.

McNicholas, P. D. (2016a). *Mixture Model-Based Classification*. Chapman and Hall/CRC, Boca Raton.

McNicholas, P. D. (2016b). Model-based clustering. *Journal of Classification*, 33(3):331–373.

Meilă, M. (2007). Comparing clusterings—an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895.

Melnykov, V., Chen, W.-C., and Maitra, R. (2012). MixSim: An R Package for Simulating Data to Study Performance of Clustering Algorithms. *Journal of Statistical Software*, 51:1–25.

Melnykov, V. and Maitra, R. (2010). Finite mixture models and model-based clustering. *Statistics Surveys*, 4(none):80–116. Publisher: Amer. Statist. Assoc., the Bernoulli Soc., the Inst. Math. Statist., and the Statist. Soc. Canada.

Melnykov, V. and Michael, S. (2020). Clustering Large Datasets by Merging K-Means Solutions. *Journal of Classification*, 37(1):97–123.

Menardi, G. (2015). A Review on Modal Clustering. *International Statistical Review*, 84.

Menardi, G. (2016). A Review on Modal Clustering. *International Statistical Review*, 84(3):413–433. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/insr.12109.

Moore, D. S. and Yackel, J. W. (1977). Consistency Properties of Nearest Neighbor Density Function Estimators. *The Annals of Statistics*, 5(1):143–154. Publisher: Institute of Mathematical Statistics.

Parzen, E. (1962). On Estimation of a Probability Density Function and Mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076. Publisher: Institute of Mathematical Statistics.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830.

Platero-Rochart, D., González-Alemán, R., Hernández-Rodríguez, E. W., Leclerc, F., Caballero, J., and Montero-Cabrera, L. (2022). Rcdpeaks: memory-efficient density peaks clustering of long molecular dynamics. *Bioinformatics*, 38(7):1863–1869.

Ray, S. and Lindsay, B. G. (2005). The topography of multivariate normal mixtures. *The Annals of Statistics*, 33(5):2042–2065.

Rinaldo, A. and Wasserman, L. (2010). Generalized density clustering. *The Annals of Statistics*, 38(5). arXiv: 0907.3454.

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

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The annals of mathematical statistics*, pages 832–837.

Schwarz, G. (1978). Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464. Publisher: Institute of Mathematical Statistics.

Scott, D. W. (2015). *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons.

Scrucca, L. (2016a). Identifying connected components in Gaussian finite mixture models for clustering. *Computational Statistics & Data Analysis*, 93:5–17.

Scrucca, L. (2016b). Identifying connected components in Gaussian finite mixture models for clustering. *Computational Statistics & Data Analysis*, 93:5–17.

Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. (2016). mclust 5: Clustering, Classification and Density Estimation Using Gaussian Finite Mixture Models. *The R Journal*, 8(1):289–317.

Sheikh, Y. A., Khan, E. A., and Kanade, T. (2007). Mode-seeking by Medoidshifts. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. ISSN: 2380-7504.

Sieranoja, S. and Fränti, P. (2019). Fast and general density peaks clustering. *Pattern Recognition Letters*, 128:551–558.

Song, H. and Lee, J.-G. (2018). RP-DBSCAN: A Superfast Parallel DBSCAN Algorithm Based on Random Partitioning. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, pages 1173–1187, New York, NY, USA. Association for Computing Machinery.

Sriperumbudur, B. and Steinwart, I. (2012). Consistency and rates for clustering with dbscan. In *Artificial Intelligence and Statistics*, pages 1090–1098. PMLR.

Steinwart, I. (2011). Adaptive Density Level Set Clustering. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 703–738. JMLR Workshop and Conference Proceedings. ISSN: 1938-7228.

Strehl, A. and Ghosh, J. (2002). Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research*, 3:583–617.

Stuetzle, W. (2003). Estimating the Cluster Tree of a Density by Analyzing the Minimal Spanning Tree of a Sample. *Journal of Classification*, 20(1):025–047.

Stuetzle, W. and Nugent, R. (2010). A Generalized Single Linkage Method for Estimating the Cluster Tree of a Density. *Journal of Computational and Graphical Statistics*, 19(2):397–418. Publisher: [American Statistical Association, Taylor & Francis, Ltd., Institute of Mathematical Statistics, Interface Foundation of America].

Tiedeman, D. V. (1955). On the study of types. *Symposium on Pattern Analysis*, pages 1–14. In S.B. Sells (Ed.).

Tron, R., Zhou, X., Esteves, C., and Daniilidis, K. (2017). Fast Multi-Image Matching via Density-Based Clustering. pages 4057–4066.

Tsybakov, A. B. (1997). On nonparametric estimation of density level sets. *The Annals of Statistics*, 25(3).

Vassilvitskii, S. and Arthur, D. (2006). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035.

Vedaldi, A. and Soatto, S. (2008). Quick Shift and Kernel Methods for Mode Seeking. In Forsyth, D., Torr, P., and Zisserman, A., editors, *Computer Vision – ECCV 2008*, Lecture Notes in Computer Science, pages 705–718, Berlin, Heidelberg. Springer.

Verbeek, J. J., Vlassis, N., and Kröse, B. (2003). Efficient greedy learning of Gaussian mixture models. *Neural Computation*, 15(2):469–485.

Verdinelli, I. and Wasserman, L. (2018). Analysis of a mode clustering diagram. *Electronic Journal of Statistics*, 12(2):4288–4312. Publisher: Institute of Mathematical Statistics and Bernoulli Society.

Vinh, N. X., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:2837–2854.

Wang, D., Lu, X., and Rinaldo, A. (2017). Optimal rates for cluster tree estimation using kernel density estimators. *arXiv preprint arXiv:1706.03113*.

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., and Liu, W. (2018a). CosFace: Large Margin Cosine Loss for Deep Face Recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274. ISSN: 2575-7075.

Wang, Q., Zhou, X., and Daniilidis, K. (2018b). Multi-image Semantic Matching by Mining Consistent Features. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 685–694, Salt Lake City, UT, USA. IEEE.

Wang, X.-F. and Xu, Y. (2017). Fast clustering using adaptive density peak detection. *Statistical Methods in Medical Research*, 26(6):2800–2811.

Wang, Y., Gu, Y., and Shun, J. (2020a). Theoretically-Efficient and Practical Parallel DBSCAN. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 2555–2571.

Wang, Y., Wang, D., Zhang, X., Pang, W., Miao, C., Tan, A.-H., and Zhou, Y. (2020b). McDPC: multi-center density peak clustering. *Neural Computing and Applications*.

Wang, Y., Wei, Z., and Yang, J. (2018c). Feature trend extraction and adaptive density peaks search for intelligent fault diagnosis of machines. *IEEE Transactions on Industrial Informatics*, 15(1):105–115.

Wang, Z. and Scott, D. W. (2019). Nonparametric density estimation for high-dimensional data—algorithms and applications. *Wiley Interdisciplinary Reviews: Computational Statistics*, 11(4):e1461.

Wasserman, L. (2018). Topological Data Analysis. *Annual Review of Statistics and Its Application*, 5(1):501–532. _eprint: https://doi.org/10.1146/annurev-statistics-031017-100045.

Wasserman, L., Azizyan, M., and Singh, A. (2014). Feature selection for high-dimensional clustering. *arXiv preprint arXiv:1406.2240*.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics*, 1:80–83.

Wishart, D. (1969). Mode analysis: A generalization of nearest neighbor which reduces chaining effects. *Numerical Taxonomy*, pages 282–311.

Wolf, L., Hassner, T., and Maoz, I. (2011). Face recognition in unconstrained videos with matched background similarity. In *CVPR 2011*, pages 529–534.

Wolfe, J. H. (1963). Object cluster analysis of social areas.

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

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

Yan, J., Ren, Z., Zha, H., and Chu, S. (2016). A constrained clustering based approach for matching a collection of feature sets. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 3832–3837.

Yan, J., Xu, H., Zha, H., Yang, X., Liu, H., and Chu, S. (2015). A Matrix Decomposition Perspective to Multiple Graph Matching. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 199–207, Santiago, Chile. IEEE.

Yang, L., Zhan, X., Chen, D., Yan, J., Loy, C. C., and Lin, D. (2019). Learning to Cluster Faces on an Affinity Graph. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2293–2301.

Yaohui, L., Zhengming, M., and Fang, Y. (2017). Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy. *Knowledge-Based Systems*, 133:208–220.

Yu, D., Liu, G., Guo, M., Liu, X., and Yao, S. (2019). Density peaks clustering based on weighted local density sequence and nearest neighbor assignment. *Ieee Access*, 7:34301–34317.