Low-Density Cluster Separators for Large, High-Dimensional, Mixed and Non-Linearly Separable Data.

Submitted by
Katie R Yates B.Sc.(Hons.), M.Res
to
Lancaster University

for the degree of
Doctor of Philosophy
in the subject of
Statistics and Operational Research

October 2017

Low-Density Cluster Separators for Large, High-Dimensional, Mixed and Non-Linearly Separable Data.

Katie R Yates Bsc. (Hons), MRes.

Submitted for the degree of Doctor of Philosophy at Lancaster University.

October 2017

Abstract

The location of groups of similar observations (clusters) in data is a well-studied problem, and has many practical applications. There are a wide range of approaches to clustering, which rely on different definitions of similarity, and are appropriate for datasets with different characteristics. Despite a rich literature, there exist a number of open problems in clustering, and limitations to existing algorithms.

This thesis develops methodology for clustering high-dimensional, mixed datasets with complex clustering structures, using low-density cluster separators that bi-partition datasets using cluster boundaries that pass through regions of minimal density, separating regions of high probability density, associated with clusters. The bi-partitions arising from a succession of minimum density cluster separators are combined using divisive hierarchical and partitional algorithms, to locate a complete clustering, while estimating the number of clusters.

The proposed algorithms locate cluster separators using one-dimensional arbitrarily oriented subspaces, circumventing the challenges associated with clustering in high-dimensional spaces. This requires continuous observations; thus, to extend the applicability of the proposed algorithms to mixed datasets, methods for producing an appropriate continuous representation of datasets containing non-continuous features are investigated. The exact evaluation of the density intersected by a cluster boundary is restricted to linear separators. This limitation is lifted by a non-linear mapping of the original observations into a feature

space, in which a linear separator permits the correct identification of non-linearly separable clusters in the original dataset.

In large, high-dimensional datasets, searching for one-dimensional subspaces, which result in a minimum density separator is computationally expensive. Therefore, a computationally efficient approach to low-density cluster separation using approximately optimal projection directions is proposed, which searches over a collection of one-dimensional random projections for an appropriate subspace for cluster identification. The proposed approaches produce high-quality partitions, that are competitive with well-established and state-of-the-art algorithms.

To David for all your love and support throughout my PhD. and every other aspect of my life for the past 10 years.

To Mum and Dad. You have always been there for me in every way possible, and I could not have achieved a fraction of what I have without your never ending love, guidance and encouragement.

To Karen for being the best sister and role model. Your achievements have always encouraged me to follow in your footsteps.

To Nanna for always encouraging me to learn. For as long as I can remember, you were always there to ask me questions and teach me new things.

To Grandma. You always made it possible for me to achieve my dreams.

Acknowledgements

I would like to thank my supervisors Nicos Pavlidis and Chris Sherlock. This project was started under difficult circumstances, and without your support I would not have made it past the first year of my PhD. Nicos, your enthusiasm and words of wisdom have got me through the last three years and I have learnt more from you than you would ever take credit for. From helping me with my writing to debugging my code with me, I could not have asked for a more dedicated supervisor. I would also like to thank Sotiris Tasoulis for his contributions to Chapter 6 of this thesis.

I would also like to thank all the staff and students at the STOR-i centre for doctoral training. Having you there through difficult times to talk about the stresses of PhD. life over a cup of tea or lunch has made this whole experience much less intimidating. I would especially like to thank David Hofmeyer, for your academic contributions to Chapter 4 of this thesis, and also for your words of advice through the first two years of my PhD.

Thank you to my very dear friends Helen and Lucy for always being there with a smile to make even the worst days in the office better. From singing along to our music in the car to our Euro-vision parities, I could not have asked for better friends to go through my time at Lancaster with.

Finally, I am very grateful for the financial support provided by the EPSRC.

Declaration

I declare that the work in this thesis has been done by myself and has not been submitted elsewhere for the award of any other degree.

All the work produced in this thesis was done so in collaboration with my supervisor Nicos Pavlidis. The work in Chapter 4 and Chapter 6 also involved collaboration with David Hofmeyr and Sotiris Tasoulis respectively.

Preliminary work from Chapter 5 is published as Yates, K. R. and Pavlidis, N. G. (2016). Minimum density hyperplanes in the feature space. *In Big Data (Big Data), 2016 IEEE International Conference on*, pages 3613–3618. IEEE.

The work included in all three main chapters of this thesis is in preparation for submission as journal papers.

Katie R Yates

Word count : 63,686

Contents

List of Figures

List of Tables

1

Introduction

The task of locating groups of related objects in data is a well studied problem in machine learning, statistics, data mining and pattern recognition. This has a number of practical applications including:

- Business and marketing : In market research, it is useful to partition the population of customers into groups with similar buying habits to infer relationships between them, assess strategic opportunities and identify competitive threats (Hruschka, 1986). In marketing and advertising, recommender systems require groups of similar products and customers allowing targeted marketing strategies where similar items are recommended to similar customers (Hameed et al., 2012).
- Computer science : Image segmentation can be used to divide a digital image into smaller segments which can be used for object recognition and border detection (Zhang, 1996). Web searching relies on grouping web pages with similar content to help locate the most relevant results as quickly as possible (Beeferman and Berger, 2000).
- Security : Learning groups of individuals with similar behaviour, for example spending patterns, allows the detection of network intrusions and potentially malicious behaviour (Portnoy et al., 2001).
- Biology and medicine : Monitoring the level of expression of groups of genes over time allows the understanding of the roles of different genes (Zhao and Karypis, 2005). In medical imaging, identifying regions of different tissue types is used to identify different tumours and assess the effect of treatments (Masulli and Schenone, 1999).
- Physical sciences : In astrophysics, grouping objects allows the detection of regions of interest such as galaxies and gas clouds (Zentner et al., 2005).

The type of learning problem is defined by the amount of information available to train the categorisation process. In *supervised learning* or *classification*, a training set of observa-

tions with associated class labels is used to construct the predictive model for the subsequent grouping of unlabelled observations (Theodoridis and Koutroumbas, 2008). In many applications, knowledge of the true class labels may be expensive or impossible to obtain. In the absence of such information, the problem becomes one of *unsupervised learning* or *clustering*, which is considered in this thesis. Clustering requires a user specified definition of similarity, which will determine the groups or *clusters*. The objective of clustering is to partition the set of observations $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ into k disjoint subsets (clusters),

$$\mathcal{C} = \{\mathbf{C}_1, ..., \mathbf{C}_k\} \tag{1.1}$$

such that

$$\mathbf{C}_i \cap \mathbf{C}_j = \emptyset \; \forall \; i, j \in 1, ..., k \; i \neq j \tag{1.2}$$

$$\mathbf{C}_1 \cup ... \cup \mathbf{C}_k = \mathcal{X} \tag{1.3}$$

so as to maximise similarity between observations within the same cluster, while minimising similarity between observations in different clusters. There exist a variety of ways to define similarity, thus there is no universally adopted definition of what constitutes a cluster (Berkhin, 2006). Different specifications of similarity give rise to numerous approaches to clustering, some of which are discussed in Chapter 2.

The remainder of this chapter outlines the structure of the main body of this thesis and summarises the contributions made, while Chapter 2 provides an overview of the main clustering literature and current challenges that are considered in this work.

1.1 Thesis Aims and Structure

1.1.1 Aims

The aim of this thesis is to address the challenges of identifying clusters in datasets with large numbers of diverse features and complex clustering structures, while estimating their number. The approaches proposed are methodological ideas, which may be applied in a variety of application areas, but are not designed for any specific clustering task. These approaches do not attempt to completely solve all of the problems discussed in Section 2.2, associated with clustering complex datasets, but offer potential techniques to allow cluster identification in datasets where current methodology is limited. With the exception of Chapter 7, the methodology presented in this thesis relies on locating low-density cluster boundaries, which separate clusters corresponding to regions of high probability density, as defined in the density-based approach to clustering. The relevant definitions which underpin the algorithms proposed in this thesis are presented in Section 2.3.

1.1.2 Structure and Contributions

The body of this thesis consists of five chapters. Chapter 3 investigates the production of continuous representations of mixed datasets. This evaluates the performance of a variety of clustering algorithms over three different continuous representations of simulated and real-world mixed datasets. The production of a suitable continuous representation then permits the application of any clustering algorithm that makes the assumption of continuous observations, including the projective density-based clustering algorithms that are proposed in this thesis. To our knowledge, a comparative study into continuous representations of mixed data for clustering has not been undertaken in the literature.

In Chapter 4, we propose divisive hierarchical and a partitional clustering algorithms,

which are able to identify arbitrary numbers of high-density clusters in high-dimensional datasets by combining hyperplane separators that intersect regions of minimal probability density. These hyperplanes are located using the minimum density hyperplane (MDH), proposed by Pavlidis et al. (2016) for binary partitions, and are computed using one-dimensional projections of the data only, avoiding the problems associated with density estimation on high dimensions. The algorithms proposed extend the MDH approach to clustering by allowing the identification of multiple clusters, and estimating their number. Through an appropriate continuous representation of datasets with mixed attributes, we further extend the applicability of the proposed approaches to mixed datasets, upon which density-based clustering would ordinarily not be possible. The proposed algorithms extend the current literature by permitting the application of the density-based approach to clustering to large, high-dimensional and mixed datasets. Our algorithms locate very high-quality clustering results, often outperforming alternative well-established and sate-of-the-art clustering algorithms across a variety of datasets.

In Chapter 5, we further extend the applicability of the proposed projective density-based approach to clustering by removing the restriction of linear cluster separators, imposed by the MDH methodology. This is done by considering a non-linear mapping of the original observations into a feature space. This non-linear mapping results in a linear separator of the feature vectors permitting a non-linear separator of the original observations. It is not possible to directly compute the feature vectors, however, we present a formulation of the MDH in the feature space, which operates on the kernel matrix of pairwise inner products between the feature vectors. Applying the MDH approach to clustering is a new research area, and significantly extends the methodology in Chapter 4 by permitting the location of clusters of arbitrary shape whose convex hulls may overlap (provided an appropriate feature

mapping exists).

The dimensionality of the optimisation problem to locate the MDH in the feature space is determined by the number of observations, and for large datasets the location of the MDH in the feature space is computationally expensive. Therefore, we consider reducing the search space using an appropriate subspace of the feature space, in which an approximate minimum density separator of the feature vectors may be computed. We also include an equivalent approach to locate minimum density hyperplane separators of the feature vectors using their projections onto an orthonormal basis of the space spanned by them, that is more straightforward to implement than the formulation of the MDH using the kernel matrix. The bi-partitions of the feature vectors located by the MDH are combined in a divisive algorithm to allow the identification of multiple clusters that are not correctly identifiable by hyperplane separators in the data space, and automatically estimate their number.

Chapter 6 introduces an approach for the computationally efficient location of low-density cluster separators of datasets with large numbers of high-dimensional observations (or mapped feature vectors) through random projection (RP). This approach locates low-density separators using the one-dimensional projections of the data onto an appropriate random vector. This avoids the computational cost of locating a minimum density separator, which involves searching over a large number of dimensions for an optimal projection vector for cluster separation, and instead searches over a finite collection of random projection directions to approximate the optimal cluster boundary. We consider different optimality criteria to quantify the suitability of a set of univariate random projections for cluster separation, and investigate the quality of the partitions located when searching over varying numbers of random projections for a projection direction which permits a low-density cluster separator. The bi-partitions induced by the low-density cluster separators are combined

in a divisive algorithm, that automatically estimates the number of clusters. The use of RP to locate approximate minimum density cluster boundaries in one-dimensional subspaces is a novel idea. This approach permits the application of minimum density cluster separation in large, high-dimensional datasets, where the optimisation techniques proposed in Chapters 4 and 5 are practically infeasible.

Finally, Chapter 7 considers how univariate random projections may be applied to locate cluster separators in one-dimensional subspaces that are related to the objectives of alternative approaches to clustering, which do not rely on low-density separation, such as *k*-means and spectral clustering.

2

Literature Review

This chapter, provides a overview of common cluster definitions and associated algorithms. It is worth noting that some formulations of the clustering problem are NP-hard, making this a non-trivial problem. This is not an exhaustive review of the wide variety of the clustering literature, however, the relevant concepts for the remainder of the thesis are discussed. In addition, Section 2.2 presents the challenges in clustering, which this work aims to address. Finally, Section 2.3 presents the relevant definitions of the clusters and cluster separators which are located by the algorithms proposed in this thesis.

2.1 Clustering

In this section, existing approaches to clustering are discussed. These are categorised by the cluster definition assumed in each case. In addition, clustering algorithms can be divided into two approaches, *hierarchical* and *flat* (*partitional*). Hierarchical clustering locates a nested structure of partitions, defined as follows. Given two partitions $\mathcal{B} = \{\mathbf{B}_1, ..., \mathbf{B}_k\}$ and $\mathcal{C} = \{\mathbf{C}_1, ..., \mathbf{C}_m\}$ as defined in Eqs. (1.1)-(1.3), $\mathcal{B}$ is nested into $\mathcal{C}$ if every component of $\mathcal{B}$, $\mathbf{B}_i \subset \mathbf{C}_j$ for one of the components of $\mathcal{C}$. This hierarchy of nested partitions is summarised in a cluster tree or dendrogram, showing the clustering structure evident at different levels of similarity (Johnson, 1967). Meanwhile, partitional methods produce an overall clustering $\mathcal{C} = \{\mathbf{C}_1, ..., \mathbf{C}_k\}$ as defined in Eqs. (1.1)-(1.3) on a single level. Hierarchical clustering can be more computationally expensive than partitional clustering, however, the nested

structure generally allows superior detection of clusters on different scales. The majority of the algorithms outlined in this section are partitional, and are discussed separately to some well established methods for hierarchical clustering. It is worth noting that there exist hierarchical adaptations of a number of the partitional approaches discussed, but these are omitted for brevity and will be included, where relevant, in later chapters.

Despite the multitude of similarity measures, it is natural to assume that information about similarity exists in the spatial proximity between the observations. The evaluation of this spatial proximity is a non-trivial problem, and there exist multiple distance metrics that may be applied in practice. A complete discussion of the wide variety of proximity measures, which are appropriate for different data types and clustering applications is beyond the scope of this introduction, however, a comprehensive overview of these methods is provided by Gan et al. (2007).

For continuous data, the most common distance metrics are special cases of the Minkowski distance,

$$D_{ij} = \left(\sum_{l=1}^{d} |x_{i,l} - x_{j,l}|^r \right)^{1/r} \quad ; \; r \geqslant 1$$

where $x_{i,l}$ is the lth dimension of datum $\mathbf{x}_i \in \mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$. Taking $r = 1, 2, \infty$, gives the Manhattan, Euclidean and maximum distance metrics respectively. The most widely used of these metrics is the Euclidean distance, hence this is assumed throughout this section.

2.1.1 Hierarchical Clustering

For some applications of clustering, the nested structure located by hierarchical clustering is intuitive and occurs naturally. For example, in the biological application of clustering organisms into species and sub-species. The location of these cluster hierarchies may take two

distinct approaches. The divisive approach begins with all observations in a single cluster and sequentially divides this into smaller groups. By contrast, the agglomerative approach begins with all observations belonging to individual clusters, and merges the most similar groups at each level of the hierarchy.

Agglomerative

The agglomerative hierarchical approach requires the specification of distances between groups of observations, to quantify the most similar groups at each level of the hierarchy. The two most popular methods for this are single-link (nearest neighbour) (Sneath et al., 1973) and complete-link (furthest neighbour) (King, 1967) clustering, upon which the majority of agglomerative algorithms are based (Jain et al., 1999).

In single-link clustering, the distance between two clusters $\mathbf{C}_l$ and $\mathbf{C}_m$ is defined as the minimum pairwise distance between any $\mathbf{x}_i \in \mathbf{C}_l$ and $\mathbf{x}_j \in \mathbf{C}_m$,

$$d(\mathbf{C}_l, \mathbf{C}_m) = \min_{\mathbf{x}_i \in \mathbf{C}_l, \mathbf{x}_j \in \mathbf{C}_m} D_{ij}$$

where D_{ij} is the pairwise distance between observations $\mathbf{x}_i$ and $\mathbf{x}_j$. In complete-link clustering, the distance between two clusters $\mathbf{C}_l$, $\mathbf{C}_m$ is defined as the the maximum pairwise distance between any two observations $\mathbf{x}_i \in \mathbf{C}_l$, $\mathbf{x}_j \in \mathbf{C}_m$,

$$d(\mathbf{C}_l, \mathbf{C}_m) = \max_{\mathbf{x}_i \in \mathbf{C}_l, \mathbf{x}_j \in \mathbf{C}_m} D_{ij}.$$

At each level of the hierarchy, the two clusters which solve

$$\min_{\substack{l,m \in \{1,\ldots,k\} \\ l \neq m}} d(\mathbf{C}_l, \mathbf{C}_m)$$

are merged. Single-link clustering requires only a single short path between two clusters for them to be merged, resulting in a tendency to locate elongated (chain-like) clusters. By contrast, complete-link clustering requires all points within the two merged clusters to be connected by short paths. This gives rise to the location of compact clusters.

A complete agglomerative clustering can be computationally expensive, and the storage of the distances between all the clusters at each level of the hierarchy may be infeasible for large datasets. Algorithms such as balanced iterative reducing and clustering using hierarchies (BIRCH) (Zhang et al., 1996) aim to address the problem of high memory usage. BIRCH reduces the memory required to locate a complete hierarchy by storing only summary information of the clusters, not the original observations. Another problem associated with single link and complete link clustering is a sensitivity to outliers. To alleviate this problem, Guha et al. (1998) proposed the clustering using representatives (CURE) algorithm, which calculates similarity using representative points of a cluster, avoiding the issues associated with outliers. Similarly, robust clustering using links (ROCK) (Guha et al., 1999) defines similarity between individual observations (or clusters) based on the number of common neighbours (links) within a specified neighbourhood between them.

Divisive

In divisive clustering, it is necessary to define appropriate rules for the selection and subsequent splitting of clusters. There exist many algorithms which apply different selection and splitting rules, but the fundamental idea behind divisive clustering remains unchanged so we omit a complete discussion of these here. One of the most well established divisive algorithms is divisive analysis (DIANA) (Kaufman and Rousseeuw, 1990) which is based on the work of Macnaughton-Smith et al. (1964). At a given level with k clusters $\mathbf{C}_i$ for $i = 1, ..., k$,

the cluster with maximal diameter (distance between the two furthest points in the cluster),

$$\max_{l \in \{1,\ldots,k\}} \max_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{C}_l} D_{ij}$$

is split. To define the split of the cluster, the observation with maximal average dissimilarity to all other points in the cluster is selected as the seed. This seed initialises a new cluster, which is built up iteratively by selecting the closest point (which has not already been considered) to its centroid, and reassigning this point to the new cluster if it is closer to its centroid than the centroid of the observations which are still allocated to the old cluster. Alternative splitting criteria include using the two furthest points in the cluster as seeds and assigning observations to the closest of these (Hubert, 1973). This idea was extended to consider the partition created by all possible pairs of seeds in the cluster and retaining the result that optimises a pre-specified criterion (Roux, 1991, 1995).

Although a complete hierarchy can be useful, it is often necessary to extract a single, final clustering from the hierarchy, with a fixed number of clusters. Potential approaches for this are discussed in Section 2.2.3.

2.1.2 Partitional Clustering

The alternative approach of partitional clustering aims to locate all the clusters on a single level, producing a clustering in a single step. This tends to be less computationally expensive than hierarchical clustering, and may be more appropriate for application areas where the nested structure of a cluster hierarchy is not intuitive. For example, in some medical applications where the clustering task may be to identify patients who either have a disease or not, and therefore fall into two distinct categories.

Centroid-Based Clustering

Perhaps the most intuitive clustering objective is to minimise the sum of squared distances between the observations $\{\mathbf{x}_i\}_{i=1}^n$ and a representative point in their assigned cluster $\mathbf{C}_j$ such as the centroid $\{\mathbf{c}_j\}_{j=1}^k$. Stated formally, the objective of such methods are to solve the following optimisation problem

$$\min \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \mathbf{C}_j} \|\mathbf{x}_i - \mathbf{c}_j\|^2.$$

This is the objective of the most widely used clustering algorithm, k-means (Forgy, 1965; MacQueen, 1967; Lloyd, 1957). This algorithm begins by selecting k points as initial centroids. Then, each observation is assigned to its closest centroid. Based on these assignments, the centroids are updated and the procedure iterates until convergence.

Although widely used, k-means has a number of limitations which have been further studied in the literature. Firstly, k-means requires the pre-specification of the number of clusters, which is likely to be unknown in practice. The problem of estimating the number of clusters is discussed in Secion 2.2.3. Secondly, k-means is only guaranteed to converge locally, so can produce poor results when initialised badly. Initialisation may be done randomly, or alternative techniques for this are given in Forgy (1965); MacQueen (1967). Initialisations have also been proposed that aim to overcome the problem of convergence to the local optima (Krishna and Murty, 1999; Patané and Russo, 2001). More recently, Arthur and Vassilvitskii (2007) proposed the k-means++ algorithm that, through appropriate initialisation, is guaranteed to give an approximation ratio between the obtained and the globally optimal solutions of $\mathcal{O}(\log k)$ in expectation (over the randomness of the algorithm).

Additional limitations of k-means clustering include a sensitivity to outliers and noise

as well as problems when the centroids cannot be calculated, for instance in non-numerical data. An alternative centroid-based algorithm which overcomes the latter limitation is *k*-medoids, also known as partitioning around medoids (PAM) (Estivill-Castro and Yang, 2000; Kaufman and Rousseeuw, 1990). This approach uses actual observations with minimal dissimilarity to the other observations (medoids) to represent the clusters. Despite the above limitations, centroid-based algorithms are widely used in practice due to their straightforward implementation and intuitive interpretation, as well as relatively low computational cost.

Graph Theoretic Clustering

In graph theoretic clustering, each data point is seen as a node of an undirected graph with edge weights proportional to the similarity between the observations. Hence, subsets of the graph with maximal edge weights correspond to observations with the greatest similarity and may be interpreted as clusters. Single-link and complete-link clustering as described above may be viewed as locating maximally connected and maximally complete subgraphs respectively (Jain and Dubes, 1988). The best known divisive graph partitioning algorithm is Zahn's algorithm (Zahn, 1971), which constructs the minimum spanning tree (MST), then removes the edges of the MST with the largest lengths. In this case, the resulting clusters remain as connected subgraphs with maximal distance between them.

In partitional clustering, the problem is locating cuts of the graph which partition nodes with minimal edge weights between them. This is known as the minimum graph cut problem. Assume a graph $\mathcal{G} = (\mathcal{X}, \mathcal{E})$ whose vertices correspond to the observations $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ with undirected weighted edges $\mathcal{E}$ defined by the adjacency matrix $\mathbf{W} \in \mathbb{R}^{n \times n}$ whose elements W_{ij} are the similarities between pairs of vertices $\mathbf{x}_i$ and $\mathbf{x}_j$. There exist multiple approaches to define the edge weights in $\boldsymbol{W}$, but the three most common are,

1. The ϵ-neighbourhood graph, where $W_{ij} = 1$ if $\mathbf{x}_j$ is in the ϵ-neighbourhood of $\mathbf{x}_i$ and $W_{ij} = 0$ otherwise.
2. The k-nearest neighbour (KNN) graph, where $W_{ij} = 1$ if $\mathbf{x}_j$ is one of the k-nearest neighbours of $\mathbf{x}_i$ and $W_{ij} = 0$ otherwise. Symmetry can be imposed on this graph by connecting observations which are mutual nearest neighbours, or alternatively, observations which belong to one of each other's k-nearest neighbours.
3. The fully connected graph, constructed using any valid kernel function. The most widely applied kernel is the Gaussian kernel where $W_{ij} = \exp\left\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right\}$ where σ is a tuning parameter.

All of these adjacency matrices rely on the selection of appropriate values of tuning parameters. These choices critically affect the clusters produced, and the development of robust selection rules for these remains an open problem. Once an appropriate graph is constructed, the degree matrix, $\mathbf{D}^{\mathcal{G}}$ is defined as the diagonal matrix of the degrees, $d_i^{\mathcal{G}}$, of the vertices $\mathbf{x}_i$ in $\mathcal{G}$,

$$\mathbf{D}^{\mathcal{G}} = \text{diag}(d_i^{\mathcal{G}}) = \text{diag}\left(\sum_{j=1}^{n} W_{ij}\right) \ \forall i = 1, \ldots, n.$$

Further for a subset of vertices $\mathcal{S} \subset \mathcal{X}$ with complement $\mathcal{S}^c = \mathcal{X} \setminus \mathcal{S}$, define

$$\Omega(\mathcal{S}, \mathcal{S}^c) = \sum_{\mathbf{x}_i \in \mathcal{S}, \mathbf{x}_j \in \mathcal{S}^c} W_{ij}.$$

The minimum graph cut problem then seeks to cut $\mathcal{G}$ into subsets $\mathcal{S}_1, ..., \mathcal{S}_k$ so as to partition its vertices $\mathcal{X}$ while cutting the edges with smallest weight,

$$\text{mincut}(\mathcal{S}_1, ..., \mathcal{S}_k) = \frac{1}{2} \sum_{i=1}^{k} \Omega(\mathcal{S}_i, \mathcal{S}_i^c)$$

where the factor of $\frac{1}{2}$ avoids counting the edges cut twice. In the majority of cases, simply

minimising this problem results in the separation of a few outlying vertices, which is not desirable for clustering. It is therefore necessary to penalise cuts into small groups. The most common objectives for this are RatioCut (Hagen and Kahng, 1992) and Ncut (Shi and Malik, 2000),

$$\text{RatioCut}(\mathcal{S}_1, ..., \mathcal{S}_k) = \frac{1}{2}\sum_{i=1}^{k}\frac{\Omega(\mathcal{S}_i, \mathcal{S}_i^c)}{|\mathcal{S}_i|}$$

$$\text{Ncut}(\mathcal{S}_1, ..., \mathcal{S}_k) = \frac{1}{2}\sum_{i=1}^{k}\frac{\Omega(\mathcal{S}_i, \mathcal{S}_i^c)}{\text{vol}(\mathcal{S}_i)}$$

where $|\mathcal{S}_i|$ is the number of vertices in $\mathcal{S}_i$ and $\text{vol}(\mathcal{S}_i) = \sum_{\mathbf{x}_j \in \mathcal{S}_i} d_j^{\mathcal{G}}$. These penalties for the location of unbalanced clusters render the solution of RatioCut and Ncut NP-hard (Wagner and Wagner, 1993). However, a relaxed solution may be found, resulting in the well known spectral clustering algorithms (von Luxburg, 2007).

Define the matrix $\mathbf{H}$ to be the $n \times k$ matrix of cluster assignments, such that $H_{ij} = 1/\sqrt{|\mathcal{S}_j|}$ or $H_{ij} = 1/\sqrt{\text{vol}(\mathcal{S}_j)}$ if $\mathbf{x}_i \in \mathcal{S}_j$, and $H_{ij} = 0$ otherwise for RatioCut and Ncut respectively. Given this, it is possible to show (von Luxburg, 2007) that both RatioCut and Ncut are equivalent to trace minimisation problems of the form

$$\min \text{Tr}(\mathbf{H}^\top \mathbf{L}\mathbf{H}) \tag{2.1}$$

where $\mathbf{L}$ is a graph Laplacian of $\mathcal{G}(\mathcal{X}, \mathcal{E})$, defined below and $\mathbf{H}$ is defined as above. von Luxburg (2007) shows that relaxing $\mathbf{H}$ to be any real matrix allows the solution of a standard trace minimisation problem (Lutkepohl, 1997), solved by taking the first k eigenvectors of $\mathbf{L}$. For the solution of RatioCut, the unnormalised graph Laplacian,

$$\mathbf{L}_{un} = \mathbf{D}^{\mathcal{G}} - W$$

is required, while for Ncut, it is necessary to use the symmetric normalised graph Laplacian,

$$\mathbf{L}_{sym} = \left(\mathbf{D}^{\mathcal{G}}\right)^{-1/2} \mathbf{L}_{un} \left(\mathbf{D}^{\mathcal{G}}\right)^{-1/2}.$$

In the ideal case, where the components of the graph (clusters) are disconnected from each other, the graph Laplacian can be trivially ordered into a block diagonal matrix. Defining $\mathbf{V} \in \mathbb{R}^{n \times k}$ to be the matrix of the first k eigenvectors of the appropriate graph Laplacian as columns, therefore results in $\mathbf{V}$ having a single non-zero entry in each row. The position of this non-zero entry corresponds to the cluster to which the ith observation belongs. In the event that the components of the graph have some connectivity between them, $\mathbf{V}$ will have more than one non-zero entry per row, and the largest entry indicates the appropriate cluster assignment for each observation. Hence $\mathbf{V}$ is a relaxed version of the cluster assignment matrix $\mathbf{H}$. To locate a partition of the graph, it is necessary to transform $\mathbf{V}$ into a discrete indicator vector, which is typically done using k-means to cluster the rows of $\mathbf{V}$.

Model-Based Clustering

In model-based clustering, the set of observations $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ are assumed to be generated from a finite mixture model, whose k components are parametric probability distributions. This mixture distribution is denoted p and has the general form,

$$p(\mathbf{x}|\mathbf{\Theta}) = \sum_{i=1}^{k} \zeta_i \; p_i(\mathbf{x}|\boldsymbol{\theta}_i),$$

where $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_k)$ is a vector of mixing proportions such that $\sum_{i=1}^{k} \zeta_i = 1, \zeta_i > 0, \forall i = 1, ..., k$ and p_i is the probability density function for the ith mixture component with associated parameter vector $\boldsymbol{\theta}_i$. Each component of this mixture model is asso-

ciated with a single cluster. Given the possibility to estimate the parameter vector $\boldsymbol{\Theta} = (\zeta_1, ..., \zeta_k, \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_k)$, observations are assigned to the mixture component (cluster) with the highest probability of generating them. Thus, elements of the cluster assignment vector $\boldsymbol{\pi}$ are given by

$$\pi_i = \underset{j \in \{1,...,k\}}{\arg\max}\, \zeta_j \; p_j(\mathbf{x}_i | \boldsymbol{\theta}_j).$$

Although any probability distribution may be assumed for the mixture components, it is common in practice to assume a Gaussian mixture model (Zhuang et al., 1996; Everitt et al., 2011). In this case, estimation of the model parameters may be done by maximum likelihood estimation using the expectation-maximisation (EM) algorithm (Dempster et al., 1977). This uses augmented missing data in the form of the missing cluster labels, and iterates between taking the expected value of the cluster labels, given the current estimates of the parameters in the mixture model, and then maximising the likelihood for the parameters, given the estimated cluster labels. This process continues until convergence, returning the estimated parameters, $\boldsymbol{\Theta}$ and the cluster assignment vector $\boldsymbol{\pi}$.

In the case of spherical Gaussian components, this approach is equivalent to k-means clustering (Celeux and Govaert, 1992). Perhaps the most attractive feature of model-based clustering is the ability to estimate the number of clusters in a rigorous statistical framework, using well established model selection techniques such as the Bayesian information criterion (BIC) (Schwarz et al., 1978). This is discussed further in Section 2.2.3.

Non-Parametric Density-Based Clustering

In non-parametric statistical, known as density-based, clustering it is again assumed that an underlying probability distribution has given rise to the observations $\mathcal{X}$. However, unlike model-based clustering, this probability density p has an unknown form. Clusters are de-

fined as subsets of observations in contiguous regions of high density, concentrated around the domains of attraction of the modes of p (which are separated by low-density regions). Hartigan (1975) define these regions of high density based on the level sets of p,

$$L(c, p) = \{\mathbf{x} \in \mathbb{R}^d | p(\mathbf{x}) \geqslant c\}. \tag{2.2}$$

This approach can locate clusters of arbitrary shape and has a natural estimate for the number of clusters present. Practically, the true density p is unknown and must be estimated by a non-parametric density estimate $\hat{p}$. A consistent approach to approximate $L(c; p)$ is through a union of spheres around points whose estimated density, $\hat{p}$ exceeds c (Walther, 1997; Cuevas et al., 2000, 2001; Rinaldo and Wasserman, 2010). All of the modern density-based clustering algorithms (of which we are aware) locate the approximate level sets of p by seeking the modes of the estimated density $\hat{p}$ (Azzalini and Torelli, 2007; Stuetzle and Nugent, 2010; Chacón et al., 2015).

Locating the levels sets of p using an estimated density is closely related to the influential density-based spatial clustering of applications with noise (DBSCAN) algorithm (Ester et al., 1996), where points are considered to be in dense regions if the ϵ-neighbourhood around them contains sufficiently many points. If two points may be connected by a path that does not go through a point whose ϵ-neighbourhood is not sufficiently dense, then the two points are assigned to the same cluster. This may be equivalently thought of as locating the level sets of an estimated density which is constructed from uniform kernels with bandwidth ϵ.

In practice, selecting an appropriate level parameter c, to define the level sets is difficult. However, it is possible to vary c, producing a hierarchical structure of clusterings, which are summarised in a cluster tree, whose leaves correspond to the modes of the estimated

density $\hat{p}$. Another practical limitation of density-based clustering is that the construction of an estimated density becomes inaccurate in even moderate dimensions, fundamentally restricting the applicability of this approach to low-dimensional datasets.

Kernel-Based Clustering

A potential limitation of some approaches to clustering, is the inability to correctly identify clusters which are not linearly separable. This is relevant for the well-established centroid-based approaches, such as k-means as well as approaches which locate clusters using orthogonal one-dimensional projections of the data, which are discussed later in Section 2.2.1. Kernel-based learning (Muller et al., 2001) allows this restriction to be lifted, by mapping the original observations into a feature space, in which the clusters are linearly separable.

Given the set of observations $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$, let

$$\begin{aligned} \Phi : &\mathbb{R}^d \to \mathcal{F} \\ &\mathbf{x}_i \mapsto \phi(\mathbf{x}_i) \end{aligned}$$

be a non-linear feature mapping of $\mathcal{X}$ to a potentially much higher dimensional space $\mathcal{F}$. Any linear algorithm can then be applied in $\mathcal{F}$, corresponding to a non-linear separation of $\mathcal{X}$. Since $\mathcal{F}$ has the potential to be infinite-dimensional, it may not be possible to compute the mapped observations $\phi(\mathbf{x}_i)$, however, a kernel function,

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{F}},$$

may be used to compute scalar products in $\mathcal{F}$ without explicitly defining the feature vectors $\{\phi(\mathbf{x}_i)\}_{i=1}^n$. Therefore, using kernels, any (linear) algorithm which uses scalar products can

be implicitly computed in $\mathcal{F}$ (Schölkopf et al., 1998).

Clustering the feature vectors into k clusters $\mathbf{C}_1, \ldots, \mathbf{C}_k$ should optimise a cluster quality function, such as that defined by Shawe-Taylor and Cristianini (2004),

$$\sum_{j=1}^{k} \sum_{\mathbf{x}_l, \mathbf{x}_m \in \mathbf{C}_j} ||\phi(\mathbf{x}_l) - \phi(\mathbf{x}_m)||^2_{\mathcal{F}}. \tag{2.3}$$

Here the subscript $\mathcal{F}$ is used to denote the distance in the feature space. It is possible to show (Shawe-Taylor and Cristianini, 2004, Proposition 8.18) that Eq. (2.3) may be solved by identifying clusters which minimise distances between the observations and the centres of their assigned cluster. Therefore, in kernel k-means (Dhillon et al., 2004), the clustering of $\mathcal{X}$ is the solution to the following non-convex optimisation problem

$$\min \sum_{j=1}^{k} \sum_{\mathbf{x}_i \in \mathbf{C}_j} ||\phi(\mathbf{x}_i) - \mathbf{c}_j||^2_{\mathcal{F}} \tag{2.4}$$

where the $\mathbf{c}_j$'s are the cluster centroids in $\mathcal{F}$,

$$\mathbf{c}_j = \frac{\sum_{\mathbf{x}_l \in \mathbf{C}_j} \phi(\mathbf{x}_l)}{|\mathbf{C}_j|}.$$

These centroids cannot be computed explicitly, but we can evaluate the Euclidean distance from each $\phi(\mathbf{x}_i)$ to centroid $\mathbf{c}_j$ in $\mathcal{F}$ by,

$$\begin{aligned}
||\phi(\mathbf{x}_i) - \mathbf{c}_j||^2_{\mathcal{F}} &= \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_i) \rangle_{\mathcal{F}} - 2\langle \phi(\mathbf{x}_i), \mathbf{c}_j \rangle_{\mathcal{F}} + \langle \mathbf{c}_j, \mathbf{c}_j \rangle_{\mathcal{F}} \\
&= \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_i) \rangle_{\mathcal{F}} - 2 \sum_{\mathbf{x}_l \in C_j} \frac{\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_l) \rangle_{\mathcal{F}}}{|C_j|} + \sum_{\mathbf{x}_l, \mathbf{x}_m \in C_j} \frac{\langle \phi(\mathbf{x}_l), \phi(\mathbf{x}_m) \rangle_{\mathcal{F}}}{|C_j|^2} \\
&= \kappa(\mathbf{x}_i, \mathbf{x}_i) - 2 \sum_{\mathbf{x}_l \in C_j} \frac{\kappa(\mathbf{x}_i, \mathbf{x}_l)}{|C_j|} + \sum_{\mathbf{x}_l, \mathbf{x}_m \in C_j} \frac{\kappa(\mathbf{x}_l, \mathbf{x}_m)}{|C_j|^2}.
\end{aligned}$$

Therefore, the objective function in Eq (2.4) may be evaluated using the kernel function, avoiding the computation of the feature vectors. Despite its popularity, the non-convexity of this optimisation problem renders kernel k-means susceptible to convergence to local minima.

If the optimisation is relaxed, allowing a non-binary cluster assignment matrix, Shawe-Taylor and Cristianini (2004) show that the solution to the now convex optimisation problem of minimising Eq. (2.3) is given by the trace minimisation problem of spectral clustering, defined in Eq. (2.1). In this case, the adjacency matrix, $\mathbf{W}$ of the graph $\mathcal{G}(\mathcal{X}, \mathcal{E})$ is equivalent to a kernel matrix $\mathbf{K}$ whose elements are the pairwise scalar products of the mapped feature vectors $K_{ij} \in \mathbb{R}^{n \times n} = \kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{F}}$.

2.2 Open Problems in Clustering

This section outlines current challenges in clustering, which are considered in this thesis. Firstly, Section 2.2.1 introduces the challenges of clustering data which contain a large number of features for each observation (high-dimensional observations). Then, Section 2.2.2 discusses the limitations of current clustering methodology when observations have attributes which are discrete or categorical. Finally, Section 2.2.3 considers the problem of estimating the number of clusters.

2.2.1 High Dimensionality

Modern computing capabilities are allowing the generation and storage of increasingly large datasets. As a result, it is common that real-world datasets contain observations with many attributes (dimensions). This is a well-documented problem (Hinneburg and Keim, 1999; Agrawal et al., 1998; Kriegel et al., 2009), and is commonly referred to as "the curse of dimensionality". The problems associated with clustering this type of data go beyond the

computational complexity associated with analysing large datasets, impeding the fundamental assumptions required for cluster detection. Steinbach et al. (2004) present this problem intuitively by considering a fixed number of uniformly distributed points contained in grids of fixed size as dimensionality increases. The number of grids contained in the space grows exponentially with dimensionality, hence, unless the number of observations increases at the same rate, the proportion of cells which will be empty increases also. Thus, high-dimensional datasets are very sparse.

Further, it is likely that some features are strongly correlated with others, or do not contain relevant information for clustering. Along these irrelevant dimensions, the data appear uniform, and i.i.d, which is not appropriate for accurate cluster identification. Practically, in sparse high-dimensional datasets with large numbers of irrelevant dimensions, measures of spatial proximity and probability density, commonly used to define similarity between observations are not meaningful. This is due to the pairwise distances between observations that should belong to the same cluster not being significantly smaller than the pairwise distances between observations that should belong to different clusters, when computed over all dimensions. Further, clustering algorithms which rely on the specification or estimation of a probability density function cannot be applied, as the density is approximately zero everywhere. Therefore, discarding irrelevant features through dimensionality reduction is a necessity to make cluster detection possible. This may be done as a pre-processing step or locally, as part of the partitioning procedure. The latter approach is more common since it is often the case that different features are relevant for the detection of different clusters, making a global dimensionality reduction inappropriate.

Subspace clustering (Parsons et al., 2004) typically refers to methods which assume a subset (or subsets) of features are relevant for cluster detection. This restricts attention to axis-

parallel subspaces in which clusters are sought. Across the different approaches to subspace clustering, it is generally assumed that dimensions which allow the location of compact clusters should be retained. A k-medoid approach to this problem is adopted by the PROCLUS algorithm (Aggarwal et al., 1999). In this algorithm, the subspaces are built to have minimal standard deviation in the distances between the points and their closest medoid along each dimension. Distances are only calculated in the relevant subspace for each cluster. This approach tends to produce equally sized clusters with a spherical shape in their subspaces.

This underlying idea of building up subspaces in which clusters are identifiable may also be applied to alternative cluster definitions. Since non-parametric density based clustering is fundamentally limited to low-dimensional spaces, but has advantages such as being capable of locating clusters of diverse shapes, subspace clustering algorithms relying on this cluster definition are attractive. There exist a number of algorithms for this that locate subspaces in which the clusters are sufficiently dense. This definition of sufficient density to indicate an appropriate clustering, and the subsequent construction of the subspaces are the main differences between the algorithms that apply this approach.

PreDeCon (Böhm et al., 2004b) is a subspace variant of DBSCAN, which applies a modified distance measure, capturing the subspace of each cluster. This distance measure incorporates the subspace preference of each cluster at each point $\mathbf{x}_i$. A given dimension is considered relevant in the subspace of $\mathbf{x}_i$ if the variance of points in the Euclidean ϵ-neighbourhood of $\mathbf{x}_i$ is below a pre-determined threshold. The subspace modified distance measure is then a weighted Euclidean distance along the dimensions in the relevant subspace.

SubClu (Kailing et al., 2004) determines dense clusters in the same way as DBSCAN, by setting a lower threshold on the number of points in the ϵ-neighbourhood of each datum.

This definition of dense clusters is similar to that applied in PreDeCon, however, in SubClu, the relevant subspaces for each cluster are built iteratively. This process begins with all one-dimensional dense clusters. The dimensionality of the subspaces for each cluster are determined such that if a $\delta + 1$-dimensional subspace (where δ is an arbitrary dimension) contains a δ-dimensional subspace that is not dense, the $\delta + 1$-dimensional subspace cannot be considered dense, hence the dimensionality of the subspace is not increased further. Likewise, the CLIQUE algorithm (Agrawal et al., 1998) constructs dense subspaces for clusters using the same iterative procedure. However, this algorithm relies on an alternative definition of regions of high density, which uses an equally spaced axis parallel grid over the observations. Any grid unit containing at least τ points is considered dense. This grid-based approach reduces the computational cost compared to SubClu, but is often less accurate. All of these density-based approaches have attractive properties, such as the ability locate clusters of diverse shapes, and estimate their number. However, the input parameters are not intuitive to set.

In practice, axis-parallel subspaces may be too restrictive for some datasets. There exist a variety of algorithms that extend the concepts adopted by the aforementioned subspace algorithms, which do not adopt this constraint, and instead permit the detection of clusters in arbitrarily oriented subspaces. We refer to such approaches as projective clustering algorithms. This is a convention in this thesis but in the literature both projective and subspace clustering are used interchangeably.

The most common dimensionality reduction technique for projective clustering is principal component analysis (PCA), which projects the data, $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ such that maximal variability is retained, and reconstruction error is minimised (Tipping and Bishop, 1999).

This is done using the covariance matrix,

$$\mathbf{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top$$

where $\boldsymbol{\mu} \in \mathbb{R}^d$ is the mean vector. The eigen-decomposition of $\mathbf{\Sigma}$,

$$\mathbf{\Sigma} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top,$$

gives an orthonormal basis, $\mathbf{V}$ whose columns correspond to directions of decreasing variance in $\mathcal{X}$. Since $\mathbf{\Lambda}$ is a diagonal matrix, any correlation structure is removed in the projected data, $\mathbf{X} \cdot \mathbf{V}$ where $\mathbf{X}$ is the $n \times d$ data matrix. The majority of projective clustering algorithms rely on PCA, either on subsets of points or on the whole dataset. The ORCLUS algorithm (Aggarwal and Yu, 2000) is a k-medoid approach to projective clustering, and is an extension of PROCLUS to arbitrarily oriented subspaces. This clusters objects by minimising the distances between each data point and its closest medoid along the directions of low variability for each cluster.

Likewise, the density-based subspace approach may be extended to arbitrarily oriented subspaces by algorithms such as 4C (Böhm et al., 2004a), which extends the approach of PreDeCon. In this algorithm, the similarity between two points is determined by the similarity of the eigen-system of their ϵ-neighbourhoods. If two points are connected by a similar correlation of attributes, they are assumed to belong in each other's correlation neighbourhoods.

In this thesis, we focus on projective methods which rely on one-dimensional subspaces for clustering. The principal direction divisive partitioning algorithm (PDDP) (Boley, 1998) is a divisive algorithm, which recursively projects the data onto the first principal compo-

nent (direction of maximal variability), and then bi-partitions $\mathcal{X}$ at the mean of these projections. This continues until the maximum scatter value in each of the clusters does not exceed the scatter value of the centroids of all the clusters found so far. Two extensions of this algorithm are proposed by Tasoulis et al. (2010) to incorporate a more explicit cluster definition. Both algorithms project the data onto the first principal component as in PDDP. However, interval PDDP (iPDDP) splits at the point of maximal distance between two consecutive projections and density enhanced PDDP (dePDDP) constructs a kernel density estimate over the projections and splits at the global minimiser of this estimated density in the range between the two outer-most modes. Both of these algorithms rely on the *low-density cluster separation assumption*, and locate separating hyperplanes orthogonal to the first principal component which result in the largest margin and lowest density separations respectively. For datasets with compact, convex clusters, projecting onto the direction of maximal variability enables accurate clustering results, since along this direction, the clusters are likely to be well-separated (Boley, 1998). PDDP and its extensions have been shown to produce high-quality clustering results for applications such as gene expression clustering and text mining.

Although PCA projections can be useful for cluster detection in a number of areas, it is trivial to construct examples where directions of high variability are not suitable for cluster detection. *Projection pursuit* (PP) algorithms (Friedman and Tukey, 1974; Huber, 1985) encompass the search for low-dimensional spaces, that are appropriate for pattern recognition as a more general concept. PP methods aim to locate optimal linear projections of high-dimensional datasets, based on some measure of "interestingness" (known as the *projection index*) of a projection direction for the specified learning task (Jones and Sibson, 1987). This approach has been applied to locate low-dimensional subspaces for clustering (Friedman

and Tukey, 1974), regression (Friedman and Stuetzle, 1981b), classification (Friedman and Stuetzle, 1981a) and density estimation (Friedman et al., 1984). The definition of an interesting projection direction is not universally accepted, and therefore the majority of classical dimensionality reduction techniques may be thought of within the projection pursuit framework. For example, PCA is equivalent to PP, where the projection index is defined as the variance along the selected projection direction.

More recently, Pavlidis et al. (2016) proposed a PP algorithm called the minimum density hyperplane (MDH), which defines the projection index based on the minimum of the estimated density of the projections of the data along a univariate projection direction. This method aims to locate projection directions which are optimal for the separation of clusters, following the density-based approach to clustering, by locating minimum density boundaries between high-density regions associated with clusters. We discuss this in detail in later chapters.

2.2.2 Mixed Data

Although there are a variety of different definitions of a cluster, it is common to assume that dissimilarity between observations is related to a measure of spatial separation, usually Euclidean distance. However, in real-world applications, it is often the case that observations have attributes of diverse types (*mixed data*). In datasets with ordinal and nominal variables, discrete features can make standard continuous distance metrics, such as Euclidean distance inappropriate to define dissimilarity between observations.

This poses a significant challenge for the majority of approaches to clustering. In centroid-based clustering, non-numeric attributes make it impossible to compute the cluster centroids for the k-means algorithm, and even for discrete numeric data, the evaluation of spatial distances between observations and their assigned cluster centroid is not an interpretable

in the same way as for continuous data. Likewise, the notion of nearest neighbours or ϵ-neighbourhoods, used to construct the adjacency matrix of the graph in spectral clustering becomes invalid when considering spatial separation alone. Similarly, the definition of clusters as regions of high probability density requires an appropriate, continuous measure of spatial separation between the observations to construct the estimated density $\hat{p}$. Therefore, clustering non-continuous data using algorithms that rely on spatial proximity between observations is inappropriate.

One naive approach is to discard any non-continuous features, making the assumption that the clustering structure is evident in the continuous dimensions. However, this risks removing information which is necessary for cluster detection. Another naive approach is to treat all features as if they were continuous and proceed with a conventional clustering technique. This is also problematic, as any observations with the same combination of possible outcomes in the discrete dimensions will have low spatial separation, introducing an inherent grouping structure, which may not be truly indicative of the clusters present.

In the literature, there are two main approaches to incorporating mixed data for clustering. The first of these is to use an alternative distance metric to define pairwise dissimilarities between observations. The most well-known distance metric for mixed variables is the Gower distance (Gower, 1971) where the pairwise dissimilarity between observations $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as,

$$D_{ij} = \frac{\sum_{k=1}^{d} w_k d_{ij,k}}{\sum_{k=1}^{d} w_k} \tag{2.5}$$

where

$$d_{ij,k} = \frac{|x_{i,k} - x_{j,k}|}{\max(\mathbf{x}_{,k}) - \min(\mathbf{x}_{,k})} \tag{2.6}$$

for continuous and ordinal attributes and

$$d_{ij,k} = \begin{cases} 1 & , x_{i,k} \neq x_{j,k} \\ 0 & , x_{i,k} = x_{j,k} \end{cases} \tag{2.7}$$

for binary and categorical attributes. Also, $x_{i,k}$ is the kth dimension of the ith observation, $\mathbf{x}_{,k} = (x_{1,k}, \ldots, x_{n,k})$ and w_k is the user-defined weight for each variable in $\mathbf{x}$, which is typically set to $w_k = 1 \ \forall k$. Using this metric, it is possible to apply any clustering algorithm which relies only on pairwise distances between observations, such as the hierarchical clustering algorithms discussed in Section 2.1.1, PAM or spectral clustering.

A similar approach has also been proposed for k-means clustering with categorical variables in Huang (1997, 1998). In this paper, the distance between an observation $\mathbf{x}_i$ with continuous and discrete features $(\mathbf{x}_i^C, \mathbf{x}_i^D)$ and a cluster centroid $\mathbf{c}_j = (\mathbf{c}_j^C, \mathbf{c}_j^D)$ is given by,

$$D_{ij} = \sum_{k=1}^{d_C} (x_{i,k}^C - c_{j,k}^C)^2 + w_j \sum_{k=1}^{d_D} d_{ij,k}, \tag{2.8}$$

where w_j is the weight of the categorical data for cluster j, d_C and d_D are the number of continuous and discrete variables respectively and $d_{ij,k}$ is defined in Eq. (2.7), replacing $\mathbf{x}_j$ with $\mathbf{c}_j$. The algorithm then aims to minimise the sum of distances between the observations and their assigned centroid, as in the classical k-means algorithm.

This work was extended by Ahmad and Dey (2007) by weighting each of the distances for the continuous attributes, based on the pairwise separations of the observations in that attribute. This assumes that attributes showing high levels of separation are more relevant for clustering than those with low levels of separation. In addition, the distance between categorical attributes is not a binary outcome, instead the probability distribution of co-occurrence of values in each attribute is considered. Ahmad and Dey (2011) also adds a local

weight for each attribute in each cluster to the distance function. This can be thought of as a subspace algorithm as the distances are weighted differently along different dimensions for each cluster.

It is also possible to apply model-based clustering to mixed datasets by assuming an appropriate finite mixture model over the clusters. Everitt (1988) take this approach, assuming a parametric model for a set of realisations of a mixed variable $\mathbf{x}$ with d_C continuous and d_D discrete components, denoted $\mathbf{x}^C$ and $\mathbf{x}^D$ respectively. The parametric model is given by,

$$p(\mathbf{x}) = \sum_{i=1}^{k} \zeta_i \; MVN_{d_C+d_D}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$$

where $\boldsymbol{\zeta} = (\zeta_i, \ldots, \zeta_k)$ is the vector of mixing proportions such that $\sum_{i=1}^{k} \zeta_i = 1$ and $MVN_{d_C+d_D}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ denotes a $d_C + d_D$-dimensional multivariate normal distribution with mean $\boldsymbol{\mu}_i$ and covariance $\boldsymbol{\Sigma}_i$. However, the d_D-dimensional, multivariate normal random variables associated with the discrete attributes cannot be observed directly. Instead, the discrete observation vector is modelled as a threshold discretised form of a multivariate normal random variable. This discretisation requires multiple integrals of multivariate normal distributions which is computationally expensive. However, thereafter parameter estimation to fit the model and locate the clusters is a standard maximum likelihood estimation problem.

2.2.3 Estimating the Number of Clusters

In unsupervised learning, it is very unlikely that the true number of clusters that should be identified is known in advance. Therefore, it is necessary to estimate this as part of the learning process. This is an open problem in the literature , and different approaches to clustering offer different approaches to determining the number of clusters, such that the resulting

groups remain consistent with the specified cluster definition.

Hierarchical Clustering

A complete hierarchical clustering, which returns a nested clustering structure completely avoids this problem by providing a clustering result for all possible numbers of clusters from $1, \ldots, n$. However, this is computationally expensive, and often, the user must still extract a final clustering from the hierarchy, and determine an appropriate number of clusters for the problem of interest. It may be desirable to define an appropriate stopping rule, to automatically terminate the recursive splitting (or merging) of clusters in the hierarchy, such that the level of the hierarchy at which this stopping rule is satisfied allows determination of the number of clusters. For some applications, a stopping rule may be intuitive to specify, although this is not always the case, making this a non-trivial problem.

Given a complete hierarchy, with a single cluster at the root of the cluster tree (dendrogram), and n leaves for each of the individual observations, the most common approach to extract a final, flat clustering is to set a horizontal threshold across the dendrogram to locate the clusters which result from a single level of similarity (Jain and Dubes, 1988). However, it is well documented that this approach is unable to detect clusters on multiple scales (Stuetzle, 2003; Kriegel et al., 2011). Therefore, Campello et al. (2013) proposed the optimal extraction of clusters from hierarchies (OCE). This permits the extraction of clusters which correspond to non-horizontal cuts of the dendrogram, and locates the clustering that maximises the quality of the resulting clusters using a local measure of cluster quality. This allows the identification of clusters on multiple scales and with different densities.

Centroid-Based Clustering

For centroid-based clustering, it is intuitive to determine the number of clusters using the within cluster sum of squared distances (or within cluster variance), since this is the function which is minimised by this cluster definition, and therefore determines the quality of a clustering. The elbow heuristic considers the reduction in the within cluster variability for increasing numbers of clusters, and estimates the number of clusters such that any additional clusters do not significantly reduce the within cluster variability.

The Gap statistic (Tibshirani et al., 2001) formalises this heuristic within a formal statistical procedure. This compares the total within cluster variability for different numbers of clusters to the expected value under a null reference distribution with no obvious clustering structure (often the uniform distribution). For a given number of clusters, k, the Gap statistic is defined as,

$$\text{Gap}_n(k) = \mathbb{E}_n\{\log(W_k)\} - \log(W_k)$$

where W_k is the within cluster sum of squared distances when the data are partitioned into k clusters and $\mathbb{E}_n(\cdot)$ denotes the expectation under a sample of size n from the reference distribution, computed by Monte-Carlo simulation. Therefore, the Gap statistic measures the deviation of the observed within cluster sum of squared distances from its expected value under the null reference distribution. The standard error of the Monte-Carlo simulation with N null samples is defined as,

$$s_k = \sigma_k \sqrt{1 + 1/N}$$

where σ_k is the standard deviation of the log within sum of squared distances when the null samples are partitioned into k clusters. Finally, the number of clusters is chosen to be the minimum value of k for which the following holds,

$$\text{Gap}_n(k) \geqslant \text{Gap}_n(k+1) - s_{k+1}.$$

Therefore, k is chosen to be the smallest value for which the Gap statistic is within one standard deviation of the Gap statistic with $k+1$ clusters.

Spectral Clustering

In spectral clustering, the number of distinct connected components within the graph indicates the number of clusters present. It has been shown (Ng et al., 2002) that the largest eigenvalue of the graph Laplacian is equal to one, and that this eigenvalue will be a repeated with multiplicity equal to the number of groups in the graph. Therefore, it is possible to determine the number of clusters by counting the number of eigenvalues of the graph Laplacian which are equal to one. However, this property only holds if the clusters correspond to completely disconnected components within the graph. If the clusters are not disconnected, the largest eigenvalues are not all equal to one. In this case, it may be possible to determine the number of clusters using the heuristic proposed by Polito and Perona (2002). This heuristic searches for the point where the eigenvalues of the graph Laplacian decrease sharply. However, the location of this point may not be clear in datasets with high levels of noise.

Zelnik-Manor and Perona (2004) propose to use the eigenvectors of the graph Laplacian to estimate the number of clusters for spectral clustering. If the clusters are completely disconnected, the graph Laplacian may be sorted into a strictly block diagonal matrix, where

each block corresponds to the Laplacian of a sub-graph associated with a single cluster. In this case, the matrix of eigenvectors of the graph Laplacian, $\mathbf{V} \in \mathbb{R}^{n \times k}$ will have non-zero values only in entries corresponding to a single cluster. For a graph with k clusters, if we compute more than k eigenvectors, $\mathbf{V}$ will have some rows which contain more than one non-zero entry. Similarly, if we compute fewer than k eigenvectors, $\mathbf{V}$ will have some rows which contain no non-zero entries. Therefore, Zelnik-Manor and Perona (2004) propose to estimate the number of clusters to be the value which allows the minimal alignment cost between the eigenvectors of the graph Laplacian and the canonical co-ordinate system $\mathbf{e}_1, \ldots, \mathbf{e}_k$.

Model-Based Clustering

The model-based approach to clustering allows the estimation of the number of clusters through standard statistical model selection techniques, provided it is possible to construct a likelihood for the chosen clustering model. The value of the likelihood for models with different numbers of mixture components (clusters) may be used to detect when a more complex model does not fit the data significantly better than a model with fewer parameters. The most common model selection techniques for this task are the Akaike information criterion (AIC) and Bayesian information criterion (BIC). For a model with p fitted parameters, with likelihood L, the AIC is defined as,

$$\text{AIC} = 2p - 2\log(L),$$

while the BIC is,

$$\text{BIC} = \log(n)p - 2\log(L)$$

where n is the number of observations. The number of clusters is determined at the point where the information criterion is non-decreasing.

Density-Based Clustering

For density-based clustering, the level set definition given in Eq. (2.2) inherently estimates the number of clusters to be the number of regions of density greater than the level parameter, c, which are separated by regions of density lower than c. Irrespective of the choice of density estimate applied by different density-based algorithms, the number of clusters equates to the number of high-density regions, concentrated around the modes of the estimated density of the data. In practice, the specification of a threshold at which the density is considered sufficiently high to constitute a cluster is non-trivial. However, varying the level parameter does allow the computation of a complete cluster hierarchy to avoid this problem.

2.3 Definitions

In this section, we define the high-density clusters and low-density separators, that we aim to locate throughout the main body of this thesis. We define high-density clusters based on the estimated density over $\mathcal{X}$, $\hat{p}_{\mathbf{x}}$ by adapting the definition in Hartigan (1975) as follows:

Definition 1. [High-density clusters] (Hartigan, 1975) Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ where $\mathbf{x}_i \in \mathbb{R}^d$ be a set of realisations of a random variable X with estimated probability density function $\hat{p}_{\mathbf{x}}$. High-density clusters are defined as maximally connected subsets of the level sets of $\hat{p}_{\mathbf{x}}$,

$$L(c; \hat{p}_{\mathbf{x}}) = \left\{ \mathbf{x} \in \mathbb{R}^d | \hat{p}_{\mathbf{x}}(\mathbf{x}) \geqslant c \right\} \; ; \; c \geqslant 0.$$

When $\hat{p}_{\mathbf{x}}$ is unimodal, $L(c; \hat{p}_{\mathbf{x}})$ is connected for all values of c, and hence no cluster

structure exists. If $\hat{p}_{\mathbf{x}}$ is multi-modal, $L(c; \hat{p}_{\mathbf{x}})$ may be connected or not depending on the value of c. If it is disconnected, it is formed by two or more connected components, which correspond to regions surrounding the modes of $\hat{p}_{\mathbf{x}}$ (Menardi and Azzalini, 2014). A direct consequence of defining clusters as observations that lie in contiguous regions of high density in $\hat{p}_{\mathbf{x}}$ is that cluster boundaries pass through regions of low density. Therefore, we define a low-density separator according to Definition 2.

Definition 2. [Low-density separator] For a connected set $\mathbf{S} \subset \mathbb{R}^d$, the surface of $\mathbf{S}$, $\partial\mathbf{S}$, is a low-density separator if $\exists c \geqslant 0$ for which the following hold:

1. there exist distinct components $\mathbf{C}_1, \mathbf{C}_2$ of $L(c; \hat{p}_{\mathbf{x}})$ s.t. $\mathbf{C}_1 \subset \mathbf{S}, \mathbf{C}_2 \cap \mathbf{S} = \emptyset$;
2. $\max_{\mathbf{x} \in \partial\mathbf{S}} \hat{p}_{\mathbf{x}}(\mathbf{x}) \leqslant c$.

If $\mathcal{X}$ contains a family of high-density clusters, then a collection of low-density separators can identify all of these clusters. However, the evaluation of the density along a cluster separator is computationally intractable for separators of arbitrary shape. Therefore, we must restrict attention to linear separators (hyperplanes) that partition dense, linearly separable sets as defined in Definition 3,

Definition 3. [Dense linearly separable sets] Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ be a set of realisations of a random variable X with estimated probability density function $\hat{p}_{\mathbf{x}}$. A family $\mathbf{C}_1, \ldots, \mathbf{C}_k$ of mutually disjoint subsets of $\mathcal{X}$ is *dense and linearly separable* if there exists $c > 0$, such that for any $\mathbf{x}_i, \mathbf{x}_j \in \mathbf{C}_m$, $m \in \{1, \ldots, k\}$,

$$\min_{t \in [0,1]} \hat{p}_{\mathbf{x}} \left(t\mathbf{x}_i + (1-t)\mathbf{x}_j \right) > c. \tag{2.9}$$

Moreover, there exists I such that $\emptyset \neq I \subsetneqq \{1, \ldots, k\}$, such that,

$$\text{conv}\left(\cup_{i \in I} \mathbf{C}_i\right) \cap \text{conv}\left(\cup_{j \in I^C} \mathbf{C}_j\right) = \emptyset, \tag{2.10}$$

and for any $\mathbf{x}_i \in \text{conv}\left(\cup_{i \in I} \mathbf{C}_i\right)$ and $\mathbf{x}_j \in \text{conv}\left(\cup_{j \in I^C} \mathbf{C}_j\right)$,

$$\max_{t \in [0,1]} \hat{p}_{\mathbf{x}}\left(t\mathbf{x}_i + (1-t)\mathbf{x}_j\right) < c, \tag{2.11}$$

where $I^C = \{1, \ldots, k\} \setminus I$ is the complement of I, and $\text{conv}\,(\cdot)$ denotes the convex hull.

As a consequence of applying Definition 3, the family of clusters in $\mathcal{X}$ is linearly separable if there exists a hyperplane along which the maximum value of $\hat{p}_{\mathbf{x}}$ is at most c, and which also separates at least one cluster from the rest of the data. This definition results in the sets $\mathbf{C}_1, \ldots, \mathbf{C}_k$ corresponding to dense clusters, as defined in Definition 1 with the additional constraint of convexity. We further define the set $\mathcal{X}$ to be *dense and linearly clusterable* (with respect to the density estimator $\hat{p}_{\mathbf{x}}$) if it contains a family of convex dense clusters, $\mathbf{C}_1, \ldots, \mathbf{C}_k$ such that any (non-trivial) subset of this family is linearly separable.

Definition 4. [Dense linearly clusterable sets] Let $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ be a set of realisations of the random variable X with estimated probability density function $\hat{p}_{\mathbf{x}}$. A family $\mathbf{C}_1, \ldots, \mathbf{C}_k$ of mutually disjoint subsets of $\mathcal{X}$ is *dense and linearly clusterable* if for any subset $I \subsetneqq \{1, \ldots, k\}$ satisfying $|I| > 1$, the family $\{\mathbf{C}_i\}_{i \in I}$ is dense and linearly separable.

3

Continuous Representations of Mixed Data

Abstract

We consider the problem of locating clusters in datasets with diverse (mixed) attributes. A number of approaches to clustering, including density-based algorithms require a set of continuous observations to correctly identify the clustering structure present. Therefore, we consider the production of a continuous representation of mixed datasets, upon which clustering may be performed. We apply three continuous representations across simulated and real-world datasets with varying characteristics, and evaluate the clustering performance of projective density-based and other well-established clustering algorithms over these representations. We find that locating an appropriate continuous representation can be challenging but in general, the most consistently high-quality results were located using the continuous representation from constant shift embedding (Roth et al., 2003).

3.1 Introduction

Although there is no single, universally adopted definition of a cluster, the vast majority of approaches to clustering rely on spatial separation in some way to define the clusters present. Consequently, many clustering algorithms rely on the set of observations $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{n}$ having continuous attributes. However, many datasets contain observations with diverse types of features (*mixed data*). In this case, defining similarity solely on spatial separation induces maximal similarity between observations with the same set of outcomes in discrete dimensions. This is not meaningful for the detection of the true clustering structure, since each possible combination of outcomes in the discrete dimensions of $\mathcal{X}$ appears as an individual cluster. There exist general distance metrics, for example the Gower distance metric (Gower, 1971), which may be used in place of metrics such as Euclidean distance. These allow the construction of a more meaningful dissimilarity matrix for mixed data. Thus, the application of clustering algorithms that can operate on pairwise dissimilarities alone is still possible. However, this is not sufficient for the density-based approach to clustering, which requires a set of continuous observations in order to define a continuous estimated probability density function, in which subsets of observations in contiguous regions of high probability density are associated with clusters. Another, related, challenge associated with density-based clustering is that density estimation becomes unreliable in even moderate dimensions by modern standards (Rinaldo and Wasserman, 2010). This means that for the practical application of density-based clustering techniques, dimensionality reduction becomes a necessity. However, it is not clear how to specify appropriate projections for clustering in the case of non-continuous observations.

These two limitations mean that to apply density-based clustering to mixed datasets, it is necessary to transform the original observations to obtain a continuous representation,

upon which clustering can be performed. In this chapter, we investigate different continuous representations of mixed datasets, and their appropriateness for cluster detection. We quantify the quality of the continuous representations by the clustering performance of projective density-based algorithms (which are the focus of this thesis) and alternative algorithms, which also require a set of continuous observations.

The only work we are aware of that discusses the problem of finding an appropriate continuous representation of mixed data for density-based clustering is Azzalini and Menardi (2016). This employs multi-dimensional scaling (MDS) (Borg and Groenen, 2005) and then locates clusters by constructing an estimated density over all dimensions of the transformed data. For this reason, this work is limited to using a small number of dimensions in the continuous representation. Since we focus on projective density-based methods, which remain applicable for high-dimensional applications, we remove this restriction and allow the continuous representations used to have higher dimensionality. We further extend this work by also investigating the continuous representations produced by mixed probabilistic principal component analysis (mPPCA) (Khan et al., 2010), and constant shift embedding (CSE) (Roth et al., 2003).

Like standard probabilistic principal components analysis (PPCA) (Tipping and Bishop, 1999), mPPCA assumes that observations originate from a Gaussian latent variable model. Each categorical variable is assumed to be sampled from a multinomial distribution, with probabilities given by a multinomial logistic regression function applied on the latent variable. Both CSE, and MDS, make no assumptions about the data generating process, and rely exclusively on pairwise dissimilarities, defined by a metric which is appropriate for non-continuous data. In all our work, we use the Gower distance metric. MDS aims to produce a continuous representation which retains the pairwise distances. Meanwhile, CSE seeks a

continuous representation, upon which k-means clustering is guaranteed to assign all observations to the same clusters as pairwise clustering on the dissimilarity matrix.

The remainder of this chapter is organised as follows: Sections 3.2, 3.3 and 3.4 present the processes of locating continuous representations by MDS, mPPCA and CSE respectively. Section 3.5 discusses our choice of dimensionality for each of the continuous representations. Section 3.6 presents experimental results for the clustering performance of projective density-based and well-established clustering algorithms across the continuous representations of simulated and real-world benchmark datasets. Finally, the work is concluded in Section 3.7.

3.2 Multi-Dimensional Scaling

MDS is a well established method for dimensionality reduction, which seeks a low-dimensional continuous representation of the data that will minimise a measure of distortion of pairwise distances. In metric MDS (Borg and Groenen, 2005), the associated cost function to be minimised is

$$J_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(D_{ij}^2 - d_{ij}^2 \right)^2$$

where D_{ij} denotes the original distance between observations i and j and $d_{ij} = \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j\|_2$ is the distance between the continuous representations of the two observations, $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively. The solution to this minimisation is given by $\mathbf{\Lambda}^{1/2}\mathbf{V}^\top$ where $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues of $\mathbf{D} = [D_{ij}]$ and $\mathbf{V}$ is the matrix containing the eigenvectors of $\mathbf{D}$.

Non-metric MDS (Borg and Groenen, 2005) allows a non-linear, monotonic transformation of the pairwise distances. In this case, the cost function is given by the SSTRESS criterion,

$$J_2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \left(f(D_{ij})^2 - d_{ij}^2 \right)^2,$$

where $f(\cdot)$ is the monotonic transformation of the input pairwise distances, which is optimised as part of an iterative procedure. In all our experiments, we use non-metric MDS to produce our continuous representation. Non-metric MDS does not offer a criterion to select the appropriate dimensionality for clustering. The only information provided is the value of the objective function for different dimensions, but this information is not related to clustering. The selection if this dimensionality is discussed in Section 3.5.

3.3 Mixed Probabilistic Principal Components Analysis

Let the d-dimensional mixed observation vector $\mathbf{x}_i = (\mathbf{x}_i^C, \mathbf{x}_i^D)$, have continuous and discrete dimensions $\mathbf{x}_i^C \in \mathbb{R}^{d_C}$ and $\mathbf{x}_i^D \in \mathbb{N}^{d_D}$ respectively. PPCA (Tipping and Bishop, 1999) reformulates standard PCA within a latent variable model. A latent representation of the observation vector is provided by the distribution of the latent variables $\mathbf{z}_i \in \mathbb{R}^{d_L}$ conditional on the observations, $p(\mathbf{z}_i|\mathbf{x}_i)$. For the continuous dimensions of $\mathbf{x}_i$, a Gaussian latent variable model is assumed,

$$\mathbf{z}_i \sim \mathcal{N}(0, \sigma_z^2 I)$$
$$\mathbf{x}_i^C|\mathbf{z}_i \sim \mathcal{N}(\mathbf{W}^C\mathbf{z}_i + \boldsymbol{\mu}^C, \sigma_{\mathbf{x}^C}^2\mathbf{I}),$$

where $\mathbf{W}^C \in \mathbb{R}^{d_C \times d_L}$ is the factor loading matrix, $\boldsymbol{\mu}^C$ is the offset parameter and $\sigma_{\mathbf{x}^C}^2\mathbf{I}$ is the covariance matrix. In the limiting case when $\sigma_{\mathbf{x}^C}^2$ tends to zero, the columns of $\mathbf{W}^C$ are the singular vectors of $\mathbf{X}^C = [\mathbf{x}_1^C, \ldots, \mathbf{x}_n^C]$, thus standard PCA is recovered. This conjugate model results in the maximum likelihood estimates of $\mathbf{W}^C$ and $\sigma_{\mathbf{x}^C}^2$ having an analytical solution. The underlying probabilistic model enables PPCA to handle missing values, and extensions including mixtures of Gaussian latent variables. However, for these problems an analytical solution is not admissible, and instead parameter estimation is performed through

the expectation maximisation (EM) algorithm.

A natural extension of PPCA to accommodate categorical variables is through the specification of a link function, which maps continuous variables in the latent space to categorical variables in the observation space. In particular, for each of the d_D discrete variables, mPPCA uses a one-of-M_j encoding, where M_j is the number of possible values of the j-th discrete variable. Let x_{ij}^D denote the j-th discrete variable of the i-th observation. In mPPCA, x_{ij}^D is assumed to follow a multinomial distribution, with probabilities conditional on $\mathbf{z}_i$,

$$\begin{aligned}
\boldsymbol{\eta}_{ij} &= \mathbf{W}_j^D \mathbf{z}_i + \boldsymbol{\mu}_j^D, \\
S(\boldsymbol{\eta}_{ij}) &= \left(\frac{\exp(\eta_{ij,1})}{\sum_{m=1}^{M_j} \exp(\eta_{ij,m})}, \ldots, \frac{\exp(\eta_{ij,M_j})}{\sum_{m=1}^{M_j} \exp(\eta_{ij,m})} \right), \\
x_{ij}^D | \mathbf{z}_i &\sim M(S(\boldsymbol{\eta}_{ij})),
\end{aligned}$$

where $\mathbf{W}_j^D \in \mathbb{R}^{M_j \times d_L}$ and $\boldsymbol{\mu}_j^D \in \mathbb{R}^{M_j}$, are the factor loading matrix and the offset for the j-th discrete variable respectively and $S(\cdot)$ is the multinomial logistic regression (also known as softmax) link function.

The model for incorporating the discrete variables prevents a closed form solution for $p(\mathbf{z}_i|\mathbf{x}_i)$ meaning a standard EM algorithm is not applicable for the estimation of the model parameters. Instead, a variational EM algorithm is proposed in Khan et al. (2010). This algorithm is computationally expensive for large datasets and, as with any EM algorithm, is not guaranteed to converge to the global maximum of the likelihood. Hence, different representations can result depending on initialisation. A further challenge for clustering after applying mPPCA is that the maximum dimensionality of the continuous representation increases linearly with the number of possible values of each categorical variable, $\max d_L = d_C + \sum_{j=1}^{d_D} M_j$. This renders the problem of high-dimensionality increasingly

problematic.

3.4 Constant Shift Embedding

CSE (Roth et al., 2003) aims to embed the data into a continuous vector space so that the relative quality of a clustering in the continuous space is the same as in the original space, in which pairwise distances were computed. CSE measures the quality of a clustering in terms of the pairwise clustering cost function (Puzicha et al., 1999),

$$H^{\text{pc}}(\mathbf{M};\mathbf{D}) = \frac{1}{2}\sum_{m=1}^{k}\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}M_{im}M_{jm}D_{ij}}{\sum_{l=1}^{n}M_{lm}}, \tag{3.1}$$

where $\mathbf{D}$ is the matrix of pairwise dissimilarities with $\text{diag}(\mathbf{D}) = 0$, and $\mathbf{M} \in \{0,1\}^{n\times k}$ is the cluster assignment matrix, with $M_{jm} = 1$ if observation j is assigned to cluster m and $\sum_{m=1}^{k} M_{jm} = 1$. The minimisation of H^{pc} is equivalent to minimising the k-means clustering criterion, if $D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$ for each $\mathbf{x}_i, \mathbf{x}_j$.

The central idea behind CSE is that if $\mathbf{D}$ can be decomposed in the form, $D_{ij} = S_{ii} + S_{jj} - 2S_{ij}$, where $\mathbf{S}$ is a positive semidefinite matrix, then $\mathbf{S}$ can be viewed as the matrix of inner products, $\mathbf{S} = \mathbf{X}\mathbf{X}^\top$ in some space $\mathbf{X}$, and therefore $D_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2^2$. Thus, $\mathbf{X}$ is the natural candidate for the representation (embedding) of the data into a continuous vector space. For any $\mathbf{D}$, there is a class of matrices $\mathbf{S}$ that satisfy the above property so Roth et al. (2003) restrict attention to the *centralised matrix*, $\mathbf{S}^c$ which is uniquely defined for each $\mathbf{D}$,

$$\mathbf{S}^c = -\frac{1}{2}\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right)\mathbf{D}\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top\right),$$

where $\mathbf{1}$ is the n-dimensional vector of ones. Let us define $\tilde{\mathbf{S}}^c = \mathbf{S}^c - \lambda_n\mathbf{I}$, where λ_n is equal to the smallest eigenvalue of $\mathbf{S}^c$. Then $\tilde{\mathbf{S}}^c$ is by construction positive semidefinite, and

can be shown to be the centralised matrix associated with $\tilde{\mathbf{D}} = \mathbf{D} - 2\lambda_n(\frac{1}{n}\mathbf{1}\mathbf{1}^\top - \mathbf{I})$. Since $H^{\text{pc}}(\mathbf{M}; \tilde{\mathbf{D}}) = H^{\text{pc}}(\mathbf{M}; \mathbf{D}) - \lambda_n(n-k)$ the relative quality of different clusterings, $\mathbf{M}$, is unaffected by the addition of the constant $-2\lambda_n$ to the non-diagonal elements of $\mathbf{D}$. We can therefore equivalently consider $\tilde{\mathbf{S}}^c$ as the inner product matrix, and obtain the continuous representation $\mathbf{X} = \mathbf{V}\mathbf{\Lambda}^{1/2}$ through the eigen-decomposition $\tilde{\mathbf{S}}^c = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$.

3.5 Dimensionality of Continuous Representation

None of the aforementioned methods provide a definitive criterion to select the dimensionality of the continuous representation. To ensure that no information is lost mPPCA requires $\min\{n, d_C + \sum_{j=1}^{d_D} M_j\}$ dimensions, where d_C is the number of continuous dimensions in the data, d_D is the number of categorical variables, and M_j is the number of discrete values the j-th categorical variable can take. For MDS a lossless continuous embedding requires $\min\{n, d_C + d_D\}$ dimensions, while for CSE the dimensionality would have to equal the number of observations, n.

In practice, using the maximum possible number of dimensions for the continuous representation is not necessary to obtain the best possible clustering result. For CSE it is possible to use the eigenvalues of $\tilde{\mathbf{S}}^c$ to select the number of dimensions by setting a threshold on the total variance captured, thus excluding dimensions that contribute very little to the overall variance. Although there is no guarantee that directions of maximum variance are appropriate for clustering (Kriegel et al., 2011), it is arguably unlikely that directions along which the data exhibits almost no variability are relevant for cluster detection. This choice will always be smaller than the number of dimensions required for a lossless representation. We set this threshold such that 90% of the variability is retained. In our experiments this significantly reduced the number of dimensions while having negligible affect on the clustering

results compared to the lossless representations.

In the case of MDS, Azzalini and Menardi (2016) recommend using no more than five dimensions, since for the datasets used in their work, additional dimensions do not significantly improve the reconstruction error and the quality of the clustering result. However, the datasets used by Azzalini and Menardi (2016) contain very few clusters which is not necessarily the case for the datasets in our study. We instead use the eigenvalues of $\mathbf{D}$ resulting from metric MDS to select the dimensionality, and using the same procedure as for CSE, exclude dimensions which do not contribute significantly to the overall variance. We then use the representation from metric MDS to initialise non-metric MDS to produce the final embedding.

For mPPCA, the dimensionality of the continuous representation (latent variable) must be specified as a parameter of the model. Without prior information on the appropriate number of dimensions for clustering, we used the maximum number to produce a lossless representation. This resulted in the representations from mPPCA having a much higher dimensionality than the representations from CSE and MDS in some examples. None of these selection criteria led to a choice of dimensionality greater than the number of observations for the datasets considered in this work. For some datasets, this is a possibility, in which case, it would be more appropriate to select the dimensionality equal to the number of observations.

3.6 Experimental Results

In this section, the quality of the continuous representations produced by MDS, mPPCA and CSE are investigated through a comparison of the clustering results produced from each representation of simulated and real-world benchmarks datasets. For this comparison, we

used projective density-based clustering algorithms and well-established algorithms that cannot be applied using the pairwise dissimilarities alone:

1. Hierarchical Minimum Density Hyperplane (MDH_{hier}). This is the algorithm proposed in Chapter 4 where data are recursively bi-partitioned by the hyperplane that intersects a region of minimal density. Splitting terminates when the minimum density hyperplane (MDH) for each cluster is not appropriate to separate modes of the estimated density associated with clusters.

2. Ensemble Minimum Density Hyperplane (MDH_{ens}). This is the partitional algorithm, also proposed in Chapter 4 where multiple bi-partitions from locally optimal minimum density hyperplanes are combined by the model-based ensemble clustering approach of Topchy et al. (2005). The number of clusters is estimated using BIC.

3. Density-enhanced principal direction divisive partitioning (dePDDP) (Tasoulis et al., 2010). This is a divisive algorithm in which data are recursively projected onto the first principal component and then partitioned at the global minimiser, in the range between the outer-most modes, in the estimated density of the projections. This is related to MDH_{hier} since the resulting separating hyperplane passes through a region of minimal density with the constraint of its normal vector being equal to the first principal component. This algorithm terminates when the estimated projected density is unimodal for all clusters.

4. k-means++ (Arthur and Vassilvitskii, 2007) which is a recent variant of the classical k-means algorithm that, under appropriate initialisation, results in a clustering guaranteed be $\mathcal{O}(\log k)$ competitive with the true k-means clustering. We used the Gap statistic (Tibshirani et al., 2001) to estimate the number of clusters.

5. Gaussian mixture model-based (GMM) clustering using BIC to estimate the number of clusters (Fraley and Raftery, 2002). This is related to density-based clustering in the sense that the clusters obtained are individual unimodal components of a Gaussian mixture density.

We also considered density-based algorithms that seek to locate dense regions by estimating the density over all the dimensions of the continuous representations, such as pdfCluster (Menardi and Azzalini, 2014) and DBSCAN (Ester et al., 1996). However, the dimensionality of the continuous representations made these algorithms unreliable so the results are omitted. For all algorithms, the parameter settings were the same as in Section 4.4.1. Clustering performance is evaluated by normalised mutual information (NMI) (Strehl and

Ghosh, 2002), which takes values in the range $[0, 1]$, with higher values indicating better performance. Other performance measures were considered but these did not alter the relative quality of the partitions on the different continuous representations.

3.6.1 Simulation Study

Here we evaluate the clustering results produced by the different continuous representations of simulated mixed data with varying numbers of dimensions and numbers of clusters. These simulations allow us to control the level of difficulty of the clustering problem. We consider two generative models to assess the continuous representations produced under different modelling assumptions. This is particularly relevant for mPPCA, where a specific generative model is assumed.

For the first generative model, we adopt the same model as mPPCA where the clustering structure is induced by generating the latent variables $\{\mathbf{z}_i\}_{i=1}^n \subset \mathbb{R}^2$ from a Gaussian mixture model whose k components represent clusters. The means in each dimension were drawn uniformly from the range $[-5, 5]$ and covariance matrices were generated to have eigenvalues in the range $[10^{-3}, 10^{-2}]$. This produced a very clear clustering structure in the latent variables, so noise was introduced in the generation of the mixed observation vectors. This was done by filling the factor loading matrices $\mathbf{W}^C$ and $\mathbf{W}_j^D$ for $j = 1, ..., d_D$ with Uniform$(0, 1)$ random variables. All offset terms were set to zero, and to avoid increasing dimensionality substantially in the continuous representations, only two possible outcomes were permitted for all discrete dimensions. We denote this generation process MixGen[1].

In the second generative model, the distribution of the observations is a mixture model in which each of the k components constitutes a cluster. For each dataset, the mixing proportions were generated as $\zeta_i = u_i / \sum_{j=1}^k u_j$, where $u_i \sim$ Uniform$[1, 2]$, and the parameters

for each of the components were generated randomly as follows,

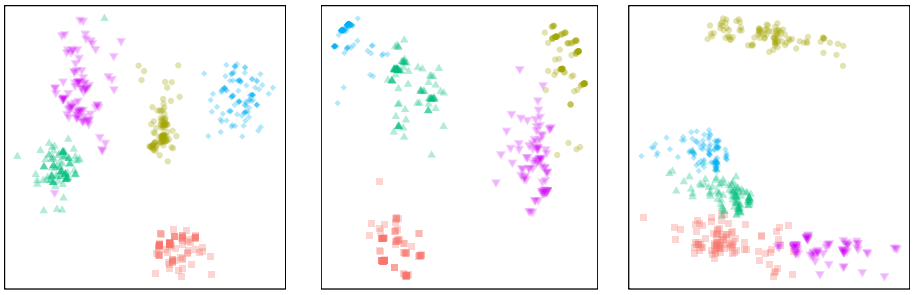
$$\begin{aligned}
&\boldsymbol{\mu}^C \sim \text{Uniform}(0, k/3)^{d_C};\\
&\mu_j^D \sim \text{Bern}(0.5),\ j = 1, \ldots, d_D;\\
&\sigma = u^2, u \sim \text{Uniform}(0.1, 1.1).
\end{aligned}$$

From each component $\lceil 100k\zeta_i \rceil$ data were generated according to,

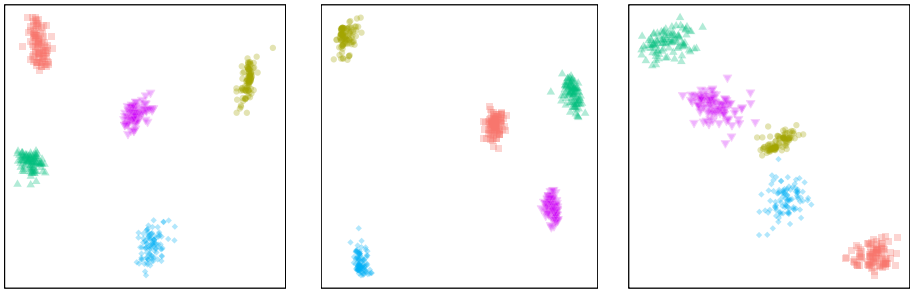
$$\begin{aligned}
&\mathbf{x}^C \sim N(\boldsymbol{\mu}^C, \sigma\mathbf{I}),\\
&\mathbb{P}(x_j^D = B) = \begin{cases} 1 - \sigma/4, & B = \mu_j^D \\ \sigma/4, & B = 1 - \mu_j^D \end{cases}.
\end{aligned}$$

The model for the continuous attributes of the data tends to induce greater separability between clusters in datasets with higher numbers of clusters and higher dimensionality. We denote this data generating process as MixGen^2.

Typical examples of the structure within the mixed data are given in Figures 3.1 and 3.2. These provide the two-dimensional representations from MDS, CSE and mPPCA of datasets generated by MixGen^1 and MixGen^2 respectively, each with different numbers of dimensions and five clusters. It is worth noting that these low-dimensional representations do not necessarily capture all of the structure in the higher-dimensional representations used for clustering. In particular, the data will be much more sparse in more dimensions, and this can make cluster detection more challenging. For the data generated by MixGen^1, all three continuous representations appear similar and seem to provide appropriate structures for clustering. For the lower-dimensional examples, there tend to be multiple dense regions within each of the clusters and the clusters are less separable. However, in the higher-



(a) MDS : $d_C/d_D = 5/5$ (b) CSE : $d_C/d_D = 5/5$ (c) mPPCA : $d_C/d_D = 5/5$

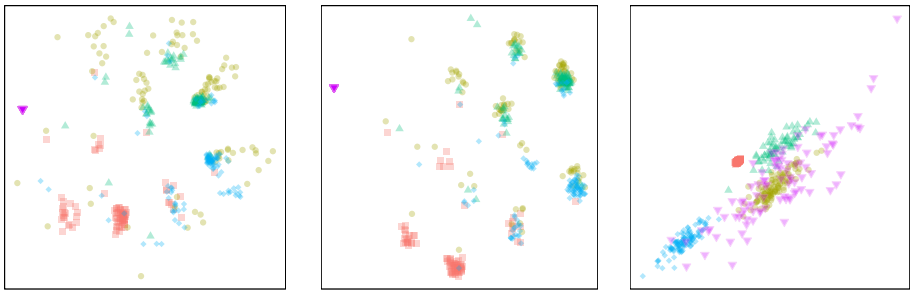


(d) MDS : $d_C/d_D = 20/20$ (e) CSE : $d_C/d_D = 20/20$ (f) mPPCA : $d_C/d_D = 20/20$

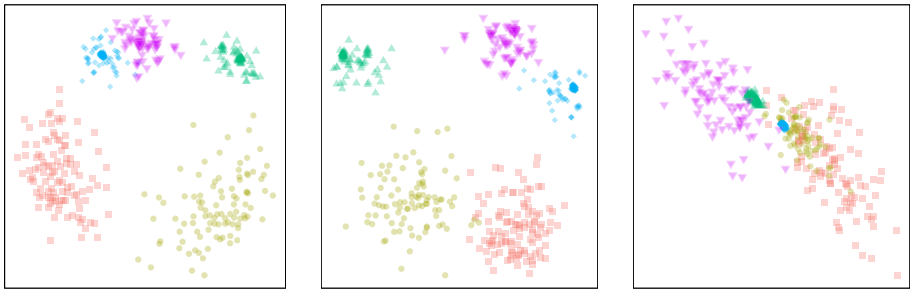
Figure 3.1: Example structure in continuous representation of simulated mixed data generated by MixGen[1].

dimensional datasets this is not the case and the clustering structure is very clear, so we would expect good clustering performance by any algorithm considered on these representations.

The continuous representations produced by MixGen[2] are more varied, with a clear difference between the approaches using the dissimilarity matrix and mPPCA. Since the modelling assumptions made by mPPCA are violated in these examples, the resulting continuous representation is much less appropriate for clustering than for the previous generative model. For the lower-dimensional examples, the continuous representations from MDS and CSE have very dense regions around the atoms of the distribution of $\mathbf{x}^D$ relative to the variability in $\mathbf{x}^C$. Thus, these continuous representations have multiple dense regions, which do not contain observations originating from a single true cluster. Hence, we expect clustering results produced from MDS and CSE representations to overestimate the number



(a) MDS : $d_C/d_D = 5/5$ (b) CSE : $d_C/d_D = 5/5$ (c) mPPCA : $d_C/d_D = 5/5$



(d) MDS : $d_C/d_D = 20/20$ (e) CSE : $d_C/d_D = 20/20$ (f) mPPCA : $d_C/d_D = 20/20$

Figure 3.2: Example structure in continuous representation of simulated mixed data generated by MixGen^2.

of clusters in the 10-dimensional datasets. Increasing the dimensionality permits a continuous representation in which the clusters are more clearly separable, so we expect all the clustering algorithms considered to perform well on the CSE and MDS representations of the 40-dimensional datasets.

Tables 3.6.1 and 3.6.1 show the clustering performance of the five algorithms considered when applied to the continuous representations of data simulated by MixGen^1 and MixGen^2 from MDS, CSE and mPPCA with respect to NMI and the number of clusters located. The top row of each cell gives the mean NMI over 30 replications of each scenario with results from MDS, CSE and mPPCA respectively, separated by a comma. The bottom row of each cell has the same format but for the average number of clusters located. For each algorithm and each scenario, the best continuous representation is highlighted. For the data generated by MixGen^1, all the continuous representations are competitive. In the 10-

Table 3.1: Mean clustering performance with respect to NMI and estimated number of clusters from MDS,CSE,mPPCA representations of data generated by MixGen[1]. The best continuous representation for each scenario and choice of clustering algorithm is highlighted in red.

			$k = 5$	$k = 10$	k=20
d_C/d_D 5/5	MDH_{hier}	NMI	0.816,0.810,0.451	0.846,0.824,0.314	0.768,0.781,0.331
		k	12.7,11.5,6.4	23.7,23.8,7.1	45.6,54.0,13.9
	MDH_{ens}	NMI	0.736,0.752,0.402	0.710,0.722,0.410	0.667,0.644,0.276
		k	8.9,8.5,5.3	14.2,14.8,8.2	22.3,21.8,9.5
	dePDDP	NMI	0.732,0.360,0.544	0.732,0.535,0.473	0.696,0.402,0.358
		k	27.7,16.3,16.8	47.0,44.9,24.4	93.3, 63.1,26.8
	k-means++ (Gap)	NMI	0.841,0.825,0.589	0.869,0.873,0.606	0.788,0.847,0.450
		k	9.8,9.8,7.1	19.4,19.2,15.3	39.3,39.1,24.1
	GMM	NMI	0.592,0.581,0.570	0.716,0.644,0.604	0.609,0.531,0.433
		k	8.9,9.0,4.9	9.0,9.0,6.2	8.9,9.0,5.5
d_C/d_D 10/10	MDH_{hier}	NMI	0.785,0.809,0.661	0.864,0.877,0.563	0.886,0.881,0.718
		k	13.0,12.9,9.3	24.4,21.4,14.4	48.8,47.7,32.8
	MDH_{ens}	NMI	0.760,0.763,0.532	0.792,0.831,0.373	0.735,0.778,0.424
		k	8.0,8.3,6.8	13.6,13.3,6.7	21.1,21.0,13.3
	dePDDP	NMI	0.464,0.373,0.718	0.806,0.540,0.563	0.779,0.448,0.604
		k	12.5,13.7,15.7	50.9,34.4,27.3	95.2, 55.3,53.3
	k-means++ (Gap)	NMI	0.778,0.798,0.768	0.904,0.905,0.620	0.920,0.922,0.731
		k	9.7,9.6,8.3	19.1,19.1,13.3	39.0,39.1,31.3
	GMM	NMI	0.749,0.636,0.766	0.873,0.781,0.642	0.681,0.668,0.626
		k	7.5,8.7,4.2	8.9,9.0,6.1	9.0,8.9,7.1
d_C/d_D 20/20	MDH_{hier}	NMI	0.299,0.806,0.816	0.842,0.879,0.706	0.903,0.930,0.540
		k	5.2,13.4,10.6	22.8,22.3,17.5	42.4,36.0,21.5
	MDH_{ens}	NMI	0.797,0.802,0.630	0.804,0.862,0.517	0.791,0.853,0.402
		k	7.8,8.5,7.9	12.0,13.4,10.3	20.7,21.2,9.9
	dePDDP	NMI	0.125,0.546,0.789	0.335,0.608,0.729	0.331,0.759,0.559
		k	2.9,11.8,17.3	16.1,25.3,26.8	30.2,71.1,30.3
	k-means++ (Gap)	NMI	0.655,0.834,0.839	0.868,0.904,0.749	0.924,0.928,0.609
		k	9.7,9.6,9.2	19.4,19.6,15.7	38.8,38.7,25.3
	GMM	NMI	0.725,0.865,0.879	0.810,0.819,0.768	0.701,0.695,0.430
		k	6.7,8.1,3.9	7.5,8.9,6.7	8.5,9.0,3.6

dimensional datasets with 5 or 10 clusters, MDS tends to produce the most appropriate representation for clustering. In the datasets with 20 clusters, both MDS and CSE perform well with similar results for all algorithms except dePDDP, where MDS seems to allow better performance for the 10 and 20-dimensional datasets, while the CSE representation is better for the 40-dimensional datasets. In the 40-dimensional datasets, mPPCA also performs well,

Table 3.2: Mean clustering performance with respect to NMI and estimated number of clusters from MDS,CSE,mPPCA representations of data generated by MixGen^2. The best continuous representation for each scenario and choice of clustering algorithm is highlighted in red.

			$k=5$	$k=10$	k=20
d_C/d_D 5/5	MDH_{hier}	NMI k	0.546,0.586,0.137 5.7,10.8,1.5	0.680,0.621,0.305 11.2,20.7,3.2	0.761,0.639,0.282 31.1,38.3,5.7
	MDH_{ens}	NMI k	0.643,0.629,0.348 8.7,8.0,5.1	0.627,0.635,0.355 13.6,14.9,6.7	0.633,0.634,0.288 24.5,24.8,6.8
	dePDDP	NMI k	0.531,0.639,0.242 15.4,34.2,5.0	0.648,0.698,0.317 35.7,64.8,8.8	0.729,0.748,0.295 79.6,142.9,12.7
	k-means++ (Gap)	NMI k	0.631,0.621,0.312 9.6,9.8,4.5	0.757,0.641,0.374 19.8,19.5,9.4	0.826,0.649,0.314 38.8,39.2,13.1
	GMM	NMI k	0.619,0.613,0.490 6.7,7.2,2.9	0.680,0.674,0.432 8.9,9.0,4.6	0.603,0.603,0.278 9.0,9.0,3.9
d_C/d_D 10/10	MDH_{hier}	NMI k	0.488,0.806,0.220 3.5,5.9,1.8	0.826,0.739,0.111 9.4,13.6,1.9	0.948,0.699,0.213 22.7,26.9,5.1
	MDH_{ens}	NMI k	0.677,0.680,0.240 8.5,7.9,3.4	0.629,0.619,0.146 12.3,12.7,3.0	0.596,0.598,0.189 21.4,21.6,5.2
	dePDDP	NMI k	0.700,0.729,0.199 15.3,35.4,3.5	0.706,0.704,0.143 28.7,71.3,3.5	0.729,0.656,0.166 61.5,118.3,7.8
	k-means++ (Gap)	NMI k	0.676,0.811,0.245 9.7,9.0,3.7	0.840,0.816,0.135 19.7,19.4,3.8	0.918,0.830,0.211 39.1,39.4,9.3
	GMM	NMI k	0.691,0.632,0.311 6.2,5.0,2.2	0.758,0.675,0.162 8.4,8.7,2.2	0.560,0.555,0.177 9.0,9.0,2.9
d_C/d_D 20/20	MDH_{hier}	NMI k	0.217,0.949,0.049 2.0,4.8,1.2	0.629,0.964,0.033 7.4,10.0,1.3	0.976,0.935,0.033 20.9,20.6,1.6
	MDH_{ens}	NMI k	0.692,0.657,0.086 8.0,7.5,1.9	0.662,0.657,0.023 19.7,11.8,1.3	0.609,0.609,0.013 19.7,19.3,1.9
	dePDDP	NMI k	0.644,0.811,0.049 10.2,35.8,1.5	0.817,0.818,0.030 22.3,70.6,1.6	0.790,0.752,0.032 44.9,41.2,1.7
	k-means++ (Gap)	NMI k	0.601,0.889,0.075 9.0,7.5,1.8	0.858,0.925,0.031 18.9,17.3,1.5	0.926,0.928,0.031 38.6,38.3,2.1
	GMM	NMI k	0.677,0.629,0.124 6.9,6.9,1.5	0.787,0.660,0.025 9.0,7.4,1.1	0.542,0.475,0.024 8.9,8.5,1.3

outperforming the other two approaches in some instances. With the exception of GMM, all the clustering algorithms considered, tend to overestimate the number of clusters in the continuous representations from CSE and MDS. However, the continuous representations from mPPCA seem less susceptible to this.

The results from the data generated by MixGen^2 indicate that when the modelling as-

sumptions of mPPCA are not satisfied, MDS and CSE produce much more appropriate continuous representations for clustering. Both MDS and CSE are capable of producing appropriate representations, and neither representation consistently permits better clustering performance than the other. For the 10-dimensional datasets, both CSE and MDS perform similarly, although CSE generally allows the best clustering performance for dePDDP and MDH_{ens}, while MDS produces a slightly better representation for clustering with MDH_{hier} , k-means++ and GMM. For the 20-dimensional datasets, CSE is generally the most appropriate continuous representation when there are only five clusters, however, the MDS representations are more appropriate with 10 or 20 clusters. This is similar for the 40-dimensional datasets, although the difference in performance between CSE and MDS is less significant in the datasets with 10 or 12 clusters. k-means++ and dePDDP tend to overestimate the number of clusters in the MDS and CSE representations, while MDH_{hier}, MDH_{ens} and GMM more accurately estimate this, except for the datasets with 20 clusters where GMM only locates about nine clusters. All algorithms underestimate the number of clusters in the mPPCA representations, indicating the clusters are not clearly separable in these representations.

3.6.2 Real Data

In this section, we consider the quality of the partitions produced by the different clustering algorithms considered on the continuous representations of real-world benchmark datasets produced by MDS, CSE and mPPCA. All of these datasets are available from the UCI repository (Lichman, 2013). Table 3.6.2 provides a summary of the datasets used with respect to the number of observations, n, number of continuous dimensions d_C, number of discrete dimensions d_D and number of clusters k.

Table 3.3: Summary of mixed benchmark datasets.

Dataset	n	d_C	d_D	k
Autodata	392	5	3	5
Credit Approval	690	5	10	2
Dermatology	366	-	34	6
Heart Disease	294	5	8	2
Soybean	682	-	35	19
Voters	435	-	16	2

The two-dimensional continuous representations of each of the datasets resulting from MDS, CSE and mPPCA are given in Figure 3.3. With the exception of Autodata, these datasets appear much more challenging than the simulated data, with a much less clear clustering structure in the continuous representations, so it is expected that the clustering performance will be poor in most cases. The representations from mPPCA are generally the least appropriate for the identification of the true clusters.

Table 3.4 provides the performance of the five clustering algorithms considered when applied to the continuous representations resulting from MDS, CSE and mPPCA. The CSE representations tend to result in the best clustering performance, followed by the MDS representations. For all datasets except Heart Disease, mPPCA provides very poor clustering performance. For *k*-means++, CSE produces the most appropriate continuous representation for all datasets except Heart Disease. This is expected since this representation explicitly considers the *k*-means objective. CSE also provides the best representations for dePDDP (except for the Heart Disease dataset) and MDH_{ens}. For MDH_{hier} CSE produces the best representations for the Dermatology, SoyBean and Voters datasets, while the clustering performance of this algorithm is marginally better on the MDS representations of the Autodata and Credit Approval datasets. For GMM, the CSE representations allow the most accurate clustering results, except for the Credit Approval and SoyBean datasets, where the MDS representation is more appropriate. The performance of all clustering algorithms is poor when

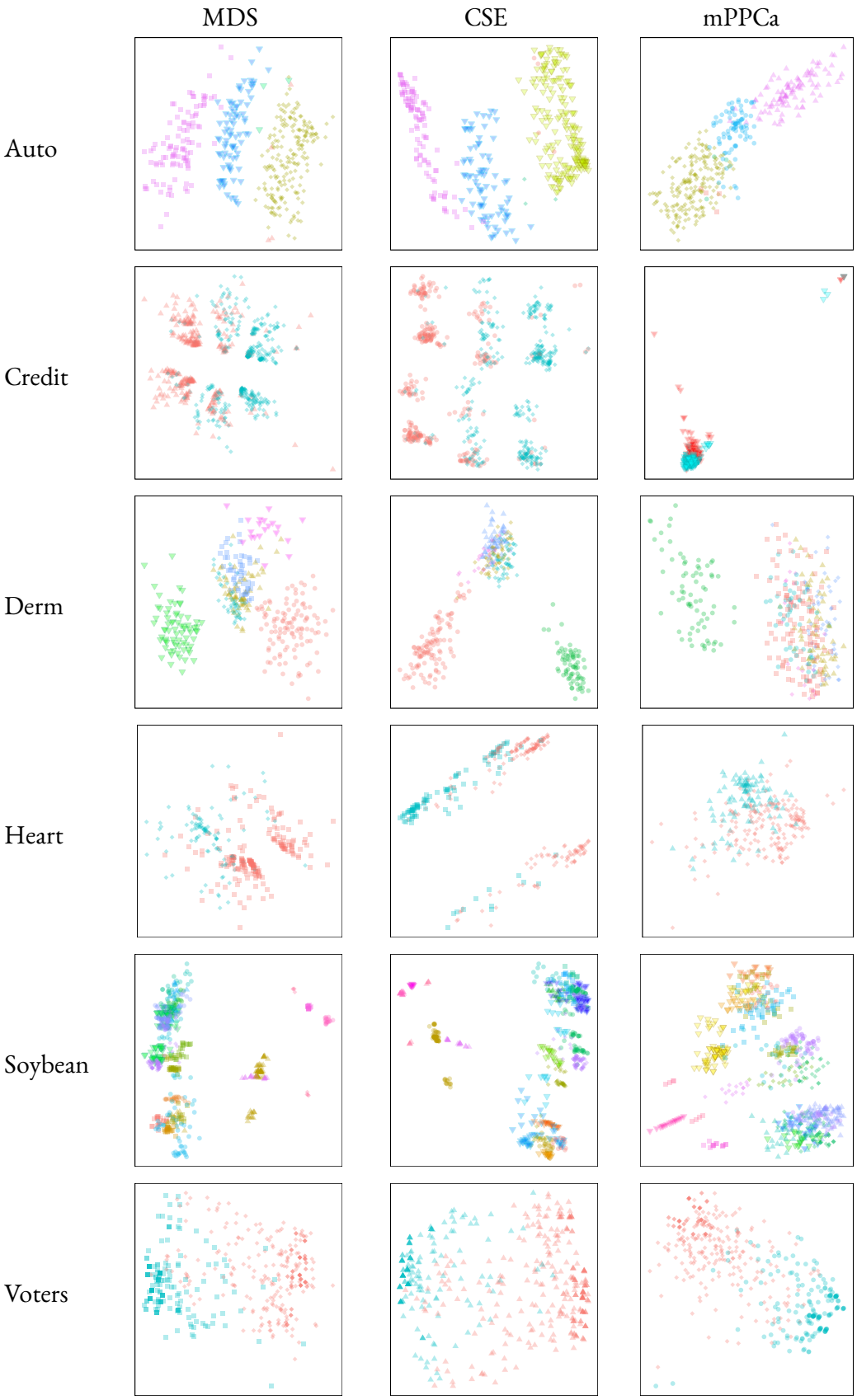


Figure 3.3: Two-dimensional continuous representations of real datasets from MDS, CSE and mPPCA.

Table 3.4: Clustering performance with respect to NMI across real benchmark datasets. The representation which resulted in the best performance for each clustering algorithm is highlighted in red.

			Auto	Credit	Derm	Heart	Soybean	Voters
MDS	MDH_{hier}	NMI	0.799	0.297	0.000	0.622	0.000	0.296
		k	4.0	5.0	1.0	15.0	1.0	2.0
	MDH_{ens}	NMI	0.642	0.163	0.522	0.253	0.431	0.445
		k	8	4	10	4	11	4
	dePDDP	NMI	0.516	0.020	0.000	0.549	0.000	0.000
		k	4.0	2.0	1.0	22.0	1.0	1.0
	k-means++ (Gap)	NMI	0.632	0.000	0.710	0.765	0.203	0.260
		k	10.0	1.0	11.0	37.0	1.0	1.0
	GMM	NMI	0.590	0.069	0.697	0.358	0.204	0.315
		k	9.0	7.0	4.0	4.0	6.0	3.0
CSE	MDH_{hier}	NMI	0.778	0.241	0.909	0.239	0.705	0.337
		k	5.0	9.0	5.0	6.0	21.0	5.0
	MDH_{ens}	NMI	0.908	0.287	0.843	0.263	0.658	0.492
		k	3.0	9.0	5.0	7.0	8.0	3.0
	dePDDP	NMI	0.674	0.258	0.860	0.225	0.727	0.395
		k	8.0	22.0	8.0	12.0	45.0	3.0
	k-means++ (Gap)	NMI	0.635	0.245	0.772	0.267	0.781	0.433
		k	10.0	4.0	10.0	4.0	38.0	4.0
	GMM	NMI	0.631	0.015	0.700	0.539	0.000	0.353
		k	9.0	2.0	3.0	6.0	2.0	8.0
mPPCA	MDH_{hier}	NMI	0.000	0.000	0.000	0.639	0.000	0.294
		k	1.2	1.0	1.0	16.8	1.0	7.3
	MDH_{ens}	NMI	0.401	0.069	0.122	0.049	0.418	0.237
		k	8.3	3.5	6.5	4.0	11.4	4.0
	dePDDP	NMI	0.000	0.000	0.339	0.000	0.000	0.000
		k	2.1	1.0	9.3	1.0	1.0	1.0
	k-means++ (Gap)	NMI	0.000	0.000	0.552	0.738	0.000	0.356
		k	7.4	1.3	11.6	36.5	1.0	3.8
	GMM	NMI	0.348	0.055	0.000	0.528	0.076	0.309
		k	5.1	4.3	1.3	4.3	6.5	4.0

applied on the mPPCA representations of the Autodata, Credit Approval, Dematology and SoyBean Datasets. However, the mPPCA representations of the Heart Disease and Voters datasets are competitive when using some of the clustering algorithms considered. For the mPPCA representation of the Voters dataset, the clustering performance of all algorithms is comparable to the MDS representation. The mPPCA representation of the Heart Disease dataset produces the best performance for MDH_{hier} and is also competitive for k-means++

and GMM.

To assess the relative performance of each of the continuous representations for the five clustering algorithms considered, Figure 3.4 provides boxplots of the regret associated with using each representation. For each clustering algorithm on each dataset, the regret associated with a representation is defined as the difference between the performance (based on NMI) on the representation of interest and the best performing representation for that algorithm on that dataset,

$$\text{Regret}(\mathcal{R}) = \text{NMI}(\boldsymbol{\pi}_{\mathcal{R}^\star}, \boldsymbol{\pi}^\star) - \text{NMI}(\boldsymbol{\pi}_{\mathcal{R}}, \boldsymbol{\pi}^\star)$$

where $\mathcal{R}$ is the continuous representation in question, $\boldsymbol{\pi}_{\mathcal{R}}$ is the partition produced on this representation and $\boldsymbol{\pi}_{\mathcal{R}^\star}$ is the partition produced on the best performing continuous representation. These regret values are grouped according to the different algorithms applied. A regret close to zero indicates the best relative performance of a continuous representation. It is important to note that this solely compares the quality of the continuous representation for clustering by each algorithm and not the relative performance of each of the algorithms on a given continuous representation. For all algorithms, CSE minimises the regret, indicating that this representation is the most appropriate for clustering with these algorithms. For GMM, MDS also achieves a low regret, indicating that for GMM, the choice of continuous representation is not as critical as for the the other algorithms.

3.7 Conclusions

In this chapter, we investigated three methods for producing continuous representations of mixed datasets, and their appropriateness for clustering with projective density-based and well-established algorithms. The methods investigated were MDS, mPPCA and CSE.

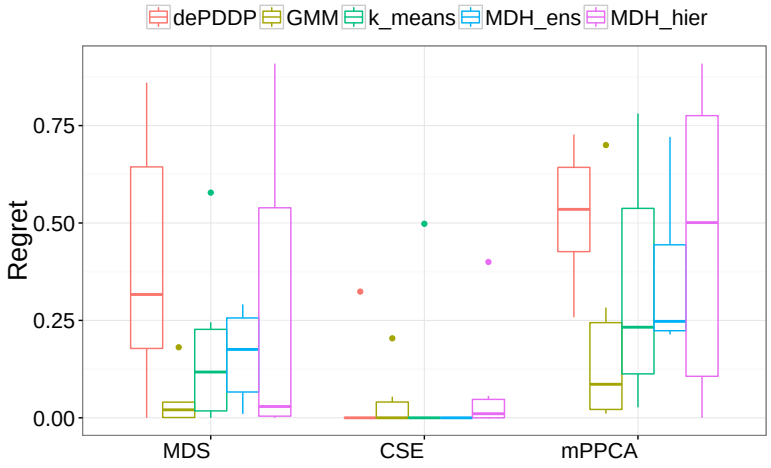


Figure 3.4: Boxplot of regret based on NMI for continuous representations produced by MDS, CSE and mPPCA.

Through a systematic simulation study, we have shown that if the generative model assumed by mPPCA is satisfied, all three continuous representations can produce an appropriate representation for effective cluster detection. However, for different generative models, the representation from mPPCA is much less competitive than CSE and MDS, which make no assumptions about the data generating processes. Over real benchmark datasets with varying characteristics, CSE produced the most appropriate continuous representation, while MDS and mPPCA had a more varied performance. In general, the real datasets were challenging, so consistently high-quality results were not possible for any of the continuous representations, instead the ability to locate meaningful clusters was dependant on both the continuous representation and the choice of clustering algorithm.

4

Combining Hyperplane Separators for Clustering

Abstract

We propose approaches to perform density-based clustering of high-dimensional datasets that may contain diverse (mixed) attributes, which are able to identify clusters in arbitrarily oriented subspaces and estimate their number. For mixed datasets, we obtain an appropriate continuous representation. Thereafter, we perform projection pursuit on the continuous data or continuous representation of the mixed data, to locate low-density linear separators that partition high-density regions associated with clusters. By combining binary partitions from multiple separators we obtain a divisive and a partitional clustering algorithm to produce a complete clustering. The resulting clusters concentrate around the modes of the estimated density of the data (or its continuous representation where necessary). Through empirical evaluation across simulated and real-world benchmark datasets with varying characteristics, we show that the proposed algorithms produce consistently high-quality results, and that their performance is competitive with alternative density-based and other state-of-the-art clustering algorithms.

4.1 Introduction

Given a set of observations $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$, the objective of clustering is to partition $\mathcal{X}$ into a number of homogeneous subsets, or *clusters*, so that observations allocated to the same cluster are more similar to each other, than observations allocated to different clusters. As there is no unique and universally accepted definition of a cluster, there are a number of approaches to clustering, each relying on a different definition.

The non-parametric statistical approach to clustering, commonly referred to as *density-based clustering*, assumes that $\mathcal{X}$ is a sample of realisations of a continuous random variable X with unknown probability density function. Clusters are then defined as regions of high probability density surrounding the modes of the density function (Hartigan, 1975; Menardi, 2016).

Since the true density function is unlikely to be known in practice, its modes must be located using an non-parametric density estimate. This imposes limitations on the applicability of density-based clustering in a number of practical applications. Firstly, density estimation is unreliable in even moderate dimensions. This problem, commonly referred to as the curse of dimensionality, makes the detection of dense regions associated with clusters challenging, unless the clusters are very well separated (Rinaldo and Wasserman, 2010). In addition, if the observations contain any non-continuous attributes, which is common in many applications, the construction of a continuous density estimate is inappropriate. If one were to construct a continuous estimator over such data, subsequent cluster detection would trivially separate observations with the same combinations of outcomes in the discrete dimensions. We propose an approach to overcome these restrictions. We consider an alternative formulation of density-based clustering, which remains applicable in high dimensions, as well as applying continuous representations of mixed data to allow this methodology to be

applied to datasets with large numbers of diverse attributes.

A direct consequence of defining clusters around the modes of a probability density function is that cluster boundaries pass through contiguous regions of low probability density, that separate the modes. This alternative formulation, known as the *low-density separation assumption*, underpins well-established algorithms such as maximum margin clustering (MMC) (Xu et al., 2004) and semi-supervised support vector machines (Joachims, 1999). These methods extend the maximum margin hyperplane approach, and have proved very successful in clustering and semi-supervised classification respectively. The justification for using the maximum margin hyperplane to partition unlabelled data is that it approximates the hyperplane that goes through the most sparse regions of the empirical density (Chapelle and Zien, 2005; Chapelle et al., 2006).

Ben-David et al. (2009) were the first to consider the learning problem associated with estimating the hyperplane which intersects the region of lowest probability density, under the minimal set of assumptions that $\mathcal{X}$ is an iid sample from an unknown probability distribution over $\mathbb{R}^d$ with continuous density. The authors quantify the *density on a hyperplane* as the integral of the probability density function along the hyperplane, and study the existence of universally consistent algorithms to estimate the hyperplane with minimum density. They find that the maximum hard margin classifier is a consistent estimator of the hyperplane with minimum density only in one-dimensional problems, while in higher dimensions only a soft-margin algorithm is consistent. Pavlidis et al. (2016) propose a method to compute the hyperplane with minimum density for a finite high-dimensional sample using one-dimensional projections of the data, and establish an asymptotic connection between this hyperplane and the maximum hard margin hyperplane.

The only work which we are aware of that applies density-based clustering to mixed data

is Azzalini and Menardi (2016). This work first applies multi-dimensional scaling (MDS) (Borg and Groenen, 2005) to produce a low-dimensional continuous representation before using the pdfCluster algorithm (Menardi and Azzalini, 2014). We also consider continuous representations produced by mixed probabilistic principal component analysis (mPPCA) (Khan et al., 2010) and constant shift embedding (CSE) (Roth et al., 2003). Due to our alternative formulation of density-based clustering, we also remove the restriction to a low-dimensional continuous embedding, which is more appropriate for datasets with larger numbers of clusters.

In this chapter, we address the aforementioned limitations of density-based clustering associated with high-dimensional and mixed data. We develop a divisive hierarchical clustering algorithm and a partitional ensemble clustering algorithm, which use low-density separators to identify dense clusters associated with the modes of the estimated continuous probability density function. These are obtained through one-dimensional projections of the data, making this applicable in high-dimensional applications, where the construction of an estimated density over all dimensions is infeasible. In the case of mixed observations, we first locate a continuous representation before attempting to identify clusters. Our algorithms can identify clusters in different arbitrarily orientated subspaces, as well as estimate their number.

The remainder of this chapter is organised as follows: Section 4.2 presents the methodology for the proposed algorithms. First, we formulate the problem of projective density-based clustering for bi-partitioning, and then present our approaches for producing a full clustering based on these binary partitions. Next, Section 4.3 considers the production of a continuous representation of mixed data, allowing our algorithms to be applied in such datasets. Section 4.4 provides a comparative evaluation of the proposed algorithms against

alternative density-based and state-of-the-art clustering algorithms on simulated and real-world datasets. The chapter ends with conclusions in Section 4.5.

4.2 Methodology

It is assumed throughout that the set of observations, $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$, constitutes a sample of realisations of a continuous random variable, or continuous representation of a mixed random variable X on $\mathbb{R}^d$ with unknown continuous probability density function, approximated by $\hat{p}_{\mathbf{x}} : \mathbb{R}^d \to \mathbb{R}_+$. The proposed approach aims to identify hyperplanes that traverse regions of low density, and separate dense regions around the modes of $\hat{p}_{\mathbf{x}}$ that are associated with clusters. We define high-density clusters based on the estimated density $\hat{p}_{\mathbf{x}}$ as in Section 2.3, Definition 1.

To identify high-density clusters in high-dimensional datasets, we apply the low-density separation assumption to define cluster boundaries, rather than locating the level sets of $\hat{p}_{\mathbf{x}}$ directly. We define a low-density separator, that identifies high-density clusters in $\mathcal{X}$ according to Definition 2 in Section 2.3. An important parameter in both Definitions 1 and 2 is the level parameter c, that sets a threshold on the maximum value of the density intersected by a cluster boundary, such that contiguous regions of density greater than c are separated. The proposed algorithms do not require the determination of this parameter in advance, but instead attempt to identify the separator with minimal density. This results in the separator that corresponds to the smallest value of c for which Definition 2 holds. These separators produce a a succession of bi-partitions, which are combined to produce an overall clustering. The location of these minimum density separators is computationally intractable for arbitrary separators, so we restrict or attention to linear separators (hyperplanes). These minimum density linear separators located by the approaches proposed in this chapter bi-

partition dense linearly separable sets, as defined in Section 2.3, Definition 3. This definition posits that convex, contiguous regions of density greater than c are linearly separable if there exists a hyperplane along which the maximum value of $\hat{p}_{\mathbf{x}}$ is at most c. This definition results in the subsets of $\mathcal{X}$ identified by a minimum density linear separator corresponding to high-density clusters, as defined in Definition 1, with the constraint of convexity in the clusters.

4.2.1 Minimum Density Hyperplanes

A hyperplane can be defined by a unit-length vector $\mathbf{v} \in \mathbb{S}^{d-1} = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\| = 1\}$ and a displacement from the origin $b \in \mathbb{R}$, as $H(\mathbf{v}, b) = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{v}^\top \mathbf{x} = b\}$. To quantify the density of the region intersected by a hyperplane with respect to $\hat{p}_{\mathbf{x}}$ we adapt the *density on a hyperplane* criterion proposed by Ben-David et al. (2009),

$$\hat{I}(\mathbf{v}, b) = \int_{\mathbf{x} \in H(\mathbf{v},b)} \hat{p}_{\mathbf{x}}(\mathbf{x}) \mathrm{d}\mathbf{x}. \tag{4.1}$$

The hyperplane that minimises $\hat{I}(\mathbf{v}, b)$ is called the *minimum density hyperplane* (MDH). $\hat{I}(\mathbf{v}, b)$ cannot be evaluated analytically for all types of density estimators, but when $\hat{p}_{\mathbf{x}}$ is constructed from an isotropic Gaussian kernel density estimate Eq. (4.1) simplifies greatly,

$$\begin{aligned} \hat{I}(\mathbf{v}, b) &= \int_{\mathbf{x} \in H(\mathbf{v},b)} \frac{1}{n(2\pi h^2)^{d/2}} \sum_{i=1}^{n} \exp\left\{ -\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2h^2} \right\} \mathrm{d}\mathbf{x}, \\ &= \frac{1}{n\sqrt{2\pi h^2}} \sum_{i=1}^{n} \exp\left\{ -\frac{(b - \mathbf{v}^\top \mathbf{x}_i)^2}{2h^2} \right\}, \\ &= \hat{p}_{\mathbf{v}^T \mathbf{x}}(b), \end{aligned} \tag{4.2}$$

where $\hat{p}_{\mathbf{v}^T\mathbf{x}}$ denotes a one-dimensional kernel density estimator constructed from the projection of $\mathcal{X}$ onto $\mathbf{v}$, and using the same bandwidth, h, as $\hat{p}_{\mathbf{x}}$. Eq. (4.2) states that $\hat{I}(\mathbf{v}, b)$ can be computed exactly by projecting the data onto $\mathbf{v}$; constructing a one-dimensional density estimator from these projections that uses Gaussian kernels with bandwidth h; and evaluating it at b. Since projections can only contract pairwise distances, it can be shown that $\hat{I}(\mathbf{v}, b)$ imposes an upper bound on the estimated density at any point on the hyperplane $H(\mathbf{v}, b)$ (Pavlidis et al., 2016),

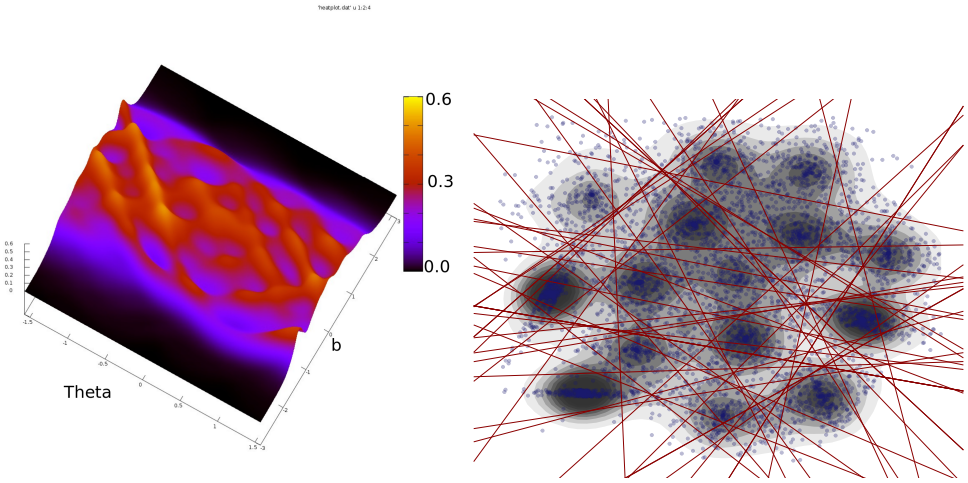
$$\max_{\mathbf{x} \in H(\mathbf{v},b)} \hat{p}_{\mathbf{x}}(\mathbf{x}) \leqslant (2\pi h^2)^{\frac{(1-d)}{2}} \hat{I}(\mathbf{v}, b).$$

This bound is tight if only one-dimensional projections of $\mathcal{X}$ are used. Therefore, the MDH imposes the lowest upper bound (that can be achieved using one-dimensional projections only) on the maximum value of $\hat{p}_{\mathbf{x}}$ along a hyperplane separator.

Assuming without loss of generality that $\mathcal{X}$ is centred at zero, the MDH is the solution to the optimisation problem,

$$\min_{\mathbf{v},b} \hat{I}(\mathbf{v}, b), \text{ s.t. } b \in [-\alpha\sigma_{\mathbf{v}}, \alpha\sigma_{\mathbf{v}}], \tag{4.3}$$

where $\sigma_{\mathbf{v}}$ denotes the standard deviation of the projected data onto $\mathbf{v}$, and $\alpha > 0$ is a user defined parameter controlling the width of the search interval for b, discussed in detail below. It is necessary to constrain the displacement of the separating hyperplane from the origin, $|b|$, as for any $\mathbf{v} \in \mathbb{S}^{d-1}$, a hyperplane of arbitrarily low density can be found for sufficiently large $|b|$, that is $\lim_{|b|\to\infty} \hat{I}(\mathbf{v}, b) = 0$. Such hyperplanes are clearly not meaningful for clustering as they assign all observations to one cluster. The constrained optimisation problem in Eq. (4.3) exhibits multiple local minima, as demonstrated in Figure 4.1, which



(a) $\hat{I}(\mathbf{v}, b)$ (b) Hyperlane separators from SQP

Figure 4.1: Illustration of local minima $\hat{I}(\mathbf{v}, b)$ and the resulting hyperplane separators from constrained optimisation with 50 random initialisations for the S4 dataset.

shows the value of $\hat{I}(\mathbf{v}, b)$ with changes in the projection angle and displacement from the origin, as well as the resulting hyperplane separators obtained through sequential quadratic programming (SQP) (with 50 random initialisations) over the S4 dataset (Fränti and Virmajoki, 2006).

To alleviate the problem of convergence to poor local minima, the following projection pursuit formulation has been proposed (Pavlidis et al., 2016),

$$\phi(\mathbf{v}) = \min_{b \in \mathbb{R}} f(\mathbf{v}, b), \tag{4.4}$$

$$f(\mathbf{v}, b) = \hat{I}(\mathbf{v}, b) + \frac{L}{\eta^{\varepsilon}} \max\{0, -\alpha\sigma_{\mathbf{v}} - b, b - \alpha\sigma_{\mathbf{v}}\}^{1+\varepsilon}, \tag{4.5}$$

where $L = (e^{1/2}h^2 2\pi)^{-1} \geqslant \sup_{b \in \mathbb{R}} |\hat{p}'_{\mathbf{v}^T\mathbf{x}}(b)|$ and $\varepsilon, \eta \in (0, 1)$. We call f the penalised density integral, and ϕ the projection index, as it quantifies the suitability of of projection vectors for low-density cluster separation. The choice of L ensures that for fixed $\mathbf{v}$ the global minimiser of $f(\mathbf{v}, b)$ will be within η of the minimiser of $\hat{I}(\mathbf{v}, b)$ in the interval $[-\alpha\sigma_{\mathbf{v}}, \alpha\sigma_{\mathbf{v}}]$ (Pavlidis et al., 2016). The parameter ε is introduced to ensure that the penalty

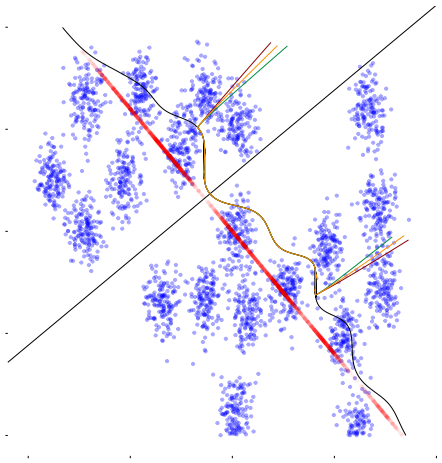


Figure 4.2: Separating hyperplane $H(\mathbf{v}, b)$, estimated density of the projections of $\mathcal{X}$ onto $\mathbf{v}$ (black line), $\hat{I}(\mathbf{v}, \cdot)$, and penalised objective function, $f(\mathbf{v}, \cdot)$, for $\eta = 0.01$ and $\varepsilon = \{0.1, 0.3, 0.9\}$ (burgundy,orange and green lines respectively).

function is continuously differentiable everywhere, while resembling the hinge loss function. For η and ε, values close to zero and one are recommended respectively. Fig. 4.2 illustrates the two dimensional A1 dataset (Kärkkäinen and Fränti, 2002), along with a candidate separating hyperplane (black line). The observations projected onto the vector perpendicular to the separating hyperplane are illustrated with red dots. The one-dimensional kernel density estimator constructed from these projections, $\hat{p}_{\mathbf{v}^\top \mathbf{x}}$, is also illustrated along with the penalised density integral, $f(\mathbf{v}, \cdot)$, for three choices of (η, ε). The figure illustrates the effect of the penalty function, which is to ensure that all minimisers of $f(\mathbf{v}, \cdot)$ are identical to the minimisers of $\hat{I}(\mathbf{v}, \cdot)$ in $[-\alpha\sigma_{\mathbf{v}}, \alpha\sigma_{\mathbf{v}}]$ and differ by at most η at the boundaries. The figure also shows that the precise choices of η and ε are not critical, but sensible values are required to avoid numerical instability.

The parameter α determines the range over which minimisers of $\hat{I}(\mathbf{v}, \cdot)$ are sought. If α is constant, then its value critically affects the quality of the estimated MDH. Setting α close to zero favours hyperplanes that induce a balanced bi-partition of $\mathcal{X}$, but there is no guarantee that clusters can be separated by a hyperplane that goes through the mean of the data. If instead a large value of α is used there is a risk that the MDH will separate the tail

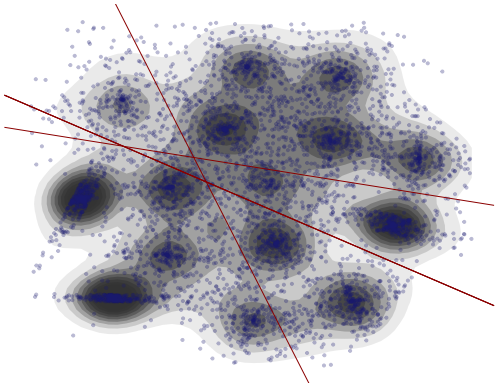


Figure 4.3: Illustration of the resulting hyperplane separators from the projection pursuit formulation with 50 random initialisations for the S4 dataset.

of $\hat{p}_{\mathbf{x}}$ rather than separating high-density regions. Instead of selecting a fixed value, it has been recommended in Pavlidis et al. (2016) to estimate the MDH for a sequence of increasing values of α, starting from zero, and using the previously identified MDH as the initial projection direction each time α is increased. Setting α to zero initially forces the algorithm to seek low-density hyperplanes that induce a balanced bi-partition of high-density clusters, while increasing α in subsequent steps fine tunes the location of the MDH. The maximum value of α is not critical in this approach as it is straightforward to detect when the MDH is no longer a local minimiser of $\hat{I}(\mathbf{v}, \cdot)$ but instead intersects the tail of $\hat{p}_{\mathbf{x}}$. Such solutions are discarded.

The formulation in Eqs. (4.4) - (4.5) can accommodate discontinuous changes of the minimiser, $b^\star = \arg\min_{b \in [-\alpha\sigma_{\mathbf{v}}, \alpha\sigma_{\mathbf{v}}]} \hat{I}(\mathbf{v}, b)$, as a result of changes in $\mathbf{v}$. It is thus less susceptible to convergence to local minima than a simple constrained optimisation formulation, as seen in Figure 4.3, which shows the hyperplane separators on the S4 dataset arising from this projection pursuit formulation with 50 random initialisations. By contrast to the constrained optimisation approach, projection pursuit converges to only a few solutions, all of which correspond to very high-quality cluster separators.

The projection index, $\phi(\mathbf{v})$, is a non-smooth non-convex locally Lipschitz continuous

function. Lewis and Overton (2013) have strongly advocated that a Broyden–Fletcher–Goldfarb–Shanno (BFGS) method using inexact line searches is very efficient for the minimisation of such functions, while being much less computationally demanding than non-smooth optimisation methods like gradient sampling (Burke et al., 2005). We call the projection pursuit algorithm that minimises the projection index $\phi(\mathbf{v})$, minimum density projection pursuit (MDP^2).

4.2.2 Divisive Hierarchical Clustering With Minimum Density Hyperplanes

To obtain a divisive hierarchical clustering algorithm capable of estimating the number of clusters, we need to specify when to terminate the successive bi-partitioning of subsets of $\mathcal{X}$ with MDP^2 (stopping rule). Let $\mathcal{X}_C \subset \mathcal{X}$ denote the observations assigned to cluster C, and $H(\mathbf{v}_C, b_C)$ be the MDH associated with this cluster. Furthermore, let $\hat{p}_{\mathbf{v}_C^\top \mathbf{x}}$ denote the density estimator constructed by projecting $\mathcal{X}_C$ onto $\mathbf{v}_C$. The *relative depth* criterion, defined in Eq. (4.6), measures the extent to which $H(\mathbf{v}_C, b_C)$ is a low-density separator of high-density clusters in $\mathcal{X}_C$. The relative depth is defined as the smaller of the relative differences in the density on the MDH, $\hat{p}_{\mathbf{v}_C^\top \mathbf{x}}(b_C)$, and the density of the two largest adjacent modes of the projected density,

$$\text{RelativeDepth}(\mathbf{v}_C, b_C; \mathcal{X}_C) = \frac{\min\left\{\hat{p}_{\mathbf{v}_C^\top \mathbf{x}}(m_l), \hat{p}_{\mathbf{v}_C^\top \mathbf{x}}(m_r)\right\} - \hat{p}_{\mathbf{v}_C^\top \mathbf{x}}(b_C)}{\hat{p}_{\mathbf{v}_C^\top \mathbf{x}}(b_C)}, \tag{4.6}$$

where m_l and m_r are the locations of the two largest modes of $\hat{p}_{\mathbf{v}_C^\top \mathbf{x}}$ to the left and right of b_C respectively. By convention, if there is no mode either to the left or the right the relative depth is zero. This criterion is equivalent to the inverse of a measure of cluster overlap for clustering with Gaussian mixtures (Aitnouri et al., 2000). The relative depth cannot be used directly as a stopping criterion because MDP^2 actively seeks projections for which the

Algorithm 1 Test of validity of MDH.

Require: Observations in cluster C, $\mathcal{X}_C \subset \mathcal{X}$, number of null samples of reference distribution m, critical quantile q, bandwidth multiplier b

```
n ← |X_C|
X_C^T = sample(X_C, ⌈n/2⌉)
X_C^H = X_C \ X_C^T
Apply MDP^2 on X_C^T to estimate MDH: H(v_C, b_C)
d ← RelativeDepth(v_C, b_C; X_C^H)
c ← 0
for i = 1 : m, do
    u ← {u_1, ..., u_{|X_C^H|}}, u_j ~ Uniform(0, 1)
    d' ← max_b{RelativeDepth(1, b; u)}
    if d' < d then
        c ← c + 1
    end if
end for
if c/m > q then
    return True
end if
```

density estimator $\hat{p}_{\mathbf{v}_C^\top \mathbf{x}}$ is multimodal. Thus, low-density hyperplanes that achieve non-zero relative depth can exist even if there are no high-density clusters to separate with respect to the true density. Moreover, the probability of identifying such hyperplanes increases as the sample size becomes smaller relative to the number of dimensions, which occurs as we move down the cluster hierarchy induced by a divisive algorithm.

We propose the following procedure to test the appropriateness of an MDH to separate high-density clusters. For a cluster C, we randomly split the data assigned to it, $\mathcal{X}_C$, into a training and a hold-out sample. We compute the MDH, $H(\mathbf{v}_C, b_C)$, using data from the training sample, while the relative depth of $H(\mathbf{v}_C, b_C)$ is estimated using data from the hold-out sample only. This estimate of the relative depth is then compared with a quantile of the distribution of the relative depth of a sample of equal size (to the hold-out sample) from a one-dimensional unimodal reference distribution. In our experiments we choose the uniform distribution as a reference as this is the standard choice in modality testing (Hartigan and Hartigan, 1985; Hartigan, 1977). If the relative depth of $H(\mathbf{v}_C, b_C)$ exceeds the chosen quantile of the relative depth of the reference distribution, we conclude that the MDH

Algorithm 2 Hierarchical Minimum Density Hyperplanes (MDH_{hier})

Require: Observations $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{n}$

Initialise with estimated number of clusters $\hat{k} = 1$ and vector of cluster assignments $\boldsymbol{\pi}$ with $\pi_i = 1 \forall i$.

repeat

For all current subsets of $\mathcal{X}$ assigned to clusters $\mathcal{X}_{C_j} = \{\mathbf{x}_i | \pi_i = j\}, j = 1, ..., \hat{k}$, locate the MDH and associated estimated projected density;

Test for multimodality in the estimated projected density of all $\mathcal{X}_{C_j}$ for $j = 1, \ldots, \hat{k}$;

Split all clusters for which the estimated projected density is multimodal resulting in new clusters $\mathcal{X}_{C_j}$;

Update current vector of cluster assignments $\pi_i = j$ iff $\mathbf{x}_i \in \mathcal{X}_{C_j}$ and $\hat{k} = \max \boldsymbol{\pi}$;

until The estimated projected density is not multimodal for all clusters.

return $\boldsymbol{\pi}, \hat{k}$

is a valid separator and C is split. This procedure is summarised in Algorithm 1.

To improve the accuracy of the separating hyperplane, the location of the split along the projection vector, b_C, is computed using the entire sample, $\mathcal{X}_C$. The steps for the our complete divisive algorithm, which we call MDH_{hier}, are summarised in Algorithm 2.

4.2.3 Ensemble Partitional Clustering With Minimum Density Hyperplanes

In Section 4.2.1 we compared the quality of MDHs obtained by optimising the constrained problem in Eq. (4.3), against the projection pursuit formulation, Eqs. (4.4) - (4.5). As Figure 4.1 illustrates, the former approach frequently converges to sub-optimal local minima. Nonetheless, using SQP to estimate MDHs is computationally less expensive because it doesn't involve the minimisation of $f(\mathbf{v}, b)$, at each function evaluation. If we consider MDHs obtained through SQP as *weak partitions* (Topchy et al., 2005) it is possible to combine them through ensemble clustering to obtain a complete clustering. We call this partitional algorithm MDH_{ens}. To produce a complete clustering into k clusters using an ensemble clustering of binary partitions, we use the probabilistic mixture model proposed in Topchy et al. (2005). This is done using a model-based clustering such that clusters correspond to components of a finite mixture model. The model for the vector of labels $\mathbf{y}_i \in \{0, 1\}^H$ assigned to the i-th observation by each of the H hyperplanes is a finite mix-

Algorithm 3 Ensemble Minimum Density Hyperplanes (MDH_{ens})

Require: Observations $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$, number of input hyperplanes H
  Initialise $n \times H$ matrix of input partitions $\mathbf{Y} = [y_{ij}]$ for $i = 1, \ldots, n$ and $j = 1, \ldots, H$.
  for $j = 1 : H$, do
    Sample initial projection vector $\mathbf{v}$ uniformly on the unit sphere $\mathbf{v} \in \mathbb{S}^{d-1}$, and initialise displacement from the origin $b = 0$;
    Locate the local minimum density hyperplane $H(\mathbf{v}^\star, b^\star)$ using SQP formulation given in Eq. (4.3);
    Store bi-partition from $H(\mathbf{v}^\star, b^\star)$ such that $y_{ij} = 0$ if $\mathbf{x}_i^\top \mathbf{v}^\star \leqslant b^\star$, $y_{ij} = 1$ if $\mathbf{x}_i^\top \mathbf{v}^\star > b^\star$;
  end for
  Combine the rowwise partitions in $\mathbf{Y}$ using the ensemble method of Topchy et al. (2005) with BIC to determine the final partition $\boldsymbol{\pi}$ and estimated number of clusters $\hat{k}$.
  return $\boldsymbol{\pi}, \hat{k}$

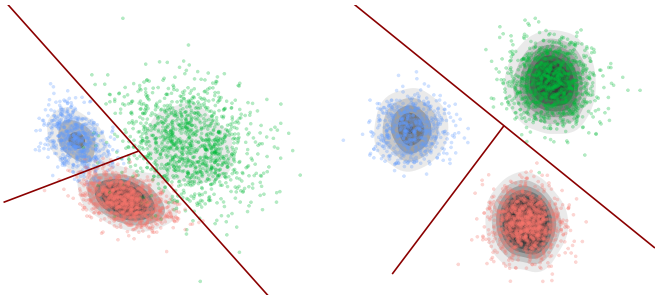
ture of Bernoulli distributions in the space of clusterings,

$$P(\mathbf{y}_i|\boldsymbol{\Theta}) = \sum_{l=1}^{k} \zeta_l \prod_{j=1}^{H} (\theta_l^{(j)})^{y_{ij}} (1 - \theta_l^{(j)})^{1-y_{ij}}, \tag{4.7}$$

where $\theta_l^{(j)}$ is the probability that $y_{ij} = 1$ if $\mathbf{y}_i$ is sampled from the l-th mixture component and $\boldsymbol{\zeta} = (\zeta_1, ..., \zeta_k)$ is the vector of mixing parameters such that $\sum_{l=1}^{k} \zeta_l = 1$. The parameter vector $\boldsymbol{\Theta} = \left(\zeta_1, \ldots, \zeta_k, \theta_1^{(1)}, \ldots, \theta_k^{(H)}\right)$ can be estimated through the expectation-maximisation (EM) algorithm assuming there exists an unobserved matrix of true cluster labels $\boldsymbol{Z} \in \{0, 1\}^{n \times k}$ whose expected value can be calculated from $\boldsymbol{\Theta}$. The row-wise maxima of $\mathbb{E}(\boldsymbol{Z})$ provide the clustering result. In this formulation, the number of clusters can be estimated by optimising a model selection criterion (McLachlan and Peel, 2000). We employ the Bayesian Information Criterion (BIC) (Fraley and Raftery, 2002). Although each of the individual separating hyperplanes used as input partitions can only separate convex clusters, after the ensemble clustering, MDH_{ens} can locate non-convex clusters. This algorithm is summarised in Algorithm 3.

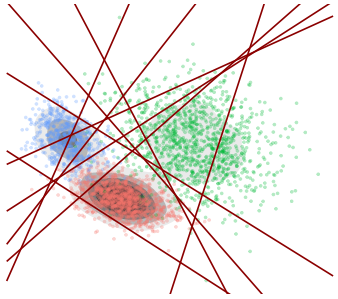
4.2.4 Visualisation of Proposed Methods

To visualise the clusters obtained by MDH_{hier} and MDH_{ens}, two-dimensional toy datasets were generated from three component Gaussian mixtures. Figures 4.4 and 4.5 show the

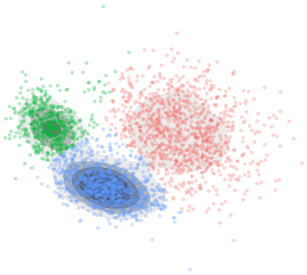


(a) (b)

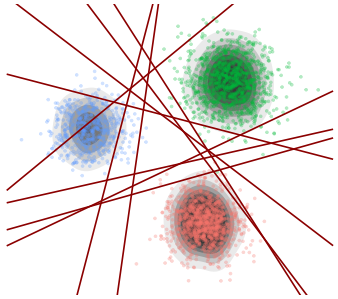
Figure 4.4: Clusters identified by divisive algorithm MDH_{hier}.



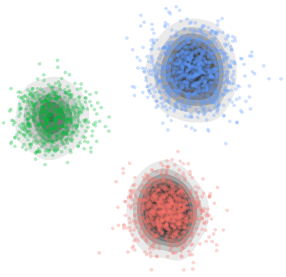
(a) Hyperplane separators for MDH_{ens}



(b) Cluster assignment



(c) Hyperplane separators for MDH_{ens}



(d) Cluster assignment

Figure 4.5: Clusters identified by partitional algorithm MDH_{ens}.

results of MDH_{hier} and MDH_{ens} respectively, applied to two of these datasets. The first dataset, Figures 4.4(a), 4.5(a) and 4.5(b), is characterised by high cluster overlap, while the second, Figures 4.4(b), 4.5(c) and 4.5(d), has very low cluster overlap. The low-density hyperplanes used to generate the clustering results are shown in red. Both methods correctly separate the high-density regions of the estimated density and achieve very low clustering error, even in the more difficult problem. Furthermore, the clusters identified are associated with the modes of the estimated density. Notice that MDH_{ens} identifies clusters effectively

despite the fact that a few hyperplane separators intersect regions of relatively high density (due to the local convergence problem).

4.3 Continuous Representations of Mixed Data

With the exception of DBSCAN (Ester et al., 1996) and its variants, which can identify high-density regions using only pairwise distances, all density-based clustering methods require continuous data to construct $\hat{p}_{\mathbf{x}}$ and identify high-density clusters. To apply these methods to data containing mixed feature types, it is therefore necessary to transform the data to obtain a continuous representation. We consider the application of three approaches for this, MDS (Borg and Groenen, 2005), mPPCA (Khan et al., 2010) and CSE (Roth et al., 2003). In this section, we briefly discuss these three methods, leaving a complete description, and an empirical evaluation of their appropriateness for density-based clustering to Chapter 3.

Both MDS and CSE require a matrix of pairwise distances, calculated using an appropriate distance metric for mixed data, such as the Gower distance (Gower, 1971). MDS aims to locate a continuous embedding which minimises the distortion between the original pairwise distances and the pairwise distances of the continuous representation. Following Azzalini and Menardi (2016) we apply non-metric MDS which minimises the SSTRESS criterion,

$$\sum_{i=1}^{n}\sum_{j=1}^{n}(f(D_{ij})^2 - d_{ij}^2)^2,$$

where D_{ij} and d_{ij} are the pairwise distances between observations i and j in the original data and continuous representation respectively, and $f(\cdot)$ is a monotonic transformation of the input distances, which is optimised during the iterative procedure. In all our experiments we use the default choice of $f(\cdot)$ in the `MASS` package for `R`. This is intuitive, although this objective is not directly related to clustering. CSE (Roth et al., 2003) explicitly considers the

ability to identify clusters in the continuous representation using the k-means algorithm. In this approach, k-means clustering on the continuous representation is guaranteed to produce the same partition as minimising the sum of within cluster pairwise distances using the original dissimilarity matrix.

mPPCA takes a different approach, assuming a Gaussian latent variable, $\mathbf{z}$, has given rise to the mixed variable $\mathbf{x}$. For the continuous dimensions of $\mathbf{x}$, this model takes the standard conjugate Gaussian form. For the discrete dimensions of $\mathbf{x}$, a multinomial distribution is assumed, whose input vector of probabilities is related to $\mathbf{z}$ via the softmax (multinomial logistic regression) link function. The distribution of $\mathbf{z}$ conditional on $\mathbf{x}$ then gives the continuous representation. The model for the discrete dimensions of $\mathbf{x}$ prevents a closed form solution for this conditional distribution. To solve this, Khan et al. (2010) propose a variational EM algorithm. Through extensive experimentation, we found that this is sensitive to initialisation, and convergence to local solutions can critically affect the continuous representation produced.

In Chapter 3 we investigate the clustering performance of MDH_{hier}, MDH_{ens}, dePDDP, k-means++ and Gaussian mixture model-based (GMM) clustering on the continuous representations produced by MDS, mPPCA and CSE. We conclude that for data simulated via the model used in the simulation study of this chapter, and the real datasets considered, CSE produced the most appropriate continuous representation. Therefore, where a continuous representation is required, the results based on this representation are reported in Sections 4.4.3 and 4.4.4.

4.4 Experimental Results

In this section, we evaluate the performance of the density-based clustering methods we propose, MDH_{hier} and MDH_{ens} across simulated and real datasets containing both continuous and mixed attributes, with varying characteristics. The proposed approaches are compared to well-established and state-of-the art clustering methods. The methods considered are:

1. Normalised spectral clustering (Ng et al., 2002) using the local bandwidth selection rule and cluster estimation method of Zelnik-Manor and Perona (2004).

2. k-means++ (Arthur and Vassilvitskii, 2007), a recent variant of the classical k-means algorithm that through appropriate initialisation is guaranteed to be $\mathcal{O}(\log k)$ competitive with the optimal k-means clustering. We use the Gap statistic (Tibshirani et al., 2001) to estimate the number of clusters. This approach to estimate the number of clusters is computationally expensive, and therefore significantly increases the computational time required compared to k-means with a pre-specified number of clusters.

3. DBSCAN (Ester et al., 1996) and its subspace clustering extension SubClu (Kailing et al., 2004). DBSCAN is arguably the most widely used density-based clustering algorithm. We use the implementation in the `R` package `dbscan`. DBSCAN has been documented to perform poorly in high-dimensional applications (Agrawal et al., 1998), and so we also considered the SubClu algorithm as a subspace variant (implemented in the `R` package `subspace`). This algorithm failed to produce meaningful partitions in any of the datasets considered due to very poor estimation of the number of clusters, and so its performance is not reported.

4. pdfCluster (Menardi and Azzalini, 2014), and its extension for mixed data (Azzalini and Menardi, 2016). This is a recently proposed density-based clustering algorithm that employs a Gaussian kernel density estimator to identify high-density clusters in the full-dimensional space. pdfCluster is limited to datasets with small numbers of observations and low dimensionality due to the computational cost and numerical instability of constructing the estimated density. For mixed datasets Azzalini and Menardi (2016) recommend using non-metric MDS to obtain a low-dimensional continuous representation, before applying pdfCluster. We use the implementation given in the `R` package `pdfCluster`.

5. Density-enhanced Principal Direction Divisive Clustering (dePDDP) (Tasoulis et al., 2010). dePDDP is a divisive projective clustering algorithm that is related to MDH_{hier}. It recursively bi-partitions the data by projecting onto the first principal component;

constructing a one-dimensional kernel density estimator from the projections; and splitting at the point that minimises this estimator within the interval between the first and last mode. If the projected density is unimodal the current cluster is not further subdivided. At each level of the hierarchy dePDDP bi-partitions the data according to hyperplane that achieves the lowest possible density out of the hyperplanes with normal vector equal to the first principal component. Comparing against this algorithm therefore highlights the impact of optimising the orientation of the separating hyperplane in MDH_{hier}.

6. Gaussian mixture model (GMM) using BIC (Fraley and Raftery, 2002) to estimate the number of clusters. We use the implementation in the `R` Package `MClust`.

4.4.1 Details of Implementation

For all algorithms, we use parameter settings recommended in the literature. For DBSCAN and SubClu, we apply the approach proposed by Ester et al. (1996) to determine ϵ and MinPts which define the neighbourhood radius, and the minimum number of points required for a point to be considered a high-density (core) point respectively. The only tuning parameter in pdfCluster is the covariance matrix employed by the kernel density estimator. The recommendation in Azzalini and Menardi (2014) is to use a diagonal covariance matrix, with $\Sigma_{ii} = 0.75\hat{\sigma}_i[4/(n(d+2))]^{1/(d+4)}$, where $\hat{\sigma}_i$ is the estimated standard deviation along the i-th dimension.

For spectral clustering, we use the normalised graph cut algorithm of Ng et al. (2002), which employs a fully connected graph. The adjacency matrix $\boldsymbol{W}$ is computed through the Gaussian kernel, $W_{ij} = \exp\left(-D_{ij}/s_i s_j\right)$, where D_{ij} is the distance between the i-th and j-th observation, and s_i (s_j) denotes the distance of the i-th (j-th) observation to its seventh nearest neighbour. This local scaling approach has been proposed by Zelnik-Manor and Perona (2004) to handle multi-scale data, and in our experience is very effective. The choice of the seventh nearest neighbour is arbitrary, but this the value recommended

by Zelnik-Manor and Perona (2004). This seems to work well in practice but the choice of this value is considered further in Section 8.2.1. To set the bandwidth of the kernel density estimator employed by dePDDP, Tasoulis et al. (2010) recommend the standard rule, $h = \hat{\sigma}_{\mathrm{pc}_1} (4/(3n))^{1/5}$, where $\hat{\sigma}_{\mathrm{pc}_1}$, is the estimated standard deviation of the projections on the first principal component.

The two most important parameters for the MDH-based algorithms are the initial projection direction, and α, the parameter that determines the range of the interval over which the density is being minimised. Following Pavlidis et al. (2016) we initialise each stage in $\mathrm{MDH}_{\mathrm{hier}}$ using both the first and second principal components. We then select the hyperplane which leads to the larger relative depth in the test sample. This relative depth is then compared with the 0.975 estimated quantile of the relative depth of a sample from the uniform distribution for our stopping rule proposed in Section 4.2.2. Our experience with the method has shown that data containing multiple density separable clusters tend to show strong multimodal signal along the optimal projection, whereas if this is not the case then the conservatism of the uniform reference distribution is effective in mitigating against substantial over partitioning. We found that all quantiles above 0.9 yield similar results in most cases. The parameter α is initialised close to zero and progressively increased to $\alpha_{\mathrm{max}} = 1$. As discussed in Pavlidis et al. (2016), using initially a small α steers the algorithm towards projection directions that exhibit a strong bi-modal structure and induce a balanced data partition. Increasing α subsequently enables the method to converge to the minimiser of the projected density. For the partitional algorithm, $\mathrm{MDH}_{\mathrm{ens}}$, a diverse set of separating hyperplanes is necessary to obtain a high-quality clustering. To this end, both the initial projection direction and α are initialised uniformly at random. In total 30 binary partitions are provided as inputs to the consensus clustering algorithm. In all MDH-based algorithms we

use $h = 0.9\hat{\sigma}n^{-1/5}$, which (Silverman, 1986) recommend for bandwidth selection when the univariate density being estimated is assumed to be multimodal. To maintain a fixed bandwidth regardless of the choice of projection vector, we take $\hat{\sigma} = \hat{\sigma}_{\text{pc}_1}$.

For mixed datasets, pairwise distances are computed using the Gower distance (Gower, 1971). For DBSCAN and spectral clustering, the dissimilarity matrix is sufficient while for the MDH variants, dePDDP, k-means++, pdfCluster and GMM, a continuous representation is necessary. Since Azzalini and Menardi (2016) have already proposed the use of non-metric MDS to produce a continuous representation of no more than five dimensions, we employ this for pdfCluster. For MDH_{hier}, MDH_{ens}, k-means++ and GMM, we use the continuous representation from CSE, since this produced the most consistently competitive clustering performance for the datasets considered. For a comprehensive evaluation of the continuous representations considered, see Chapter 3.

4.4.2 Measuring Clustering Performance

We evaluated the performance of all competing algorithms using different performance measures that are appropriate for comparing clusterings with potentially different numbers of clusters, such as normalised mutual information (NMI) (Strehl and Ghosh, 2002), Rand Index (Rand, 1971), Adjusted Rand Index (Hubert and Arabie, 1985) and V-measure (Rosenberg and Hirschberg, 2007). The choice of performance measure did not alter the relative performance of the different algorithms, and we thus report performance with respect to NMI only. NMI is an information theoretic measure that quantifies the statistical information shared between two distributions. Given a clustering $\boldsymbol{\pi}$ of n observations into k assigned clusters and the true cluster assignment $\boldsymbol{\pi}^\star$ with $k^\star$ true clusters, let n_i^{π} be the number of observations in assigned cluster i, and $n_j^{\pi^\star}$ be the number of observations in true cluster j. Further, let $n_{i,j}$ be the number of observations from true cluster j in assigned

cluster i. NMI is defined as,

$$\mathrm{NMI}(\boldsymbol{\pi}, \boldsymbol{\pi}^\star) = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k^\star} n_{i,j} \log\left(\frac{n_{i,j} n}{n_i^{\pi} n_j^{\pi^\star}}\right)}{\sqrt{\left(\sum_{i=1}^{k} n_i^{\pi} \log\frac{n_i^{\pi}}{n}\right)\left(\sum_{j=1}^{k^\star} n_j^{\pi^\star} \log\frac{n_j^{\pi^\star}}{n}\right)}}.$$

The value of NMI is in the range $[0, 1]$ with higher values indicating better performance.

4.4.3 Simulated Data

Here we evaluate the performance of $\mathrm{MDH}_{\mathrm{hier}}$ and $\mathrm{MDH}_{\mathrm{ens}}$ across simulated continuous and mixed data with varying numbers of dimensions and clusters. These simulations allow us to control the level of difficulty of the clustering problem. In all cases the distribution represents a mixture in which each of the k components constitutes a cluster. For each dimensionality and number of clusters, 30 data sets were generated, each originating from a probability distribution with randomly selected parameters. The mixing proportions were generated as

$$\zeta_i = \frac{u_i}{\sum_{j=1}^{k} u_j},$$

where

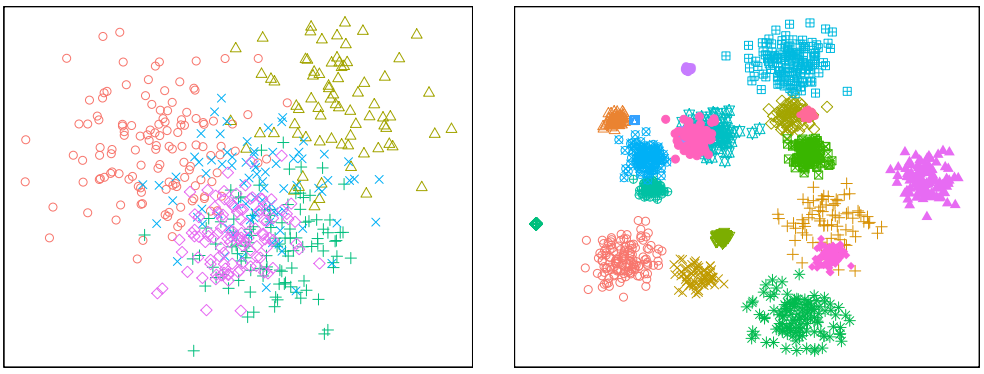
$$u_i \sim \mathrm{Uniform}[1, 2],\ i = 1, \ldots, k$$

and the parameters for each of the components were generated randomly as follows,

$$\boldsymbol{\mu}^C \sim \mathrm{Uniform}[0, k/3]^{d_C};$$

$$\mu_j^D \sim \mathrm{Bern}(0.5),\ j = 1, \ldots, d_D;$$

$$\sigma = u^2, u \sim \mathrm{Uniform}[0.1, 1.1].$$



(a) $d = 10, k = 5$ (b) $d = 40, k = 20$

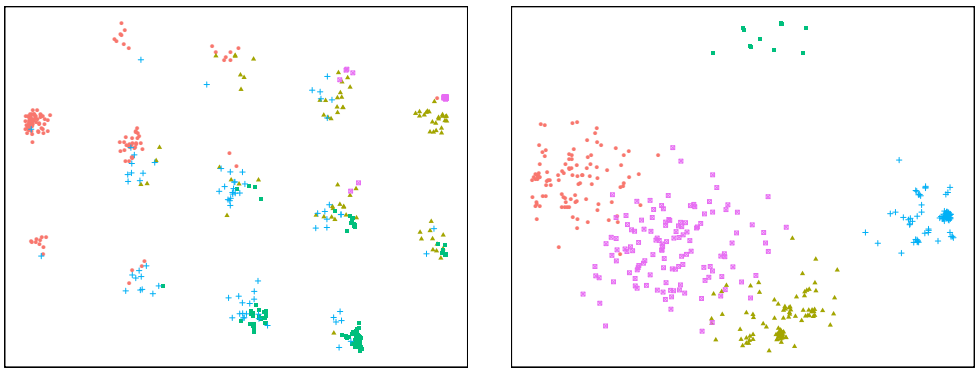
Figure 4.6: Example structure in continuous simulated data produced by projecting onto the first two principal components

From each component $\lceil 100k\zeta_i \rceil$ data were generated according to,

$$\mathbf{x}^C \sim N(\boldsymbol{\mu}^C, \sigma\mathbf{I}),$$

$$\mathbb{P}(x_j^D = B) = \begin{cases} 1 - \sigma/4, & B = \mu_j^D \\ \sigma/4, & B = 1 - \mu_j^D \end{cases}.$$

The model for the continuous data tends to induce greater separability between clusters in datasets with higher numbers of clusters and higher dimensionality. Figure 4.6 shows two-dimensional principal component projections of typical examples of the two most extreme cases for the numbers of clusters and dimensionality of the continuous datasets generated in our experiments. For datasets like the one depicted in Figure 4.6(a), the high degree of overlap in the true clusters means that the density-based definition may not be appropriate for distinguishing all clusters. Hence, we expect methods relying on this approach to find these data challenging. In contrast, the density-based cluster definition is appropriate for datasets like the one in Figure 4.6(b). The dimensionality of these datasets can pose a challenge for pdfCluster and DBSCAN, but the projective density algorithms are expected to perform well.



(a) $d_C/d_D = 5/5$ (b) $d_C/d_D = 20/20$

Figure 4.7: Example structure in continuous representation of simulated mixed data produced using CSE

The Gower distance function, used for computing the pairwise distances of the mixed data, is given by the sum of the normalised dissimilarities in each dimension. Therefore each entry in the dissimilarity matrix, $\mathbf{D}$, can be thought of as a convolution of a discrete random variable with a continuous one. Our understanding of these convolutions is informed by kernel density estimation, where the number of atoms in the discrete distribution and their separation relative to the variability in the continuous dimensions are the main determining factors in the cluster structure of the convolution. For density-based clustering, we require the continuous representation to exhibit a unimodal structure for each cluster. We expect this to be possible with a higher number of discrete dimensions, inducing more atoms in the distribution of $\mathbf{x}^D$, and moderate variability in the continuous component of each cluster. Typical examples of the structure within the mixed data are given in Figure 4.7, which provides the two-dimensional CSE representation of datasets generated with different numbers of dimensions, each with five clusters. In cases like the one depicted in Figure 4.7(a), there are insufficient discrete dimensions, resulting in very high probability density around the atoms of the distribution of $\mathbf{x}^D$ relative to the variability in $\mathbf{x}^C$. Thus, the continuous representation has multiple dense regions for each cluster. Further, these dense regions

Table 4.1: Clustering performance on simulated continuous datasets. The top row of each cell of the table reports NMI and the second the estimated number of clusters. Each cell reports mean performance over 30 experiments.

		$k = 5$			$k = 10$			$k = 20$		
	d	10	20	40	10	20	40	10	20	40
MDH_{hier}	NMI	0.353	0.645	0.869	0.870	0.973	0.997	0.985	0.999	1.000
	k	3.1	3.5	4.6	8.6	9.9	10	20	20	20
MDH_{ens}	NMI	0.663	0.776	0.919	0.875	0.975	0.998	0.973	0.997	1.000
	k	6.7	6.0	5.7	10.5	10.0	10.0	19.8	19.8	19.2
dePDDP	NMI	0.620	0.774	0.921	0.821	0.937	0.974	0.961	0.981	0.985
	k	17.5	13.8	9.7	26.5	20.2	15.1	40.2	31.2	27.7
k-means++ (Gap)	NMI	0.705	0.784	0.859	0.868	0.916	0.914	0.929	0.930	0.925
	k	8.3	8.6	8.1	16.8	16.9	15.5	31.9	32.7	31.2
$\text{Spectral}_{\text{auto}}$	NMI	0.832	0.812	0.804	0.713	0.710	0.699	0.661	0.635	0.656
	k	3.6	3.6	6.5	4.8	4.9	4.9	8.1	7.5	7.7
DBSCAN	NMI	0.301	0.374	0.349	0.659	0.692	0.658	0.787	0.770	0.767
	k	2.2	2.5	2.3	7.1	7.1	6.5	16	14	13
pdfCluster	NMI	0.414	0.407	0.368	0.661	0.619	0.631	0.677	0.648	0.805
	k	2.2	2.1	1.8	5.0	4.6	4.5	11.1	10.7	11.9
GMM	NMI	0.891	0.958	0.876	0.951	0.970	0.970	0.733	0.724	0.719
	k	4.8	4.9	3.9	9.0	9.0	9.0	9.0	9.0	9.0

do not contain observations originating from a single true cluster. Hence, we expect algorithms which rely on this representation to perform poorly and to drastically overestimate the number of clusters. Increasing the dimensionality (Figure 4.7(b)) permits a continuous representation where associating the modes of the estimated density with a single true cluster is more appropriate. Here, projective density-based methods should perform well.

The clustering results for the continuous and mixed data are summarised in Tables 4.1 and 4.2, respectively. The tables report average performance with respect to NMI, as well as the average number of clusters found. The best performing algorithm in each case is indicated in red. Our experience indicates that the variability in performance arising from randomness in the sampling distribution giving rise to the data, completely dominates the randomness induced by the non-deterministic nature of MDH_{hier}. We therefore only run MDH_{hier} once for each of the 30 replications of each scenario. For the higher-dimensional continuous data with more clusters, MDH_{hier} and MDH_{ens} perform the best since the clus-

ters located are associated with the modes of the estimated density. For these datasets, dePDDP and k-means++ also perform competitively while spectral clustering, DBSCAN and pdfCluster produce lower quality partitions. GMM also performs well for the datasets with 5 and 10 clusters but not for the datasets with 20 clusters, where BIC penalises too heavily for fitting a more complex model, leading to an underestimation of the number of clusters. For the lower-dimensional continuous data with fewer clusters, the non-parametric density-based definition is not appropriate due to high cluster overlap so the MDH variants, dePDDP, pdfCluster and DBSCAN find these datasets challenging. In these situations, GMM tends to produce the highest quality partitions. Increasing the dimensionality induces greater separation between the true clusters so the projective density-based approaches such as dePDDP, MDH_{hier} and MDH_{ens} perform well (better than k-means++, GMM and spectral clustering) for the 40-dimensional datasets. However, DBSCAN and pdfCluster still perform poorly as the estimated density is unreliable in dimensions as high as this.

For the higher-dimensional mixed data, the clusters are associated with unimodal high-density regions in the continuous representation. In these examples, MDH_{hier} and MDH_{ens} produce high-quality partitions. Similarly, dePDDP and k-means++ perform well on these datasets. When the dimensionality is lower, the discrete attributes induce modes in the estimated density of the continuous representation around the atoms of the distribution of $\mathbf{x}^D$. This inhibits the accurate estimation of the number of clusters, and causes relatively poor performance by all algorithms. dePDDP provides the best NMI scores for the low-dimensional data although it locates substantially more clusters than the other algorithms.

Table 4.2: Clustering performance on simulated mixed datasets. The top row of each cell of the table reports NMI and the second the estimated number of clusters. Each cell reports mean performance over 30 experiments.

		$k = 5$			$k = 10$			$k = 20$		
	d_C/d_D	5/5	10/10	20/20	5/5	10/10	20/20	5/5	10/10	20/20
MDH_{hier}	NMI	0.586	0.806	0.949	0.621	0.739	0.964	0.639	0.699	0.935
	k	10.8	5.9	4.8	20.7	13.6	10.0	38.3	26.9	20.6
MDH_{ens}	NMI	0.570	0.835	0.936	0.517	0.720	0.911	0.413	0.576	0.777
	k	8.6	5.7	5.0	15.4	10.9	9.5	24.6	20.7	17.8
dePDDP	NMI	0.639	0.729	0.811	0.698	0.704	0.818	0.748	0.656	0.752
	k	34.2	35.4	35.8	64.8	71.3	70.6	142.9	118.3	141.2
k-means++ (Gap)	NMI	0.621	0.811	0.889	0.641	0.816	0.925	0.649	0.830	0.928
	k	9.8	9.0	7.5	19.5	19.4	17.3	39.2	39.4	38.3
$\text{Spectral}_{\text{auto}}$	NMI	0.489	0.611	0.637	0.490	0.399	0.479	0.484	0.431	0.365
	k	3.1	2.6	2.6	3.4	2.5	2.9	5.5	4.7	3.7
DBSCAN	NMI	0.493	0.498	0.513	0.613	0.626	0.667	0.736	0.733	0.767
	k	12.4	5.7	3.6	23.1	15.5	7.8	56.6	33.5	16.5
pdfCluster	NMI	0.550	0.711	0.867	0.413	0.457	0.622	0.425	0.208	0.371
	k	4.7	4.3	4.6	6.9	5.6	6.9	8.3	6.4	8.4
GMM	NMI	0.613	0.632	0.629	0.674	0.675	0.660	0.603	0.555	0.475
	k	7.2	5.0	6.9	9.0	8.7	7.4	9.0	9.0	8.5

4.4.4 Real Data

We now consider the performance of our proposed methods on benchmark datasets from the UCI machine learning repository (Lichman, 2013). The main properties of the datasets are summarised in Table 4.3.

Table 4.3: Main characteristics of UCI datasets considered.

Dataset	n	d_C	d_D	k
Image Segmentation	2309	19	-	7
Isolet	7797	617	-	26
Multi. Digits	2000	216	-	10
Opt. Digits	5620	64	-	10
Pen Digits	10992	16	-	10
Satellite	6435	36	-	6
Smartphone	10929	561	-	12
Autodata	392	5	2	5
Credit Approval	690	6	9	2
Dermatology	366	1	33	6
Heart Disease	294	5	8	5
Soy Bean	682	7	28	19
Voters	435	-	16	2

Table 4.4: Clustering performance on continuous real datasets. The top row of each cell of the table reports NMI and the second the estimated number of clusters (when applicable). For the non-deterministic MDH_{hier} the mean performance over 30 runs is given.

		O. Dig.	P. Dig.	Isolet	Smart.	Im. Seg.	Sat.	M. Dig.
MDH_{hier}	NMI	0.753	0.792	0.746	0.701	0.620	0.638	0.739
	k	12.0	18.2	25.5	3.0	11.1	4.1	11.1
MDH_{ens}	NMI	0.708	0.660	0.650	0.565	0.641	0.568	0.583
	k	20	20	10	3	7	7	7
dePDDP	NMI	0.000	0.625	0.402	0.565	0.593	0.606	0.610
	k	3	41	4	6	38	46	10
k-means (gap)	NMI	0.719	0.735	0.698	0.545	0.568	0.589	0.703
	k	19	20	50	24	14	11	20
$\text{Spectral}_{\text{auto}}$	NMI	0.728	0.378	0.637	0.574	0.415	0.393	0.724
	k	9	2	15	2	3	2	9
DBSCAN	NMI	0.509	0.018	0.000	0.117	0.122	0	0.017
	k	10	3	1	10	8	1	2
GMM	NMI	0.627	0.727	0.395	0.000	0.617	0.546	0.000
	k	9	9	2	1	8	9	1

We first discuss performance on the continuous datasets. Table 4.4 reports the performance of all the algorithms considered except pdfCluster, which was not able to run successfully due to the number of observation in these datasets. As before, we report the values of NMI and the estimated number of clusters for each algorithm with the best NMI for each dataset indicated in red. All of these values originate from a single run of each algorithm, except for the non-deterministic algorithm MDH_{hier} where the mean performance over 30 runs is reported. Although both MDH_{hier} and MDH_{ens} have an element of randomness in the determination of the final clustering, the variability in performance was very low for both algorithms. Further, the NMI computed between partitions resulting from different runs of these algorithms was very high (approximately 0.95 for MDH_{hier} and 0.9 for MDH_{ens}). In all cases the best performance is exhibited by one of the MDH-based algorithms. It is also clear that the divisive algorithm, MDH_{hier}, performs better than the partitional algorithm, MDH_{ens} on these datasets. This is not unexpected as partitions using MDP^2 aim to identify hyperplanes that do not split any clusters and separate at least one cluster from the rest of the data in successive subsets of $\mathcal{X}$. Clusters which are difficult (or impossible) to separate effectively when all the data are considered, can become easier

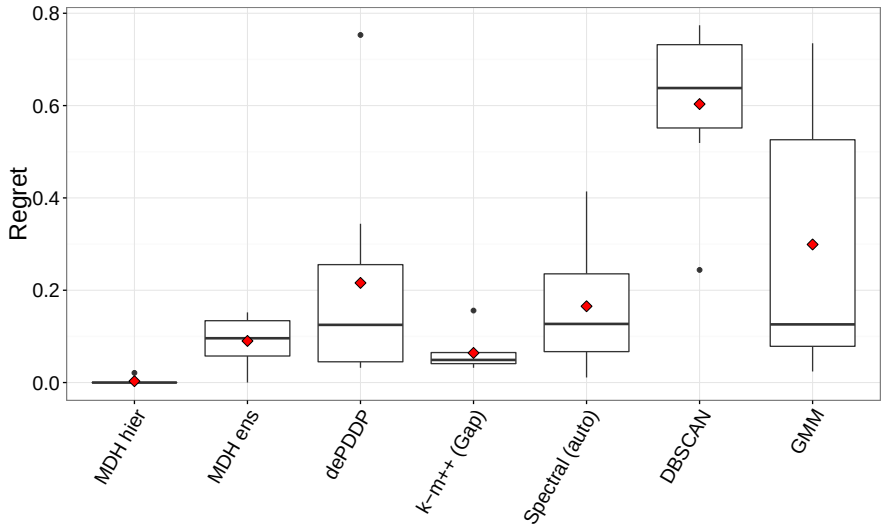
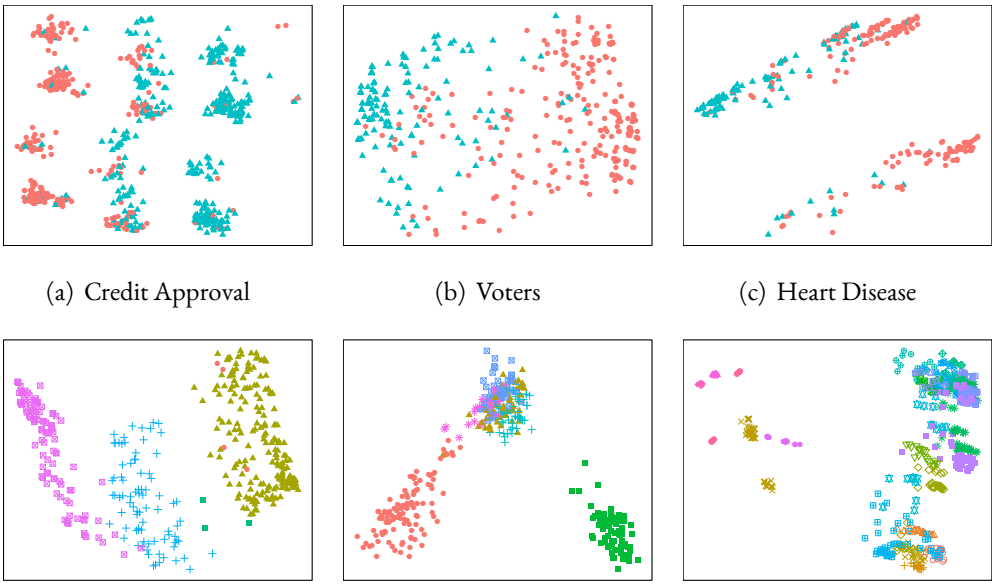


Figure 4.8: Box plot of regret based on the NMI over continuous real datasets

to separate when observations from other clusters are removed from the dataset. A divisive procedure can exploit this fact. Nonetheless MDH_{ens} always outperforms DBSCAN while outperforming dePDDP, GMM and spectral clustering in the majority of cases. Of the alternative density-based clustering methods dePDDP and GMM perform best, while DBSCAN exhibits relatively poor performance on all datasets. This poor performance is attributable to the difficulty of identifying high-density clusters in high dimensions. In general, the MDH variants determine the number of clusters relatively accurately, with large over or underestimation being rare. This is not the case for the other algorithms, with dePDDP often dramatically overestimating due to the separation of outliers in the tails of the estimated projected density, or underestimating due to the lack of multimodality in the projected density along the first principal component. The Gap statistic overestimates the number of clusters in all cases, while self-tuning spectral clustering, GMM and DBSCAN tend to underestimate this in general.

To assess the relative performance of each algorithm across all the continuous datasets, Figure 4.8 provides boxplots of regret with respect to NMI. The regret of an algorithm for a given dataset is defined as the difference between the performance of the best algorithm for



(a) Credit Approval (b) Voters (c) Heart Disease

(d) Autodata (e) Dermatology (f) Soy Bean

Figure 4.9: Two-dimensional visualisation of mixed real datasets after the application of CSE

this dataset and the performance of the algorithm in question,

$$\text{Regret}(\mathcal{A}) = \text{NMI}(\boldsymbol{\pi}_{\mathcal{A}^\star}, \boldsymbol{\pi}^\star) - \text{NMI}(\boldsymbol{\pi}_{\mathcal{A}}, \boldsymbol{\pi}^\star) \tag{4.8}$$

where $\mathcal{A}$ is the algorithm in question, $\boldsymbol{\pi}_{\mathcal{A}}$ is the partition induced by this algorithm and $\boldsymbol{\pi}_{\mathcal{A}^\star}$ is the partition induced by the highest performing algorithm. Therefore, a regret of zero indicates that the algorithm performs best on a dataset, while higher values indicate worse relative performance. As the figure shows MDH_{hier} achieves a median regret very close to zero, so has the best relative performance, followed by *k*-means++. Both MDH-based methods outperform the other density-based clustering methods, and dePDDP performs similarly to spectral clustering and GMM, all of which outperform DBSCAN.

We next discuss the clustering of the mixed real datasets. Figure 4.9 provides a 2-dimensional CSE visualisation of these datasets, which indicates that most of these datasets have clustering structures that are challenging for all algorithms considered. Table 4.5 reports the

Table 4.5: Clustering performance on mixed real datasets. In each cell of the table the first row reports NMI and the second the estimated number of clusters (when applicable). For the non-deterministic MDH_{hier} the mean performance over 30 runs is given.

		Credit	Voters	Heart	Auto	Derm	Soybean
MDH_{hier}	NMI	0.241	0.337	0.239	0.778	0.909	0.705
	k	17.6	4.7	5.7	5.1	4.9	15.3
MDH_{ens}	NMI	0.287	0.492	0.263	0.908	0.843	0.658
	k	9	3	7	3	5	8
dePDDP	NMI	0.258	0.395	0.225	0.674	0.860	0.727
	k	22	3	12	8	8	45
k-means++ (gap)	NMI	0.349	0.433	0.243	0.637	0.734	0.742
	k	4	4	4	10	12	38
Spectral_{auto}	NMI	0.258	0.103	0.250	0.902	0.734	0.451
	k	9	3	3	3	3	5
DBSCAN	NMI	0.012	0.000	0.000	0.739	0.000	0.749
	k	2	1	1	3	1	19
pdfCluster	NMI	0.222	0.385	0.265	0.896	0.835	0.536
	k	13	3	3	5	4	5
GMM	NMI	0.015	0.353	0.539	0.631	0.700	0.000
	k	2	8	6	9	3	2

performance of the competing clustering algorithms on the mixed datasets. On the mixed datasets, no algorithm has consistently superior performance. All algorithms appear to perform poorly on the credit approval dataset. An explanation for the low NMI scores is that although the dataset contains only two true clusters, Figure 4.9(a) indicates that in the CSE representation the number of dense compact regions is much larger, causing the number of clusters to be overestimated. Note that this structure does not appear to be an artefact of the continuous representation as the self-tuning spectral clustering algorithm (which uses the original pairwise distances) also overestimates the number of clusters to be nine. If the penalty for overestimation of the number of clusters is removed, the performance of all the algorithms improves significantly for this dataset, with clustering accuracy (purity) of around 0.8 in almost all cases.

On the Voters dataset MDH_{ens} performs best followed by k-means++, while spectral clustering and and DBSCAN perform poorly on this dataset. On the Heart Disease dataset

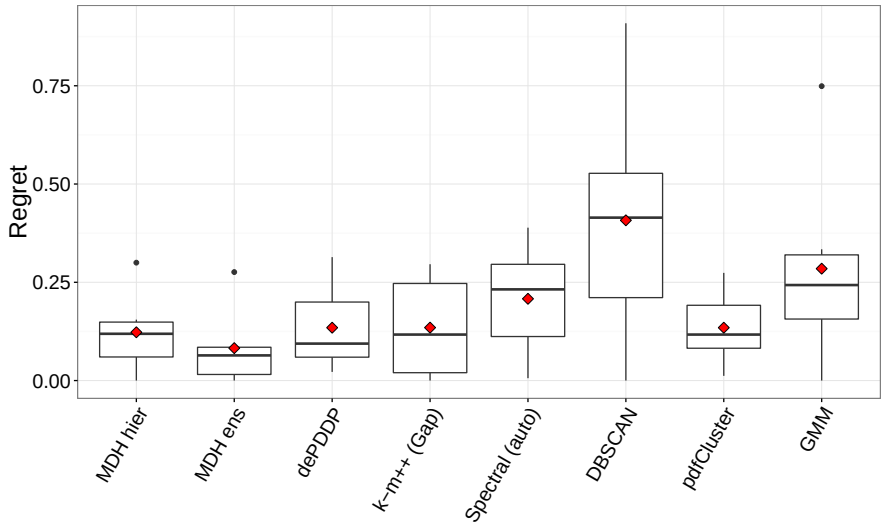


Figure 4.10: Box plot of regret with respect to NMI over mixed real datasets

GMM performs best, significantly outperforming the other algorithms. On this dataset, all other algorithms which use the CSE representation perform similarly, while spectral clustering and DBSCAN produce lower quality partitions. MDH_{ens} has the best performance on Autodata, followed by spectral clustering and pdfCluster. An inspection of Figure 4.9(d) reveals why MDH_{ens} is very effective on this dataset, with the three main clusters being separable by low-density regions. In contrast the divisive algorithm MDH_{hier} is more suitable for the Dermatology dataset. As Figure 4.9(e) shows in this dataset it is possible to effectively separate two clusters from the rest of the data, but the remaining clusters are much less separable from one another when the entire dataset is considered. This structure is apparent along the directions of high variability, also explaining the good performance of dePDDP on this dataset. Once these groups are removed, the projection of the remaining points can reveal the additional clusters, highlighting the potential advantages of the divisive approach. The worst performance on Dermatology is exhibited by DBSCAN. Finally, on the SoyBean dataset DBSCAN achieves the highest NMI closely followed by *k*-means++, dePDDP and MDH_{hier}. MDH_{ens} is also competitive on this dataset while spectral clustering and GMM perform poorly.

It is important to note that on the majority of datasets the best performing methods used a continuous representation of the data rather than the original pairwise distances. The performance of MDH_{hier} and MDH_{ens} are, in most cases, competitive with the best performing methods. This is clearly seen in Figure 4.10 which depicts boxplots of regret with respect to the NMI measure on the mixed datasets. Overall the cluster structures in these datasets are more favourable to the partitional algorithm MDH_{ens} than the divisive algorithm, MDH_{hier}. MDH_{ens} achieves the lowest median regret and its regret exhibits very little variability.

4.4.5 Image Segmentation

Finally, we assess the performance of MDH_{hier} and MDH_{ens} for the task of image segmentation. Fig. 4.11 shows the segmentation of six images by the considered algorithms. Each image contains approximately 40,000 pixels, and segmentation was performed by clustering the R,G,B values representing each pixel. In the reconstructed images of Fig. 4.11 the colour of each pixel is determined by the average R,G,B values of the pixels assigned to the same cluster.

The size of these datasets was too large for spectral clustering, DBSCAN and pdfCluster, so we used a pre-processing step in which each dataset was summarised with 5000 micro-clusters (obtained through *k*-means++). For all algorithms, the number of segments was determined automatically, however, DBSCAN failed to segment any of the images so the results are omitted. MDH_{hier} produces very high-quality segmentations, with an accurate representation of the true colours, a sensible identification of boundaries, and relatively few segments. MDH_{ens} also produced good results, although this approach locates more clusters than MDH_{hier}, with the exception of the second and sixth images.

pdfCluster and spectral clustering also produce high quality segmentations (with the ex-

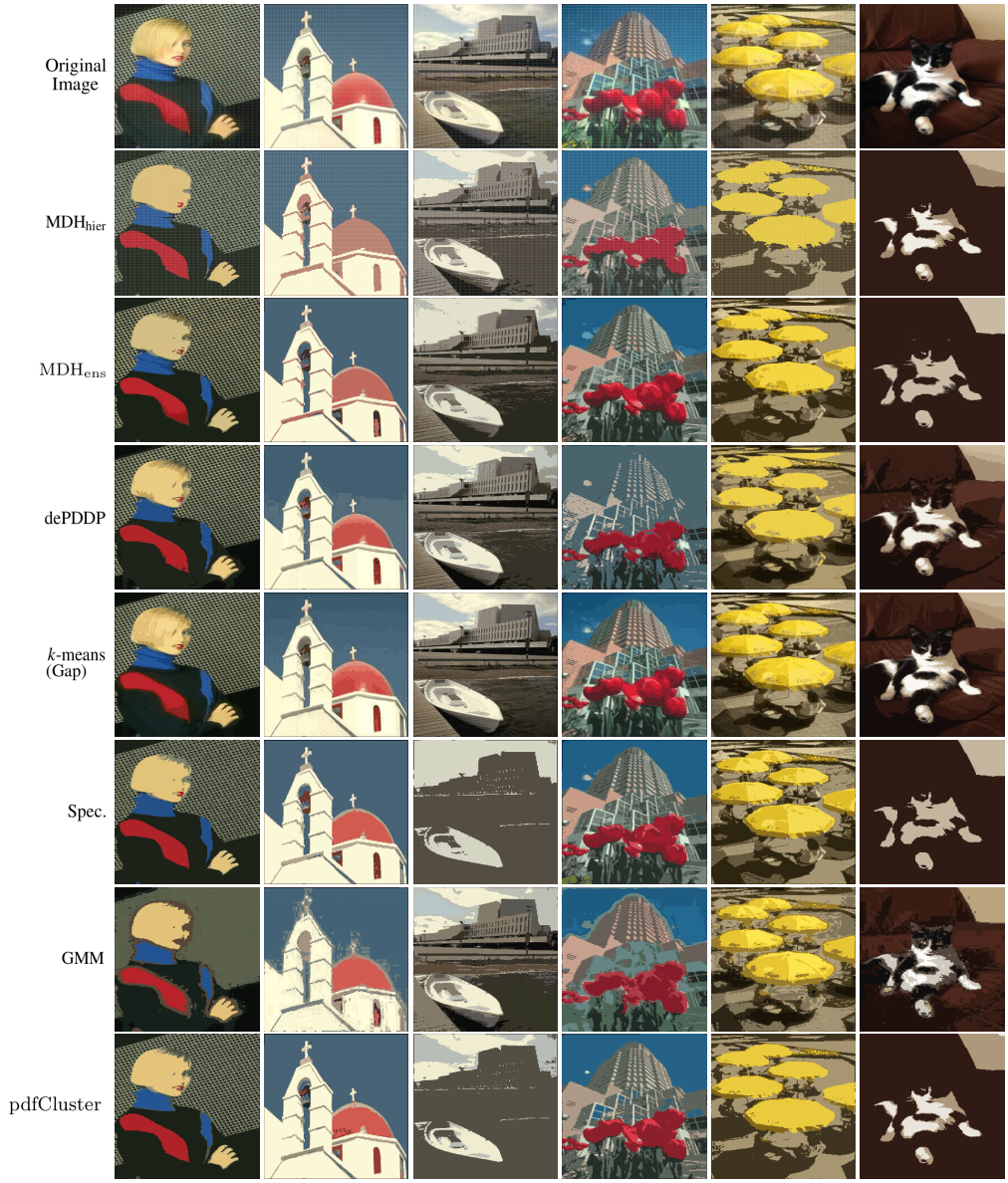


Figure 4.11: Image segmentation from MDH$_{\text{hier}}$, MDH$_{\text{ens}}$ and competing algorithms

ception of the third and sixth images in the case of spectral clustering). The segmentations from dePDDP and k-means++ with the Gap statistic appear almost identical to the original image, but this is because they locate approximately 30 clusters in each image compared to approximately five by MDH$_{\text{hier}}$. GMM produces a less accurate representation of the images with colours being mixed at the boundaries of segments.

4.5 Conclusions

We introduced an approach for density-based clustering for large, high-dimensional datasets containing diverse (mixed) types of attributes, and multiple clusters. High dimensionality and mixed data types are two typical properties of many real-world datasets that severely restrict the applicability of density-based clustering algorithms. To overcome the difficulties associated with high dimensionality we seek subspaces in which the data are optimally separable, in the sense that the induced linear cluster boundary does not intersect regions of high density, associated with clusters. This is achieved by either globally or locally minimising the density on a hyperplane criterion, so that the vector normal to the optimal hyperplane is the optimal one-dimensional projection to bi-partition the data. In contrast to established density-based clustering algorithms that attempt to identify regions of high estimated probability density in the full dimensional space, this approach requires only one-dimensional projections, mitigating the limitation to low-dimensional problems.

To extend the applicability of the proposed approaches to non-continuous observations, we investigate the location of an appropriate continuous representation of the mixed data, upon which low-density hyperplane separators may be computed. The choice of continuous representation critically affects the performance of all clustering algorithms. Of the three approaches we considered for this task we recommend using the constant shift embedding algorithm since this algorithm consistently enabled superior clustering performance by all the clustering methods, compared to alternative approaches.

We proposed a partitional and a divisive hierarchical algorithm based on a collection of minimum density hyperplanes to obtain the complete clustering and estimate the number of clusters.

A systematic simulation study across continuous and mixed data showed that if the true

clusters are associated with the modes of the continuous estimated density, the proposed approaches outperform competing density-based clustering algorithms, as well as *k*-means++, spectral clustering and GMM. Further, experiments across mixed and continuous real-world benchmark datasets with varying characteristics indicate that our approaches are highly competitive. Of the two clustering algorithms proposed, the most consistently completive performance was exhibited by the divisive hierarchical algorithm, MDH_{hier} so we advocate the use of this approach in practice.

5

Non-Linear Minimum Density Separators in Kernel Defined Feature Spaces

ABSTRACT

We introduce a kernel formulation of the minimum density hyperplane approach to clustering. This enables the identification of clusters that are not correctly identifiable using linear cluster separators in the input space, by non-linearly mapping the original observations into a, potentially high-dimensional, feature space. The location of the minimum density hyperplane in the feature space requires the solution of an n-dimensional, non-smooth, non-convex optimisation problem (where n is the number of observations). This is computationally expensive for large datasets, so we also propose an approximation technique using a subspace of the feature space to locate an approximate minimum density hyperplane. Using these hyperplanes to recursively bi-partition the mapped feature vectors in a divisive algorithm, allows the location of non-linearly separable clusters in arbitrarily oriented subspaces of the feature space, while estimating their number. An empirical analysis across benchmark datasets with varying characteristics suggests that the proposed approach is capable of locating high-quality partitions, which are highly competitive with alternative kernel-based clustering algorithms.

5.1 Introduction

In the density-based approach to clustering, clusters are defined as subsets of observations belonging to contiguous regions of high probability density, concentrated around the modes of some unknown probability density function $p_{\mathbf{X}}$, which may be estimated by a non-parametric estimated density $\hat{p}_{\mathbf{X}}$. As discussed in Chapter 4, the inaccuracy of density estimation in even moderate dimensions, restricts the direct location of clusters associated with high-density regions of $\hat{p}_{\mathbf{X}}$ to low-dimensional problems (Rinaldo and Wasserman, 2010). However, it is possible to apply the alternative formulation of locating low-density cluster boundaries that separate these high-density regions. This is known as the *low-density separation assumption*. These low-density cluster separators may be located using one-dimensional orthogonal projections of the data, making this alternative formulation applicable in high-dimensional datasets. However, the evaluation of the density intersected by a cluster boundary is computationally intractable for boundaries of arbitrary shapes, and therefore, the resulting separator is restricted to be a linear cluster boundary (hyperplane).

In Chapter 4, we proposed approaches to locate high-density clusters using a collection of minimum density hyperplane separators that identify linear cluster boundaries that intersect regions of minimal density while separating the regions of contiguous high probability density around the modes of $\hat{p}_{\mathbf{X}}$, since the subsets of observations in these regions are associated with clusters. This approach is capable of locating high-quality partitions in arbitrarily oriented subspaces. However, the ability to correctly identify clusters that are not linearly separable is an attractive property of density-based clustering generally, and the restriction to linear cluster boundaries imposed by the minimum density hyperplane (MDH) is an important limitation.

In this chapter, we propose the kernel MDH (KMDH) to overcome this limitation, al-

lowing the application of our approaches to low-density cluster separation to high-dimensional datasets whose clusters cannot be correctly identified by a collection of hyperplane separators in the space of the original observations. We first map the data non-linearly into a feature space, and a MDH is sought in the new feature space, where the hyperplane separator corresponds to a non-linear separator in the input space. The potentially infinite dimensionality of the feature space means it is not feasible to calculate the mapped observations (feature vectors) explicitly. However, we provide a formulation that permits the location of the KMDH in the feature space using the kernel matrix of pairwise inner products between the feature vectors, that is computed directly by the kernel function on the original observations. This also permits the KMDH to be computed for any dataset that permits the construction of a kernel matrix, including data with discrete or non-numeric attributes.

The location of the KMDH involves a non-smooth, non-convex optimisation problem over n variables, where n is the number of observations. In many applications of interest n can be very large, in which case an exhaustive search over all n dimensions for the KMDH is infeasible, and unlikely to be necessary to locate a high-quality separator. To overcome this we propose an approximation method, which we call the subspace KMDH (S-KMDH), that seeks hyperplanes in a subspace of the feature space. This reduces the search space for a low-density separator, and avoids searching over dimensions of the feature space that are unlikely to be meaningful for cluster separation.

Since any projection vectors that permit a meaningful cluster separator will lie in the n-dimensional space spanned by the feature vectors, the KMDH may be equivalently located using the projections of the feature vectors onto an n-dimensional orthonormal basis of the feature space, that spans the same space as the feature vectors. For the practical location of the KMDH we take this approach, using the orthonormal basis defined by kernel principal

component analysis (KPCA) (Schölkopf et al., 1998). This also permits an intuitive specification of an appropriate subspace for S-KMDH, which is located using the projections of the feature vectors onto the first $n' \ll n$ kernel principal components.

The remainder of this chapter is organised as follows: Section 5.2 presents the formulation of the MDH in feature space (KMDH), and the approximation of this using a smaller subspace of the feature space (S-KMDH) using the kernel matrix directly. Section 5.3 then describes how we locate the KMDH and S-KMDH practically, using the projections of the feature vectors onto the kernel principal components. Next, in Section 5.4 we discuss how we combine bi-partitions resulting from hyperplane separators of the feature vectors in a divisive algorithm, producing a complete clustering. Section 5.5 provides an empirical evaluation of the clustering results from the proposed divisive algorithm using bi-partitions from the KMDH and S-KMDH at each level of the hierarchy. The proposed approaches are compared to alternative kernel-based clustering algorithms across benchmark datasets with varying characteristics. Conclusions are given in Section 5.6.

5.2 Minimum Density Hyperplanes in the Feature Space

We assume a finite set of observations, $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ and a non-linear feature mapping $\mathbf{x}_i \mapsto \phi(\mathbf{x}_i)$ of $\mathcal{X}$ into the feature space $\mathcal{F}$, where $\phi(\cdot)$ is an arbitrary non-linear function. Let $\kappa(\cdot,\cdot)$ be the associated kernel function satisfying $\kappa(\mathbf{x}_i,\mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{F}}$. In order to define the density on the hyperplane with normal vector $\mathbf{v} \in \mathcal{F}$ and displacement from the origin $b \in \mathbb{R}$, as defined in Eq. (4.2) in the feature space, it is necessary to define the projections of the feature vectors onto $\mathbf{v}$. Depending on the choice of feature mapping and kernel function, $\mathcal{F}$ has the potential to be infinite-dimensional, making it infeasible to compute vectors in this space explicitly. Therefore, we cannot define the $\{\phi(\mathbf{x}_i)\}_{i=1}^n$ or an

arbitrary vector $\mathbf{v} \in \mathcal{F}$ directly.

However, any meaningful projection directions for clustering must lie within $\text{span}(\phi(\mathbf{x}_1), ..., \phi(\mathbf{x}_n))$. To prove this, consider the alternative case that $\mathbf{v}$ is orthogonal to $\text{span}(\phi(\mathbf{x}_1), ..., \phi(\mathbf{x}_n))$. In this case, the orthogonal projections of any $\phi(\mathbf{x}_j)$ onto $\mathbf{v}$ is $\langle \phi(\mathbf{x}_j), \mathbf{v} \rangle_{\mathcal{F}} = 0 \ \forall j$. Further consider the case that $\mathbf{v}$ has a component $\mathbf{v}_1$ in $\text{span}(\phi(\mathbf{x}_1), ..., \phi(\mathbf{x}_n))$ and a component $\mathbf{v}_2$ orthogonal to $\text{span}(\phi(\mathbf{x}_1), ..., \phi(\mathbf{x}_n))$. By definition, $\mathbf{v}_1$ may be expressed as a linear combination of the feature vectors, $\mathbf{v}_1 = \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i)$. Then, $\mathbf{v}$ may be decomposed into these two orthogonal components $\mathbf{v} = \mathbf{v}_1 + \mathbf{v}_2 = \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i) + \mathbf{v}_2$. In this case, the orthogonal projections of any $\phi(\mathbf{x}_j)$ onto $\mathbf{v}$ are given by $\langle \phi(\mathbf{x}_j), \mathbf{v} \rangle_{\mathcal{F}} = \langle \phi(\mathbf{x}_j), \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i) + \mathbf{v}_2 \rangle_{\mathcal{F}} = \langle \phi(\mathbf{x}_j), \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i) \rangle_{\mathcal{F}}$ since $\langle \phi(\mathbf{x}_j), \mathbf{v}_2 \rangle_{\mathcal{F}} = 0 \ \forall j$. Therefore, the projections of the feature vectors onto any vector $\mathbf{v} \in \mathcal{F}$ are independent of any component of $\mathbf{v}$ which lies outside $\text{span}(\phi(\mathbf{x}_1), ..., \phi(\mathbf{x}_n))$.

Defining the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ of pairwise inner products between $\{\phi(\mathbf{x}_i)\}_{i=1}^{n}$ such that $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ and the dual representation $\boldsymbol{\alpha} \in \mathbb{R}^n = (\alpha_1, ..., \alpha_n)$ of any vector $\mathbf{v}$ in $\text{span}(\phi(\mathbf{x}_1), ..., \phi(\mathbf{x}_n))$, such that $\mathbf{v} = \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i), \|\mathbf{v}\| = (\boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha})^{1/2} = 1$, the projection of a feature vector, $\phi(\mathbf{x}_j)$ onto $\mathbf{v}$ is given by (Shawe-Taylor and Cristianini, 2004),

$$\langle \phi(\mathbf{x}_j), \mathbf{v} \rangle_{\mathcal{F}} = \sum_{i=1}^{n} \alpha_i K_{ij}. \tag{5.1}$$

The integral of the estimated density along a hyperplane with unit normal $\mathbf{v} \in \mathcal{F}$ and

displacement from the origin $b \in \mathbb{R}$ in the feature space is therefore,

$$\hat{I}(\boldsymbol{\alpha}, b) = \frac{1}{nh\sqrt{2\pi}} \sum_{j=1}^{n} \exp\left\{ -\frac{\left(b - \langle \phi(\mathbf{x}_j), \mathbf{v} \rangle_{\mathcal{F}}\right)^2}{2h^2} \right\}, \tag{5.2}$$

where use use the notation $\hat{I}(\boldsymbol{\alpha}, b)$ to stress the fact that we rely on the dual representation of $\mathbf{v}$. This permits the location of the *kernel minimum density hyperplane* (KMDH) $H(\boldsymbol{\alpha}^\star, b^\star)$ with normal vector whose dual representation is $\boldsymbol{\alpha}^\star$ and displacement from the origin $b^\star$, that intersects a region of minimal density in the feature space. This is subject to sensible constraints on b, as discussed in Section 4.2.1, to ensure that the resulting hyperplane separates high-density regions in the feature space, and does not lie in low-density regions outside the range of $\{\phi(\mathbf{x}_i)\}_{i=1}^n$. The KMDH is the hyperplane that minimises the following projection pursuit optimisation problem,

$$\theta(\boldsymbol{\alpha}) = \min_{b \in \mathbb{R}} f(\boldsymbol{\alpha}, b), \tag{5.3}$$

$$f(\boldsymbol{\alpha}, b) = \hat{I}(\boldsymbol{\alpha}, b) + \frac{L}{\eta^\varepsilon} \max\{0, -\gamma\sigma_{\boldsymbol{\alpha}} - b, b - \gamma\sigma_{\boldsymbol{\alpha}}\}^{1+\varepsilon} \tag{5.4}$$

where $L = (e^{1/2}h^2 2\pi)^{-1}$, $\varepsilon, \eta \in (0, 1)$, $\sigma_{\boldsymbol{\alpha}}$ is the standard deviation of the projections of the feature vectors onto the vector whose dual vector is $\boldsymbol{\alpha}$ and γ is a user-defined parameter controlling the width of the search interval for b. This optimisation problem is closely related to the formulation of the MDH in the original data space in Section 4.2.1, where the properties and parameters of $f(\cdot)$ are discussed in more detail. We optimise $\theta(\boldsymbol{\alpha})$ using BFGS with inexact line searches as advocated by Lewis and Overton (2013).

5.2.1 Minimum Density Hyperplanes in Subspaces of the Feature Space

The search space for the KMDH in the feature space is n-dimensional. However, when n is large, the global optimisation of $\theta(\boldsymbol{\alpha})$ over all n dimensions is computationally expensive, and it is likely that a number of these dimensions are not necessary to locate a low-density separator of the feature vectors. Hence, in this section we consider using only a subspace of $\mathcal{F}$ to search for an approximate minimum density hyperplane. Our specific choice of subspace is discussed in Section 5.3, however, the methodology in this section is applicable to any subspace of $\mathcal{F}$. We denote the subspace of interest $\mathcal{F}' \subseteq \mathbb{R}^{n'}$ where $n' \ll n$.

Let $\mathbf{U} \in \mathbb{R}^{n \times n'} = [\mathbf{u}^{(1)}, ..., \mathbf{u}^{(n')}]$ be the matrix of the dual vectors of the n' orthogonal basis vectors of $\mathcal{F}'$ as columns. In $\mathcal{F}'$, the matrix of pairwise inner products of the feature vectors is $\mathbf{K}' = \mathbf{U}^\top \mathbf{K} \mathbf{U} \in \mathbb{R}^{n' \times n'}$. Then, the projection of the feature vector $\phi(\mathbf{x}_j)$ onto the unit vector $\mathbf{v} \in \text{span}(\phi(\mathbf{x}_1), ..., \phi(\mathbf{x}_n))$ whose dual vector is $\boldsymbol{\beta} = \sum_{i=1}^{n'} \alpha'_i \mathbf{u}_i \in \mathbb{R}^n$ is,

$$\langle \phi(\mathbf{x}_j), \mathbf{v} \rangle_{\mathcal{F}'} = \sum_{i=1}^{n'} \alpha'_i K'_{ij}. \tag{5.5}$$

We then seek the $\boldsymbol{\alpha}'^\star$ and $b^\star$ which minimise

$$\theta(\boldsymbol{\alpha}') = \min_{b \in \mathbb{R}} f(\boldsymbol{\alpha}', b), \tag{5.6}$$

$$f(\boldsymbol{\alpha}', b) = \hat{I}(\boldsymbol{\alpha}', b) + \frac{L}{\eta^\varepsilon} \max\{0, -\gamma\sigma_{\boldsymbol{\alpha}'} - b, b - \gamma\sigma_{\boldsymbol{\alpha}'}\}^{1+\varepsilon}, \tag{5.7}$$

$$\hat{I}(\boldsymbol{\alpha}', b) = \frac{1}{nh\sqrt{2\pi}} \sum_{j=1}^{n} \exp\left\{ -\frac{\left(b - \langle \phi(\mathbf{x}_j), \mathbf{v} \rangle_{\mathcal{F}'}\right)^2}{2h^2} \right\} \tag{5.8}$$

where $\sigma_{\boldsymbol{\alpha}'}$ is the standard deviation of the projections defined by Eq. 5.5. The *subspace kernel minimum density hyperplane* (S-KMDH) is then the hyperplane $H(\boldsymbol{\alpha}'^\star, b^\star)$ that solves the optimisation problem in Eqs.(5.6) - (5.8). The smaller dimensionality of $\mathcal{F}'$ reduces

the computational cost of locating a low-density separator in the feature space, and avoids searching over dimensions which are unlikely to be useful for cluster detection.

5.3 Locating the KMDH using Kernel Principal Component Analysis

Since the search space for the KMDH is practically limited to the n-dimensional space spanned by the feature vectors, the formulation for locating the dual vector $\boldsymbol{\alpha}$ above is equivalent to locating the KMDH using the projections of the feature vectors onto an n-dimensional orthonormal basis of $\text{span}(\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n))$. These n-dimensional projections of the feature vectors may be treated as a mapped set of observations, and clustered by the same procedure as the original observations, therefore avoiding the explicit formulation of the MDH in the feature space. To construct this basis we use kernel principal component analysis (KPCA) (Schölkopf et al., 1998), which is an extension of standard (linear) PCA to feature spaces. KPCA operates directly on the kernel matrix, and locates an orthonormal basis of the space spanned by the feature vectors with decreasing variability along its axes.

Given a kernel matrix $\mathbf{K} = [K_{ij}]$ such that $K_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{F}}$, that has been centred such that $\sum_{i=1}^{n} \phi(\mathbf{x}_i) = \mathbf{0}$, the covariance matrix associated with $\{\phi(\mathbf{x}_i)\}_{i=1}^{n}$ is,

$$\boldsymbol{C} = \frac{1}{n} \sum_{i=1}^{n} \phi(\mathbf{x}_i) \phi(\mathbf{x}_i)^{\top}.$$

In KPCA, we require the eigenvalues $\lambda^{(1)}, \ldots, \lambda^{(n)}$ and eigenvectors $\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}$ of $\boldsymbol{C}$ which satisfy $\lambda^{(k)} \mathbf{v}^{(k)} = \boldsymbol{C} \mathbf{v}^{(k)}$ for $k = 1, \ldots, n$. Since all the eigenvectors $\mathbf{v}^{(k)}$ must lie in $\text{span}(\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n))$, we may consider the equivalent system,

$$\lambda^{(k)} \langle \phi(\mathbf{x}_i), \mathbf{v}^{(k)} \rangle_{\mathcal{F}} = \langle \phi(\mathbf{x}_i), \boldsymbol{C} \mathbf{v}^{(k)} \rangle_{\mathcal{F}} \tag{5.9}$$

for all $i, k = 1, ..., n$. If we let $\mathbf{u}^{(k)}$ be the dual vector of $\mathbf{v}^{(k)}$ such that $\mathbf{v}^{(k)} = \sum_{j=1}^{n} u_j^{(k)} \phi(\mathbf{x}_j)$ where $u_j^{(k)}$ denotes the jth element of the kth dual vector $\mathbf{u}^{(k)}$, then

$$\langle \phi(\mathbf{x}_i), \mathbf{v}^{(k)} \rangle_{\mathcal{F}} = \langle \phi(\mathbf{x}_i), \sum_{j=1}^{n} u_j^{(k)} \phi(\mathbf{x}_j) \rangle_{\mathcal{F}} = \sum_{j=1}^{n} u_j^{(k)} \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathcal{F}} = \mathbf{K}_{i,:} \mathbf{u}^{(k)} \tag{5.10}$$

where $\mathbf{K}_{i,:}$ denotes the ith row of $\mathbf{K}$. Therefore, considering all $i = 1, ..., n$, $\langle \mathbf{\Phi}, \mathbf{v}^{(k)} \rangle_{\mathcal{F}} = \mathbf{K}\mathbf{u}^{(k)}$ where $\mathbf{\Phi}$ is the matrix associated with the set of feature vectors $\{\phi(\mathbf{x}_i)\}_{i=1}^{n}$. Also, by definition of $\mathbf{K}$, $\boldsymbol{C} = \frac{1}{n}\mathbf{K}$, so the eigen-system in Eq. (5.9) becomes,

$$n\lambda^{(k)} \mathbf{K}\mathbf{u}^{(k)} = \mathbf{K}^2 \mathbf{u}^{(k)}$$

$$n\lambda^{(k)} \mathbf{u}^{(k)} = \mathbf{K}\mathbf{u}^{(k)}$$

for $k = 1, \ldots, n$. To ensure that the corresponding principal component vector $\mathbf{v}^{(k)}$, is normalised, it is necessary to set the constraint,

$$\langle \mathbf{v}^{(k)}, \mathbf{v}^{(k)} \rangle_{\mathcal{F}} = \mathbf{u}^{(k)\top} \mathbf{K}\mathbf{u}^{(k)} = 1.$$

By Eq. (5.10), the projection of the mapped feature vector $\phi(\mathbf{x}_i)$ onto the kernel principal component vector $\mathbf{v}^{(k)}$ is given by

$$\langle \phi(\mathbf{x}_i), \mathbf{v}^{(k)} \rangle_{\mathcal{F}} = \sum_{j=1}^{n} u_j^{(k)} K_{ij} = \sum_{j=1}^{n} u_j^{(k)} K_{ji} = \boldsymbol{K}_{i,:} \mathbf{u}^{(k)}$$

For the practical implementation of the KMDH, we use the projections of each of the fea-

ture vectors onto the orthonormal basis of $\mathcal{F}$ defined by the kernel principal components,

$$\mathcal{X}^{\mathcal{F}} = \{\mathbf{x}_i^{\mathcal{F}}\}_{i=1}^n = \{\mathbf{K}_{i,:}\boldsymbol{U}\}_{i=1}^n \subset \mathbb{R}^n, \tag{5.11}$$

where $\mathbf{U} = [\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(n)}]$ is the matrix of column-wise dual kernel principal components. Then, the MDH may be located using $\mathcal{X}^{\mathcal{F}}$ in the same way as for any dataset $\mathcal{X}$, the formulation for which is given in Chapter 4.

This approach to locating the KMDH is also convenient for the consideration of a smaller subspace of $\mathcal{F}$, in which to search for an approximate minimum density separator, as discussed in Section 5.2.1. To construct an appropriate, lower-dimensional subspace of $\mathcal{F}$, which we denote $\mathcal{F}'$, in which to search for a low-density separator of the mapped feature vectors, we use the eigenvalues from KPCA to exclude directions that contribute very little to the overall variability in $\{\phi(\mathbf{x}_i)\}_{i=1}^n$. Although there is no guarantee that directions of high variability will be meaningful for cluster detection (Kriegel et al., 2009), it is arguably unlikely that directions which exhibit almost no variability are relevant for clustering. Hence, we consider the subspace $\mathcal{F}'$ spanned by the first $n' \ll n$ kernel principal components, which capture a pre-specified percentage of the variability in $\{\phi(\mathbf{x}_i)\}_{i=1}^n$. Then, we locate the S-KMDH using the projections of the feature vectors onto the orthonormal basis defined by the first n' kernel principal components,

$$\mathcal{X}^{\mathcal{F}'} = \{\mathbf{x}_i^{\mathcal{F}'}\}_{i=1}^n = \{\mathbf{K}_{i,:}\boldsymbol{U}_{:,1:n'}\}_{i=1}^n \subset \mathbb{R}^{n'}, \tag{5.12}$$

where $\boldsymbol{U}_{:,1:n'}$ denotes the first n' columns of $\boldsymbol{U}$.

To obtain a complete clustering, we recursively bi-partition successive subsets of the feature vectors using the KMDH (or the S-KMDH). To allow the estimation of the number of clusters, we require a stopping rule to determine when a subset of $\{\phi(\mathbf{x}_i)\}_{i=1}^n$ should not be separated further. For this, we take the same approach as MDH_{hier}, proposed in Section 4.2.2, where instead of the set of observations $\mathcal{X}$, we have the set of projections of the feature vectors onto the kernel principal components $\mathcal{X}^{\mathcal{F}} = \{\mathbf{x}_i^{\mathcal{F}}\}_{i=1}^n$, as defined in Eq. (5.11). Given the set of projections assigned to the cluster of interest, $\mathcal{X}_C^{\mathcal{F}} \subseteq \mathcal{X}^{\mathcal{F}}$ and the corresponding KMDH with unit normal $\mathbf{v}_C$, the relative depth along $\mathbf{v}_C$ is

$$\text{RelativeDepth}(\mathbf{v}_C, b_C; \mathcal{X}_C^{\mathcal{F}}) = \frac{\min\left\{\hat{p}_{\mathbf{v}_C^\top \mathbf{x}^{\mathcal{F}}}(m_l), \hat{p}_{\mathbf{v}_C^\top \mathbf{x}^{\mathcal{F}}}(m_r)\right\} - \hat{p}_{\mathbf{v}_C^\top \mathbf{x}^{\mathcal{F}}}(b_C)}{\hat{p}_{\mathbf{v}_C^\top \mathbf{x}^{\mathcal{F}}}(b_C)}, \quad (5.13)$$

where $\hat{p}_{\mathbf{v}_C^\top \mathbf{x}^{\mathcal{F}}}$ is the one-dimensional estimated density of the projections of $\mathcal{X}_C^{\mathcal{F}}$ onto $\mathbf{v}_C$ and m_l and m_r are the locations of the two largest modes of $\hat{p}_{\mathbf{v}_C^\top \mathbf{x}^{\mathcal{F}}}$ to the left and right of b_C respectively.

At lower levels of the hierarchy, the increased sparsity of the points $\mathcal{X}_C^{\mathcal{F}}$ spanning n dimensions allows the KMDH to locate projection vectors along which $\hat{p}_{\mathbf{v}_C^\top \mathbf{x}^{\mathcal{F}}}$ is multimodal, even if a true low-density separator of dense regions in $\mathcal{X}_C^{\mathcal{F}}$ does not exist. Therefore, we test the appropriateness of the KMDH to separate $\mathcal{X}_C^{\mathcal{F}}$ by randomly splitting $\mathcal{X}_C^{\mathcal{F}}$ into a training and a hold-out sample. We compute the KMDH on the training sample, and then evaluate the relative depth of this hyperplane on the hold-out sample. This relative depth is compared to the Monte-Carlo estimates of the relative depth in a null unimodal sample, to assess if the multimodality in $\hat{p}_{\mathbf{v}_C^\top \mathbf{x}^{\mathcal{F}}}$ is sufficient to indicate an appropriate separator of $\mathcal{X}_C^{\mathcal{F}}$. We use the uniform distribution to generate our null samples, as this is the

standard choice in modality testing (Hartigan and Hartigan, 1985; Hartigan, 1977). If the relative depth of the KMDH on the hold-out sample exceeds a specified percentile of the relative depth in the null samples, we accept the partition and locate the final KMDH on the entire set $\mathcal{X}_C^{\mathcal{F}}$. This same procedure is applied for S-KMDH, using the projections of the feature vectors into an orthonormal basis of $\mathcal{F}'$, $\mathcal{X}^{\mathcal{F}'}$ as defined in Eq. (5.12). We refer to the divisive clustering algorithms which bi-partition the feature vectors using KMDH and S-KMDH at each level of the hierarchy as $\text{KMDH}_{\text{hier}}$ and $\text{S-KMDH}_{\text{hier}}$ respectively. These two divisive algorithms are summarised in Algorithms 4 and 5 respectively.

Algorithm 4 Hierarchical Kernel Minimum Density Hyperplanes ($\text{KMDH}_{\text{hier}}$)

Require: Kernel matrix $\mathbf{K}$
Compute the dual vectors $\boldsymbol{U}$ of the kernel principal components of the mapped feature vectors $\{\phi(\mathbf{x}_i)\}_{i=1}^n$ whose inner products are contained in $\mathbf{K}$.
Project $\{\phi(\mathbf{x}_i)\}_{i=1}^n$ onto the kernel principal components $\mathcal{X}^{\mathcal{F}} = \{\mathbf{K}_{i,:}\boldsymbol{U}\}_{i=1}^n$.
Cluster $\mathcal{X}^{\mathcal{F}}$ using MDH_{hier} as described in Algorithm 2 to give the vector of cluster labels $\boldsymbol{\pi}$ and estimated number of clusters $\hat{k} = \max \boldsymbol{\pi}$.
return $\boldsymbol{\pi}, \hat{k}$

Algorithm 5 Hierarchical Subspace Kernel Minimum Density Hyperplanes ($\text{S-KMDH}_{\text{hier}}$)

Require: Kernel matrix $\mathbf{K}$
Compute the first n' dual vectors $\boldsymbol{U}_{:,1:n}$ of the kernel principal components of the mapped feature vectors $\{\phi(\mathbf{x}_i)\}_{i=1}^n$ whose inner products are contained in $\mathbf{K}$.
Project $\{\phi(\mathbf{x}_i\})_{i=1}^n$ onto the first n' kernel principal components $\mathcal{X}^{\mathcal{F}'} = \{\mathbf{K}_{i,:}\boldsymbol{U}_{:,1:n'}\}_{i=1}^n$.
Cluster of $\mathcal{X}^{\mathcal{F}'}$ using MDH_{hier} as described in Algorithm 2 to give the vector of cluster labels $\boldsymbol{\pi}$ and estimated number of clusters $\hat{k} = \max \boldsymbol{\pi}$.
return $\boldsymbol{\pi}, \hat{k}$

5.4.1 Computational Complexity

In this subsection, we discuss the computational complexity of $\text{KMDH}_{\text{hier}}$ and $\text{S-KMDH}_{\text{hier}}$. First, the construction of the kernel matrix had computational cost $\mathcal{O}(n^2 d)$ where n and d are the number of observations and dimensions in the original dataset respectively. This cost may be reduced by constructing an approximate kernel matrix by techniques such as the Nyström approximation (Fowlkes et al., 2004). For both $\text{KMDH}_{\text{hier}}$ and $\text{S-KMDH}_{\text{hier}}$,

it is necessary to compute the eigenvalues and eigenvectors of $\mathbf{K}$ for KPCA and the projections of $\{\phi(\mathbf{x}_i)\}_{i=1}^n$ onto the kernel principal components of interest. Both these operations have cost $\mathcal{O}(n^3)$ for KMDH$_{\text{hier}}$ and $\mathcal{O}(n^2 n')$ for S-KMDH$_{\text{hier}}$. This is computationally expensive for large n but are only performed once, and not at each level of the hierarchy. Once the projections onto the kernel principal components have been computed, the KMDH and S-KMDH are located at each level of the divisive procedure, by iteratively optimising the normal vector to the separating hyperplane $\mathbf{v}$ and its displacement from the origin b in the feature space.

At each iteration, KMDH$_{\text{hier}}$ projects $\mathcal{X}^{\mathcal{F}}$ onto $\mathbf{v}$, at a cost of $\mathcal{O}(n(n+1))$. Then, to obtain the projection index $\theta(\boldsymbol{\alpha})$, it is necessary to minimise the penalised objective $f(\boldsymbol{\alpha}, b)$ with respect to b. This minimisation is possible by evaluating $f(\boldsymbol{\alpha}, b)$ on a grid of m points, involving m evaluations of a density estimate with n components. The cost of this may be reduced from $\mathcal{O}(mn)$ to $\mathcal{O}(n+m)$ using the improved fast Gauss transform (Morariu et al., 2009). To compute the minimiser(s) to within the desired accuracy, ϵ, bisection may be used which requires $\mathcal{O}(-\log_2 \epsilon)$ iterations. The subsequent minimisation of $\theta(\boldsymbol{\alpha})$ is done using BFGS as advocated by Lewis and Overton (2013). This can be done at a cost of $\mathcal{O}(n^2)$ per iteration (Nocedal and Wright, 2006, Pg. 140) plus the cost of function evaluations of $f(\boldsymbol{\alpha}, b)$ and gradient evaluations with cost $\mathcal{O}(n(n+1))$. For S-KMDH$_{\text{hier}}$, the computational cost of computing the projections of $\mathcal{X}^{\mathcal{F}'}$ onto $\mathbf{v}$ and the gradient evaluations is reduced to $\mathcal{O}(n'(n'+1))$. Therefore, given that the set of mapped feature vectors and their projections onto an orthonormal basis have been computed, the overall computational complexity of locating the KMDH and S-KMDH at each level of the hierarchy are $\mathcal{O}(n^2+n)$ and $O(n'^2+n')$ per iteration respectively. On a representative dataset for our experiments, KMDH$_{\text{hier}}$ took about 30 minutes to produce a complete clustering, while

S-KMDH$_{\text{hier}}$ took approximately 20 minutes using R code which was not particularly optimised. These results were comparable to other competing algorithms and were significantly faster than k-means++ with the gap statistic.

5.5 Experimental Results

In this section, we conduct an empirical evaluation of the proposed approaches, KMDH$_{\text{hier}}$ and S-KMDH$_{\text{hier}}$. We compare the quality of the partitions produced by these algorithms to alternative kernel-based approaches. The methods considered are:

1. Kernel k-means (Dhillon et al., 2004; Zhang and Rudnicky, 2002) which is a kernel variant of the classical k-means algorithm, where the clustering solution minimises the sum of squared distances between the feature vectors and their assigned cluster centroid in the feature space. The particulars of this algorithm are given in Section 2.1.2. We are not aware of a procedure to estimate the number of clusters for this algorithm so we provide the true number of clusters as an input parameter.

2. Spectral clustering (von Luxburg, 2007) where the kernel matrix is equivalent to the adjacency matrix of the graph $\mathcal{G}(\mathcal{X}, \mathcal{E})$. In our experiments we use normalised spectral clustering (Ng et al., 2002).

3. dePDDP (Tasoulis et al., 2010) extended to the feature space by projecting the data onto the first kernel principal component, and splitting at the minimiser of the estimated density of the projections between the two outer-most modes. As in the original dePDDP algorithm, we terminate the divisive splitting procedure when the estimated density of the projections is unimodal. This is equivalent to applying the standard dePDDP algorithm on $\mathcal{X}^{\mathcal{F}}$ as defined in Eq. (5.11). We refer to this extension to feature spaces as K-dePDDP. Comparing KMDH to this algorithm highlights the advantage of optimising the projection direction to locate a minimum density separator of the feature vectors.

5.5.1 Details of Implementation

For all algorithms we rely on the same kernel matrix, so differences in performance relate to the different clustering objectives, and not a different feature mapping. We use the Gaussian

(radial basis) kernel,

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right\}$$

since this is the most widely used in the literature. The performance of any kernel-based algorithm is sensitive to the selection and tuning of an appropriate kernel function. This is an open problem, and a detailed investigation into this is beyond the scope of this work. We therefore apply the local scaling approach for the Gaussian kernel proposed in Zelnik-Manor and Perona (2004),

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{s_i s_j}\right\}$$

where s_i and s_j are the distances from the ith and jth observations to their seventh nearest neighbours respectively, as recommended by Zelnik-Manor and Perona (2004). This allows for clusters on multiple scales, and is very effective in our experience.

For KMDH$_{\text{hier}}$, S-KMDH$_{\text{hier}}$ and K-dePDDP, the bandwidth used to construct the estimated density of the projections can critically affect performance. For K-dePDDP, we use the standard rule recommended by Tasoulis et al. (2010) of $h = \hat{\sigma}_{kpc_1}(4/(3n))^{1/5}$ where $\hat{\sigma}_{kpc_1}$ is the standard deviation of the projections of the feature vectors onto the first kernel principal component. Since KMDH$_{\text{hier}}$ and S-KMDH$_{\text{hier}}$ actively seek projection directions with a multimodal density, we apply the rule $h = 0.9\hat{\sigma}n^{-1/5}$ since this is the optimal choice for multimodal densities (Silverman, 1986). In our experiments we fix $\hat{\sigma} = \hat{\sigma}_{kpc_1}$ to maintain a fixed bandwidth regardless of the projection vector.

The other parameter which affects the quality of a bi-partition using KMDH and S-KMDH is the interval width parameter γ. As described in Section 4.4.1, we follow the approach of Pavlidis et al. (2016), this is initialised close to zero, inducing a balanced partition.

This is gradually increased to $\gamma_{\max} = 1$ to allow convergence to the minimum integrated density. Although generally robust to local convergence, the quality of bi-partitions located using KMDH and S-KMDH can be dependent on initialisation. We investigated using the kernel principal components and a random initialisation. We found the most effective technique was to initialise on both the first and second kernel principal components, and retain the hyperplane with the best relative depth. The results presented in Section 5.5.2 use this approach. For the choice of dimensionality of the subspace n' in S-KMDH$_{\text{hier}}$, we used the eigenvalues from KPCA to select the dimensionality that retained 90% of the variability in the feature vectors $\{\phi(\mathbf{x}_i)\}_{i=1}^{n}$. In our experience, this significantly reduced the dimensionality of the search space, without a substantial sacrifice in clustering performance. Finally, for the stopping procedure described in Section 5.4, we compare the relative depth of the KMDH (or S-KMDH) to the 97.5th percentile of the relative depth from 10,000 null uniform samples. In practice, we found that any threshold above the 90th percentile was effective for the rejection of hyperplanes which were not suitable low-density separators.

For the comparison to kernel k-means and spectral clustering, we used the implementations in the `kernlab` package for R Karatzoglou et al. (2004) with the same kernel matrix as for our algorithms and K-dePDDP. These implementations operate directly on the kernel matrix.

As described in Section 4.4.2, we evaluated the performance of all competing algorithms using different performance measures that are appropriate for comparing clusterings with potentially different numbers of clusters. The choice of performance measure did not alter the relative performance of the different algorithms, and we thus report performance with respect to normalised mutual information (NMI) (Strehl and Ghosh, 2002). NMI takes values in the range $[0, 1]$ with higher values indicating greater levels of information shared

between the distributions of the assigned and true cluster labels.

5.5.2 Performance Evaluation

In this section, we present the performance of KMDH$_{\text{hier}}$ and S-KMDH$_{\text{hier}}$ compared to the competing algorithms considered over 22 real-world benchmark datasets with varying numbers of observations, n, dimensions, d, and clusters, k. These characteristics are summarised in Table 5.1.

Table 5.1: Main characteristics of real datasets considered.

Dataset	n	d	k
Banknote[3]	1372	4	2
Cal101-16[1]	2901	256	6
Cal101-28[1]	2901	784	6
Coil20[2]	1420	16384	20
Dermatology[3]	366	34	6
Heart Disease[3]	294	13	5
Image Segmentation[3]	2309	19	7
Ionosphere[3]	351	33	2
Iris[3]	150	4	3
Isolet[3]	7797	617	26
Multi. Digits[3]	2000	216	10
Opt. Digits[3]	5620	64	10
Pen Digits[3]	10992	16	10
Phoneme [4]	4506	256	5
Satellite[3]	6435	36	6
Seeds[3]	210	7	3
Smartphone[3]	10929	561	12
Soy Bean[3]	682	35	19
Synth[3]	600	60	6
Vote[3]	435	16	2
Wine[3]	178	13	3
Yale Faces[5]	5850	1200	10

[1] UCI machine learning repository (Lichman, 2013)

[2] (Marlin, 2014) available from people.cs.umass.edu/~marlin/data.shtml

[3] (Nene et al., 1996) available from cs.columbia.edu/CAVE/software/softlib/coil-20.php

[4] (Hastie et al., 1995) available from statweb.stanford.edu/tibs/ElemStatLearn/data.html

[5] (Georghiades et al., 2001) available from cervisia.org/machine_learning_data.php

Table 5.2 reports the performance of KMDH$_{\text{hier}}$, S-KMDH$_{\text{hier}}$ and the competing algorithms across the 22 datasets considered. Each cell reports the NMI and the estimated number of clusters (where applicable) for each algorithm on each dataset. For each dataset,

Table 5.2: Clustering performance of $KMDH_{hier}$, S-$KMDH_{hier}$, K-dePDDP, Kernel *k*-means and spectral clustering on real benchmark datasets. The top row of each cell reports NMI and the second the estimated number of clusters (where applicable). For each dataset the best performing algorithm is highlighted.

		$KMDH_{hier}$	S-$KMDH_{hier}$	K-dePDDP	K-*k*-means	Spectral
Banknote	NMI	0.193	0.225	0.364	0.032	0.566
	k	10	14	194	-	4
Cal101-16	NMI	0.591	0.593	0.570	0.515	0.581
	k	15	17	40	-	2
Cal101-28	NMI	0.584	0.573	0.548	0.540	0.484
	k	15	17	47	-	2
Coil 20	NMI	0.779	0.780	0.677	0.488	0.573
	k	22	24	172	-	5
Dermatology	NMI	0.752	0.759	0.789	0.645	0.042
	k	3	4	5	-	7
Heart Disease	NMI	0.242	0.195	0.055	0.280	0.011
	k	2	2	21	-	2
Image Seg.	NMI	0.507	0.524	0.568	0.152	0.625
	k	25	31	239	-	3
Ionosphere	NMI	0.536	0.366	0.000	0.275	0.368
	k	2	3	16	-	4
Iris	NMI	0.717	0.000	0.451	0.213	0.759
	k	2	1	8	-	2
Isolet	NMI	0.571	0.598	0.565	0.442	0.604
	k	20	22	55	-	52
Multi Features	NMI	0.730	0.733	0.597	0.558	0.702
	k	17	15	16	-	8
Opti. Digits	NMI	0.717	0.694	0.613	0.475	0.661
	k	19	27	39	-	17
Pen Digits	NMI	0.707	0.702	0.338	0.403	0.375
	k	56	63	27	-	2
Phonome	NMI	0.777	0.732	0.635	0.735	0.655
	k	6	6	17	-	3
Satellite	NMI	0.550	0.545	0.000	0.372	0.393
	k	22	26	1	-	2
Seeds	NMI	0.589	0.602	0.525	0.485	0.588
	k	2	2	15	-	6
Smartphone	NMI	0.606	0.572	0.549	0.487	0.559
	k	13	18	54	-	2
Soy Bean	NMI	0.662	0.631	0.000	0.466	0.390
	k	13	14	1	-	3
Synth	NMI	0.802	0.851	0.757	0.742	0.765
	k	5	5	19	-	3
Votes	NMI	0.437	0.456	0.000	0.282	0.103
	k	2	3	1	-	3
Wine	NMI	0.000	0.741	0.779	0.892	0.393
	k	1	3	7	-	6
Yale Faces	NMI	0.724	0.710	0.000	0.430	0.039
	k	48	63	1	-	2

the best performing algorithm is indicated in red. The uniform sample in the stopping criterion for $KMDH_{hier}$ and S-$KMDH_{hier}$induces an element of variability in the results of

these algorithms. However, the difference in the partitions for these methods with different random samples in the stopping criterion induced very small differences in the overall result, with very low standard deviation in the performances of partitions, and an NMI between partitions of approximately 0.95 in all cases. Across these datasets $\text{KMDH}_{\text{hier}}$ and $\text{S-KMDH}_{\text{hier}}$ perform very competitively compared to the alternative algorithms, often providing the best performance. For the majority of the datasets, the MDH-based algorithms perform better than K-dePDDP, suggesting that optimising the projection vector is worthwhile to locate a higher quality partition. The performance of $\text{S-KMDH}_{\text{hier}}$ is similar to, or better than $\text{KMDH}_{\text{hier}}$ in all datasets except Iris. This indicates that in the majority of cases, an exhaustive search over all n dimensions spanned by the feature vectors is not required to locate a suitable low-density separator. In fact, in some cases, for example on the Wine dataset, failing to exclude dimensions which do not contain useful information for clustering severely inhibits the performance of $\text{KMDH}_{\text{hier}}$. Both spectral clustering and kernel k-means can locate good quality partitions, in some cases performing better than the MDH-based approaches. However, these algorithms do not perform as consistently well as $\text{KMDH}_{\text{hier}}$ and $\text{S-KMDH}_{\text{hier}}$ for these datasets. In general, the low-density separation approaches have a tendency to overestimate the number of clusters. This is a result of the sparsity of the feature vectors in the high-dimensional feature space allowing the location of low-density separators, which incorrectly split the true clusters. This is especially evident for K-dePDDP, which does not terminate until the estimated density of the projections of the feature vectors onto the first kernel principal component is strictly unimodal, unlike the more pessimistic stopping rule applied for $\text{KMDH}_{\text{hier}}$ and $\text{S-KMDH}_{\text{hier}}$. However, these partitions achieve high cluster homogeneity (purity), indicating that the clusters located contain observations from the same true class. By contrast, the auto-tuned spectral cluster-

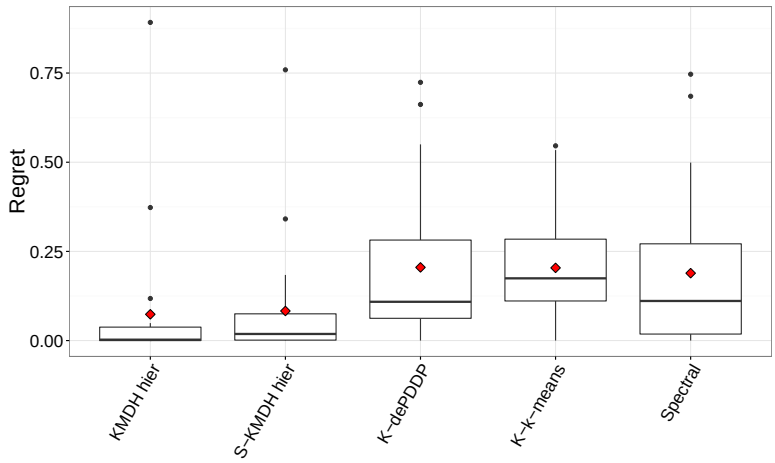


Figure 5.1: Boxplots of regret for each algorithm considered based on NMI over benchmark datasets. Mean regret is depicted with a red dot.

ing algorithm tends to underestimate the number of clusters.

To assess the relative performance of our proposed methods and alternative algorithms, Figure 5.1 provides boxplots of regret, with respect to NMI (defined in Eq. (4.8)), associated with each algorithm over the benchmark datasets considered. For each dataset, the regret of an algorithm is the difference in performance between the best performing algorithm and the algorithm of interest. Therefore, a regret of zero indicates the best performance. In these experiments, the value of the regret did not noticeably differ with the number of observations or dimensions in the dataset. Figure 5.1 shows that $\text{KMDH}_{\text{hier}}$ achieves the lowest regret for these datasets, with a median regret very close to zero, indicating that this algorithm has the best relative performance. S-$\text{KMDH}_{\text{hier}}$ also has a median regret very close to zero but with slightly more variability than $\text{KMDH}_{\text{hier}}$. However, for larger datasets, the reduction in computational cost of using a subspace to approximate the minimum density hyperplane may be beneficial for a small sacrifice in performance. The three alternative algorithms have similar median regret, although spectral clustering has more variable relative performance by comparison to K-dePDDP and kernel k-means. Both K-dePDDP and spectral clustering have slightly lower median regret than kernel k-means despite this algorithm

being provided with the correct number of clusters.

5.6 Conclusions

In this chapter, we introduced an approach to locate minimum density separators, which are appropriate for locating high-density clusters in high-dimensional datasets, with clusters that cannot be correctly identified using linear cluster boundaries. This is done by non-linearly mapping the input observations into a feature space via a valid kernel function. Hyperplane separators in the feature space then correspond to non-linear separators in the input space. We call the minimum density linear separator in the feature space the KMDH. Since the density intersected by a hyperplane in the feature space can be evaluated with only the pairwise inner products between the mapped observations and the dual representation of the vector normal to the hyperplane, the explicit calculation of the mapped feature vectors is avoided, and the kernel matrix may be used to compute the KMDH.

In practice, the search space for the KMDH is restricted to the n-dimensional space spanned by the feature vectors. For large datasets, searching over all of these n dimensions becomes computationally expensive. Furthermore, it is likely that some of these dimensions do not contain useful information for cluster detection. Therefore, we propose to approximate the KMDH using a smaller subspace of the feature space. The resulting hyperplane separator is called the S-KMDH.

Practically, we locate the KMDH using the projections of the feature vectors onto the n-dimensional orthonormal basis spanning the same space, defined by KPCA. This is equivalent to locating the KMDH in the feature space directly, using the kernel matrix. Similarly, the S-KMDH is computed using the projections of the feature vectors onto the n'-dimensional orthonormal basis of the feature vectors spanned by the first n' kernel principal

components, that retain a specified proportion of the variability in the feature space.

We combine the bi-partitions from KMDH and S-KMDH in divisive algorithms, called $\text{KMDH}_{\text{hier}}$ and $\text{S-KMDH}_{\text{hier}}$ respectively. These divisive algorithms automatically estimate the number of clusters by assessing the suitability of a potential hyperplane for the separation of high-density regions in the feature space, associated with clusters.

Experimentation across real-world benchmark datasets with varying characteristics indicate that our proposed approaches locate high-quality clustering results, which are often better than alternative kernel-based algorithms. Our results indicate that in most cases searching over a subspace of the feature space is sufficient to locate a high-quality separator, whose performance is competitive with, or better than the global KMDH. The advantage of this subspace approach is particularly relevant for large datasets where a search over all n dimensions may be computationally infeasible.

6

Computationally Efficient Low-Density Cluster Separation with Random Projection

Abstract

We propose an approach for the computationally efficient location low-density cluster separators of large, high-dimensional datasets using univariate random projections. We bi-partition the data at the minimiser of the estimated density of an appropriate set of one-dimensional random projections to locate a cluster boundary that separates high-density regions associated with clusters. We combine these bi-partitions in a divisive algorithm to locate a complete clustering, which automatically estimates the number of clusters. A systematic simulation study and an empirical evaluation of the performance of our proposed approach across real-world benchmark datasets indicate that random projection allows the location of high-quality low-density cluster separators. The performance of the partitions located through random projection are competitive with the low-density separators located by univariate projections computed by principal component analysis, independent component analysis and the minimum density hyperplane, and are much less computationally expensive than these alternative projection techniques. Therefore, the proposed approach is highly attractive for clustering large, high-dimensional datasets, where the computational cost of alternative projection techniques makes their implementation infeasible practically.

6.1 Introduction

For datasets with large numbers of features (dimensions), where the clustering structure is not clear when all dimensions are considered, the search for low-dimensional subspaces that discard irrelevant dimensions is a necessity to permit accurate cluster identification. In Chapters 4 and 5, we have seen that optimally projecting a set of datapoints $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{n}$, and using these projections to identify low-density cluster separators permits high-quality clustering results using one-dimensional projections of $\mathcal{X}$ only.

The approaches proposed in Chapters 4 and 5 locate projections that result in a cluster separator that intersects a region of minimal density, as proposed by Pavlidis et al. (2016). This approach can accurately identify clusters in a variety of real-world datasets. However, there are alternative projection techniques that have been applied in the literature which optimise different criteria to locate appropriate subspaces for clustering. For example, principal component analysis (PCA), as applied in the well-known principal direction divisive partitioning (PDDP) algorithm (Boley, 1998), and its extensions such as interval PDDP (i-PDDP) and density enhanced PDDP (dePDDP) (Tasoulis et al., 2010). Independent component analysis (ICA) has also been successfully applied to projective clustering problems in Saidi et al. (2004); Tasoulis et al. (2011). Both PCA and ICA have been shown to locate effective projections for clustering in applications such as gene expression clustering and text mining. Although all the projection techniques mentioned are capable of locating subspaces that permit accurate cluster separation, their computation becomes infeasible in large, high-dimensional datasets, even when using highly optimised linear algebra packages.

Random projection (RP) (Achlioptas, 2001; Dasgupta, 2000) has been proposed as a computationally inexpensive way to reduce dimensionality, which has a much lower computational cost than the aforementioned projection techniques. The use of RP is theoreti-

cally justified by the following lemma, which states that any set of n points may be projected into a space of dimension $r < \mathcal{O}(\epsilon^{-2} \log n)$ while preserving relative pairwise distances up to $\pm\epsilon$ for $0 < \epsilon < 1$,

Lemma 1. *[Johnson–Lindenstrauss lemma (Johnson and Lindenstrauss, 1984)] Given* $0 < \epsilon < 1$ *and an integer* n*, let* r *be a positive integer such that* $r \geqslant r_0 = \mathcal{O}(\epsilon^{-2} \log n)$*. For every set* $\{\mathbf{x}_i\}_{i=1}^n$ *of* n *points in* $\mathbb{R}^d$*, there exists* $g : \mathbb{R}^d \to \mathbb{R}^r$ *such that for all* $\mathbf{x}_i, \mathbf{x}_j$*,*

$$(1 - \epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2 \leqslant \|g(\mathbf{x}_i) - g(\mathbf{x}_j)\|^2 \leqslant (1 + \epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|^2.$$

This bound is pessimistic, and often pairwise distances are accurately preserved in a much lower dimensional subspace than suggested by the Johnson–Lindenstrauss lemma. Further motivation for the application of RP is that data generated from a number of high-dimensional distributions appears more Gaussian after being randomly projected into a lower-dimensional subspace (Diaconis and Freedman, 1984), and irregularly shaped clusters become more spherical (Dasgupta, 2000).

RP has been applied with success in a number of clustering applications, where high dimensionality and large numbers of observations are a common problem. Generally, these approaches seek a low-dimensional random subspace of $\mathcal{X}$, in which the clusters are identifiable using the chosen algorithm. For example, Avogadri and Valentini (2009) apply RP to reduce the dimensionality of gene expression data, before clustering using the fuzzy k-means algorithm (Bezdek, 2013). Bingham and Mannila (2001) and Goal et al. (2005) investigate the suitability of random subspaces for the task of facial recognition and text mining respectively. These two investigations show that the subspaces located through RP are capable of locating similar results to the subspaces obtained by PCA for these tasks, while offering a significant reduction in computational cost. Further, Tasoulis et al. (2012) have combined

RP with PCA to significantly reduce the computational cost of locating low-density cluster separators using univariate projections onto directions of maximal variability. This method applies the dePDDP algorithm (Tasoulis et al., 2010) on the projections of $\mathcal{X}$ into a random subspace, which avoids locating the principal components of the original set of observations. The authors show that this can significantly reduce the computational cost of locating an effective cluster separator, while maintaining competitive performance compared with clustering the original observations. In addition, Fern and Brodley (2003) show how the potential diversity of clustering results from model-based clustering in different random subspaces may be combined by ensemble clustering (Strehl and Ghosh, 2002) to improve the accuracy and stability of RP approaches.

Since the low-density separation algorithms proposed in this thesis partition clusters in one-dimensional subspaces of $\mathcal{X}$ only, our aim is to locate univariate projections of $\mathcal{X}$, that are approximately optimal for cluster identification, in a computationally efficient manner. We propose to search over a finite collection of one-dimensional random subspaces, for the univariate projections of $\mathcal{X}$ that permit the highest-quality cluster separator. In later sections, we show that if the true cluster labels were known, thus permitting the definition of the most appropriate random subspace based on the clustering accuracy of a low-density separator computed in that space, then only a small number of random projections are required before a very high-quality bi-partition is located. However, in clustering, we cannot determine the suitability of a set of projections based on the resulting clustering performance. Therefore, we consider different optimality criteria to quantify the appropriateness of a set of random projections for cluster identification using low-density separators. These choices are discussed in later sections. The bi-partitions of $\mathcal{X}$ in these approximately optimal subspaces are combined in a divisive algorithm to locate a complete clustering, and

estimate the number of clusters.

Computing the projections of $\mathcal{X}$ onto a collection of randomly generated vectors only requires a single matrix multiplication, which is a very computationally efficient operation, that has a linear computational cost with respect to both the number of observations and dimensions in $\mathcal{X}$. Further, we can search over the same random subspaces at each level of the divisive algorithm, avoiding repeated generation of projection vectors, and subsequent computation of projections into the subspaces defined by them. This offers a significant computational advantage compared to the aforementioned projection techniques, that seek the optimal projection of each successive subset of $\mathcal{X}$ (with a polynomial computational cost) at each level of a divisive algorithm. Therefore, we find that seeking approximately optimal projections through RP can produce a complete clustering of $\mathcal{X}$ significantly faster than locating globally optimal one-dimensional subspaces through PCA, ICA and MDH. The proposed approach is restricted to linear cluster boundaries in the space of the original observations, so we lift this restriction by non-linearly mapping the observations into a feature space. In this case, the search space for an appropriate projection vector is determined by the number of observations, making the efficient computation of projections increasingly relevant for datasets with large numbers of observations.

The remainder of this chapter is organised as follows, Section 6.2 provides the methodology for the proposed approach. This begins with a formulation of linear low-density cluster separation. We then discuss possible projection techniques for locating one-dimensional subspaces that may be appropriate for low-density cluster separation, and consider how RP may be applied to locate projections that approximately optimise criteria that are related to the objectives of the alternative techniques discussed. Later, we present a divisive algorithm to combine the resulting low-density separators to produce a complete clustering of $\mathcal{X}$. Sec-

tion 6.3 begins by comparing the computational time required to locate bi-partitions, and complete clusterings of $\mathcal{X}$ using RP, PCA, ICA and MDH. Then we evaluate the performance of the partitions located through RP and the alternative projection techniques across simulated and real-world datasets with varying characteristics when $\mathcal{X}$ is the set of original observations. Section 6.4 then investigates the performance of the proposed RP approach compared to alternative projection methods when $\mathcal{X}$ is a set of feature vectors, that have been projected onto an n-dimensional orthonormal basis, allowing the identification of clusters which are not linearly separable in the original data space. The results of our experiments are summarised in Section 6.5. Finally, this work is concluded in Section 6.6.

6.2 Methodology

In this section, the methodology for the proposed approach is outlined. We begin by formulating the problem of bi-partitioning using low-density cluster separators. This formulation requires one-dimensional projections of the set of points to be clustered, $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{n}$, to evaluate the integrated density along a hyperplane separator. We therefore present possible approaches for the location of one-dimensional projections, which may be appropriate for cluster separation. We then consider the computationally efficient location of one-dimensional projections for clustering using RP. Finally, we propose a divisive procedure to combine the resulting bi-partitions to locate a complete clustering of $\mathcal{X}$.

Throughout this chapter, $\mathcal{X}$ may be the original observations which span d dimensions in the space of the original observations, or the set of non-linearly mapped feature vectors, which span n dimensions in the feature space that have been projected onto an orthonormal basis. We adopt this notation for brevity since, as discussed in Section 5.3, any meaningful projection vectors will lie within the span of the feature vectors $\{\phi(\mathbf{x}_i)\}_{i=1}^{n}$.

Hence, using the projections of the potentially infinite-dimensional feature vectors onto an n-dimensional orthonormal basis, spanning the same space as $\{\phi(\mathbf{x}_i)\}_{i=1}^n$, permits the location of the equivalent set of optimal univariate projections as directly computing the optimal one-dimensional subspace in the feature space.

6.2.1 Cluster Separation using One-Dimensional Projections

It is assumed throughout that $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ is a set of realisations of a continuous random variable X with continuous estimated probability density function $\hat{p}_{\mathbf{x}}$. We adopt the low-density separation formulation of the clustering problem, seeking low-density cluster boundaries, which partition but do not intersect high-density regions in $\hat{p}_{\mathbf{x}}$, associated with clusters. The evaluation of the integrated density along a cluster boundary is computationally intractable unless attention is restricted to linear separators (hyperplanes). However, in the case that $\mathcal{X}$ is a set of non-linearly mapped feature vectors, these linear separators correspond to non-linear separators of the original observations. The dense, linearly separable sets of $\mathcal{X}$, which may be separated by a low-density hyperplane are defined in Section 2.3, Definition 3.

As a consequence of applying Definition 3, the family of clusters $\mathbf{C}_1, ..., \mathbf{C}_k$ in $\mathcal{X}$ is linearly separable if there exists a hyperplane along which the maximum value of $\hat{p}_{\mathbf{x}}$ is at most $c \geqslant 0$, and which also separates at least one cluster from the rest of the data. This definition results in the clusters in $\mathcal{X}$, located by a low-density linear separator corresponding to dense clusters (Section 2.3, Definition 1), provided their convex hulls do not intersect. A collection of low-density linear separators is able to identify all the clusters $\mathbf{C}_1, \ldots, \mathbf{C}_k$ if $\mathcal{X}$ is dense and linearly clusterable (with respect to the density estimator $\hat{p}_{\mathbf{x}}$), as defined in Section 2.3, Definition 4.

Following Chapters 4 and 5, we define the density intersected by a hyperplane separator

$H(\mathbf{v}, b)$ with unit normal $\mathbf{v} \in \mathbb{S}^{d-1} = \{\mathbf{x} \in \mathbb{R}^d \mid \|\mathbf{x}\| = 1\}$ and displacement from the origin $b \in \mathbb{R}$ as,

$$\hat{I}(\mathbf{v}, b) = \frac{1}{n\sqrt{2\pi h^2}} \sum_{i=1}^{n} \exp\left\{ -\frac{(b - \mathbf{v}^\top \mathbf{x}_i)^2}{2h^2} \right\} = \hat{p}_{\mathbf{v}^\top \mathbf{x}}(b). \tag{6.1}$$

where $\hat{p}_{\mathbf{v}^\top \mathbf{x}}(b)$ is the estimated density of the univariate projections of $\mathcal{X}$ onto $\mathbf{v}$, evaluated at b and h is the bandwidth of the density estimate used in both $\hat{p}_{\mathbf{x}}$ and $\hat{p}_{\mathbf{v}^\top \mathbf{x}}$. To locate a low-density cluster separator, we require the determination an appropriate projection direction $\mathbf{v}$, along which the clusters are identifiable. Possible projection techniques for this are discussed in Sections 6.2.2 and 6.2.3. Thereafter, we seek to partition high-density regions in $\hat{p}_{\mathbf{x}}$ using the projections of $\mathcal{X}$ onto $\mathbf{v}$ to determine a suitable value for b with low values of $\hat{I}(\mathbf{v}, b)$. This is considered in Section 6.2.4.

6.2.2 Optimal Projections

In this section we outline methods for locating one-dimensional projections of $\mathcal{X}$ which globally optimise criteria that may be indicative of appropriate projection directions for cluster detection. The methods considered are principal component analysis (PCA), independent component analysis (ICA) and the minimum density hyperplane (MDH).

Principal Component Analysis (PCA)

In PCA (formulated in Section 2.2.1), the one-dimensional projection vector located retains the maximal variability in $\mathcal{X}$ and minimises the reconstruction error. This is an intuitive approach provided the clusters in $\mathcal{X}$ are not heavily elongated, since it is likely that the clusters will be separable along the direction that the data are most dispersed. This was extended to allow the computation of directions of maximum variability in feature spaces by kernel

principal component analysis (KPCA) (Schölkopf et al., 1998). The formulation of KPCA is presented in Section 5.3, so we omit this here. If $\mathcal{X}$ is the set of feature vectors, projected onto an n-dimensional orthonormal basis, spanning the same space, then the univariate projection of the vectors in $\mathcal{X}$ onto their first linear principal component is equivalent to the univariate projection of the feature vectors, onto the first kernel principal component.

Independent Component Analysis (ICA)

ICA (Hyvärinen et al., 2004) projects $\mathcal{X}$ such that the dimensions of the projected data are independent and non-Gaussian. This originates from signal processing, allowing the separation of multivariate signals into additive subcomponents. The independence between the components located may be specified by minimising the mutual information (Bell and Sejnowski, 1995) or by minimising the Gaussianity (Cardoso and Souloumiac, 1993, 1996). Throughout this chapter, we use the Joint Approximation Diagonalization of Eigen-matrices (JADE) algorithm for ICA (Cardoso and Souloumiac, 1993, 1996), which adopts the latter approach. This relies on the Lindeberg condition, which states that for a set of independent random variables X_i (which are not necessarily Gaussian) with means and variances μ_i and σ_i^2 respectively, a linear combination of these random variables tends to a normal distribution as the number of terms in the linear combination tends to infinity, conditional on none of the σ_i^2 dominating the variances and sufficiently weak dependence between the variables. Therefore, locating axes with minimal Gaussianity equates to recovering independent components. For our purposes, we only require a one-dimensional projection and hence, we only consider the first independent component.

Excess kurtosis, may be thought of as a measure of non-Gaussianity, with higher absolute values indicating a distribution that is further from a unimodal Gaussian distribution. Therefore, the first independent component located by the JADE algorithm is equivalent to

locating the projection vector $\mathbf{v}$ with maximal absolute excess kurtosis in the projections of $\mathcal{X}$ onto $\mathbf{v}$ (Roberts and Everson, 2001),

$$\mathcal{K}(\mathbf{v}^\top \mathbf{x}) = \left| \frac{\frac{1}{n}\sum_{i=1}^{n}(\mathbf{v}^\top \mathbf{x}_i - \hat{\mu}_\mathbf{v})^4}{\hat{\sigma}_\mathbf{v}^4} - 3 \right|$$

where $\hat{\mu}_\mathbf{v}$ and $\hat{\sigma}_\mathbf{v}$ are the mean and standard deviation of the projections of $\mathcal{X}$ onto $\mathbf{v}$ respectively. Peña and Prieto (2001) show that for datasets with clear clustering structures, univariate projection directions with minimal kurtosis have maximal bi-modality in the distribution of the projections, while minimising the effect of outliers. In datasets with well separated clusters, it is likely that $\mathcal{K}(\mathbf{v}^\top \mathbf{x})$ is maximised by the projection direction with the most negative excess kurtosis. Therefore, ICA often locates a projection direction with a bi-modal structure. However, maximising the absolute value of excess kurtosis sometimes results in the location of projections that have a high positive excess kurtosis, and are highly unimodal. Such directions are not suitable for low-density separation, so ICA may locate inappropriate projections in some datasets.

Bach and Jordan (2002) extended ICA to kernel defined feature spaces where independence is defined using contrast functions based on canonical correlations of the feature vectors. This measure of independence is related to mutual information and kurtosis, but the algorithms presented are not directly comparable to the JADE algorithm in the data space. Therefore, for consistency, when implementing ICA on the non-linearly mapped feature vectors, we use the JADE algorithm on the n-dimensional projections of the feature vectors onto an orthonormal basis of the feature space.

Minimum Density Hyperplane (MDH)

The MDH (Pavlidis et al., 2016) is a projection pursuit approach that seeks the optimal one-dimensional projection direction for low-density cluster separation. This method seeks to partition dense, linearly separable clusters by locating linear cluster boundaries, which intersect regions of minimal density in $\hat{p}_{\mathbf{x}}$, by minimising $\hat{I}(\mathbf{v}, b)$ as defined in Eq. (6.1), subject to sensible constraints on b. This is discussed in detail in Section 4.2.1. This approach was extended to feature spaces in Chapter 5 by the kernel minimum density hyperplane (KMDH). The optimal univariate projections of the feature vectors that permit the hyperplane separator which minimises $\hat{I}(\mathbf{v}, b)$ can be computed directly using the kernel matrix, or equivalently using the n-dimensional projections of the feature vectors onto an n-dimensional orthonormal basis spanned by them.

6.2.3 Random Projection (RP)

As an alternative to the projection techniques discussed above, we consider RP (Achlioptas, 2001; Dasgupta, 2000) to locate approximately optimal projections for clustering. In RP, the set $\mathcal{X}$ is projected into a random subspace of dimension r by a random orthogonal matrix $\mathbf{R} = [r_{ij}] \in \mathbb{R}^{d \times r}$. There is no universally adopted approach for the construction of $\mathbf{R}$. However, for Lemma 1 to hold, the entries of $\mathbf{R}$, r_{ij} for $i = 1, \ldots, d,\ j = 1, \ldots, r$ must satisfy $\mathbb{E}(r_{ij}) = 0,\ \mathrm{Var}(r_{ij}) = 1$. Therefore, the following three examples are attractive,

1. Bernoulli random projections,

$$\mathbf{R}^{\star} = \frac{1}{\sqrt{r}}[r_{ij}]\ ,\ \mathbb{P}(r_{ij} = p) = \begin{cases} 1/2 & ,\ p = -1 \\ 1/2 & ,\ p = 1 \end{cases}$$

2. Achlioptas random projections (Achlioptas, 2001),

$$\mathbf{R}^\star = \frac{1}{\sqrt{r}}[r_{ij}] \ , \ \mathbb{P}(r_{ij} = p) = \begin{cases} 1/6 & , \ p = -\sqrt{3} \\ 2/3 & , \ p = 0 \\ 1/6 & , \ p = \sqrt{3} \end{cases}$$

3. Normal random projections (Bingham and Mannila, 2001),

$$\mathbf{R}^\star = \frac{1}{\sqrt{r}}[r_{ij}] \ , \ r_{ij} \sim N(0,1).$$

The projections of $\mathcal{X}$ into a the random subspace defined by $\mathbf{R}$ are given by

$$\mathcal{X}^r = \{\mathbf{R}^\top \mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^r.$$

The orthogonalisation of $\mathbf{R}$ is computationally expensive, however, as the dimensionality of $\mathcal{X}$ increases, a set of random projections become asymptotically orthogonal to each other (Hecht-Nielsen, 1994). Therefore, if the dimensionality of $\mathcal{X}$ is sufficiently high, this additional cost may be avoided. Without this orthogonalisation, it is trivial to see that the computation of the projections of $\mathcal{X}$ into the subspace defined by $\mathbf{R}$ only requires a single matrix multiplication. This efficient linear operation is highly attractive compared to the optimisation techniques required for PCA, ICA and MDH, which have a quadratic computational cost, as discussed in Section 6.2.7.

The computational efficiency of computing projections, combined with the theoretical justification, and successful applications to clustering problems discussed in Section 6.1 make RP an attractive dimensionality reduction technique. These results motivate our consideration of RP to locate projections of $\mathcal{X}$ that permit low-density cluster separators, that are related to the globally optimal projection vectors located by the techniques discussed in Section 6.2.2.

We propose to use RP to search for approximately optimal one-dimensional subspaces for

clustering. We search over a collection of univariate projections of $\mathcal{X}$ onto multiple random vectors, given by the columns of $\mathbf{X}^r = \mathbf{X} \cdot \mathbf{R}$ where $\mathbf{X}$ is the $n \times d$ data matrix associated with $\mathcal{X}$. We then partition $\mathcal{X}$ using the one-dimensional random projections which best satisfy a specified optimality criterion, indicating the suitability of a given set of projections for cluster identification. The partitions located through our RP approach approximate the partitions from the projection vectors in Section 6.2.2, which globally optimise related criteria over all possible one-dimensional subspaces of $\mathcal{X}$.

Since we consider the projections onto the random vectors independently, we do not require $\mathbf{R}$ to be orthogonal. However, to appropriately sample the search space, we do require the random vectors stored columnwise in $\mathbf{R} = [\mathbf{r}_j]$ to be sampled uniformly from the unit sphere, $\mathbf{r}_j \in \mathbb{S}^{d-1}$ for $j = 1, \ldots, r$. In the original data space, this can be done by generating from a multivariate $N(0, 1)$ distribution and then normalising the vector to have unit length (Rubinstein, 1982).

However, it is not possible to guarantee this when generating vectors directly in the feature space. This results from being unable to define a vector, $\mathbf{w}$ in the theoretically infinite-dimensional space, and instead relying upon the generation of n-dimensional dual vectors $\boldsymbol{\alpha}$, such that $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \phi(\mathbf{x}_i)$ where $\phi(\mathbf{x}_i)$ is the mapped feature vector of the ith observation. To our knowledge, there is no way to generate $\boldsymbol{\alpha}$ such that $\mathbf{w}$ is uniformly sampled from the unit sphere in the feature space. However, since we restrict attention to the n-dimensional space spanned by the feature vectors, and can equivalently locate low-density separators of the projections of the feature vectors onto an n-dimensional orthonormal basis, we do not generate vectors directly in the potentially infinite-dimensional feature space. Instead, treating the n-dimensional projections of the feature vectors as a set of mapped observations, allows the generation of random vectors from the n-dimensional unit sphere as

above.

6.2.4 Divisive Clustering with Low-Density Separators

To produce a complete clustering of $\mathcal{X}$, we combine low-density cluster separators located in one-dimensional subspaces computed by the projection techniques considered above in a divisive, hierarchical algorithm. Given a projection vector computed by any of the methods discussed in Sections 6.2.2 - 6.2.3, the divisive procedure requires three decisions:

1. Which cluster to split at each level of the hierarchy (selection rule);
2. How to split that cluster (splitting rule);
3. When to terminate (stopping rule).

Selection Rule

At each level of the hierarchy, we select the cluster which contains the set of univariate projections that optimise (or approximately optimise) our specified criterion. Let $\mathcal{X}_{C_j} \subset \mathcal{X}$ for $j = 1, \ldots, k$ denote the subsets of $\mathcal{X}$ assigned to each of the k clusters located so far, with associated data matrices $\mathbf{X}_{C_j}$. Further, let $f(\cdot)$ be a function of the univariate projections of the observations in $\mathcal{X}_{C_j}$ onto a vector $\mathbf{v}$, computed by $\mathbf{X}_{C_j} \cdot \mathbf{v}$, which we are seeking to maximise (it is trivial to consider a minimisation problem instead). Our choices for $f(\cdot)$ are discussed in Section 6.2.6. If we globally optimise $f(\cdot)$ for each $\mathcal{X}_{C_j}$, we select the cluster which solves the following problem,

$$j^\star = \underset{j \in \{1,\ldots,k\}}{\arg\max} \left\{ \max_{\mathbf{v} \in \mathbb{S}^{d-1}} f(\mathbf{X}_{C_j} \cdot \mathbf{v}) \right\}.$$

If we use RP to locate a projection vector which only approximately optimises $f(\cdot)$ for each $\mathcal{X}_{C_j}$, we select the cluster which solves,

$$j^\star = \underset{j\in\{1,\ldots,k\}}{\arg\max} \left\{ \max_{i\in\{1,\ldots,r\}} f(\mathbf{X}_{C_j} \cdot \mathbf{r}_i) \right\} \tag{6.2}$$

where $\mathbf{r}_i = (r_{i,1}, \ldots, r_{i,d})$ is the ith random vector stored in $\mathbf{R}$ and r is the total number of random vectors over which we search for an appropriate projection direction for clustering. We then denote the subset of observations currently assigned the selected cluster $\mathcal{X}_{C_{j^\star}}$ by $\mathcal{X}_C$.

Splitting Rule

To bi-partition $\mathcal{X}_C$, we seek a separating hyperplane that intersects a region of low-density in $\hat{p}_{\mathbf{x}}$ (quantified by the integrated density in Eq. (6.1)) while separating high-density clusters. This is done using the estimated density of the univariate projections of $\mathcal{X}_C$ onto the vector which optimises (or approximately optimises) our specified optimality criterion $f(\cdot)$. For the projection techniques which globally optimise $f(\cdot)$, the selected univariate projections are given by

$$\mathbf{p}_C = \mathbf{X}_C \cdot \mathbf{v}^\star \tag{6.3}$$

$$\mathbf{v}^\star = \underset{\mathbf{v}\in\mathbb{S}^{d-1}}{\arg\max}\, f(\mathbf{X}_C \cdot \mathbf{v}), \tag{6.4}$$

where $\mathbf{X}_C$ is the data matrix associated with $\mathcal{X}_C$, whereas for our RP approach, the selected univariate projections are given by

$$\mathbf{p}_C = \mathbf{X}_C \cdot \mathbf{r}^\star \tag{6.5}$$

$$\mathbf{r}^\star = \underset{\mathbf{r}_i \,;\, i \in \{1,\ldots,r\}}{\arg\max} \; f(\mathbf{X}_C \cdot \mathbf{r}_i). \tag{6.6}$$

To ensure that the low-density separators obtained partition high-density regions associated with clusters, we do not simply select b to minimise the estimated density of the selected univariate projections $\hat{p}_{\mathbf{p}_C}$, since this result in a hyperplane that lies in the tails of $\hat{p}_{\mathbf{x}}$. Instead, we determine b to be the minimiser of $\hat{p}_{\mathbf{p}_C}$ which maximises the relative depth criterion,

$$b^\star = \underset{b \in \mathbb{R}}{\arg\max}\, \text{RelativeDepth}(\mathbf{p}_C, b) \tag{6.7}$$

$$\text{RelativeDepth}(\mathbf{p}_C, b) = \frac{\min\left\{\hat{p}_{\mathbf{p}_C}(m_l), \hat{p}_{\mathbf{p}_C}(m_r)\right\} - \hat{p}_{\mathbf{p}_C}(b)}{\hat{p}_{\mathbf{p}_C}(b)}, \tag{6.8}$$

where m_l and m_r are the locations of the two largest modes of $\hat{p}_{\mathbf{p}_C}$ to the left and right of b respectively. By convention, if there is no mode either to the left or the right of b the relative depth is zero. This choice of b results in a hyperplane separator of $\mathcal{X}_C$ which intersects a region of low density, and assigns observations in high-density regions of $\hat{p}_{\mathbf{x}}$ to different clusters.

Stopping Rule

Our choice of cluster definition and splitting rule lends itself to an intuitive stopping rule that terminates the divisive procedure when it is not possible to locate a sufficiently low-density separator that partitions high-density regions in $\hat{p}_{\mathbf{x}}$ using the selected set of projec-

tions $\mathbf{p}_C$. Therefore, the number of clusters may be estimated automatically. This stopping rule considers the relative depth of the estimated density of a set of univariate projections, $\mathbf{p}_C$ to assess their suitability for low-density cluster separation. If $\hat{p}_{\mathbf{p}_C}$ is not multimodal (or equivalently has a sufficiently small relative depth), this indicates that it is not possible to locate a hyperplane separator, with the normal vector selected, that intersects a region of sufficiently low-density to indicate an appropriate cluster boundary that separates high-density clusters. We propose to set a threshold on the value of the relative depth of $\hat{p}_{\mathbf{p}_C}$, which determines if a suitable low-density separator is permitted using these projections.

We do not simply test for the presence of more than one mode in the estimated density of $\mathbf{p}_C$, since we do not want to accept a bi-partition at a minimiser in $\hat{p}_{\mathbf{p}_C}$ between two small modes, as such separators are unlikely to partition high-density clusters. This problem is particularly relevant for approaches that actively seek projection directions with a multimodal estimated projected density. We propose to test for multimodality in $\hat{p}_{\mathbf{p}_C}$ by comparing the relative depth of this density to the Monte-Carlo estimated quantiles of the relative depth of the estimated density of a sample from a null unimodal reference distribution. For this we use the uniform distribution, since this is the standard choice for modality testing (Hartigan and Hartigan, 1985; Hartigan, 1977). Our specific choices for this procedure are discussed in Section 6.4.1.

6.2.5 Combining RP Trees by Ensemble Clustering

Topchy et al. (2005) discuss the combination of multiple *weak partitions* located by hyperplane separators that arise from projecting the data into random one-dimensional subspaces through ensemble (consensus) clustering (Strehl and Ghosh, 2002; Dimitriadou et al., 2002). Empirical studies in this work indicate that combining the information from varied input partitions by an ensemble clustering can dramatically improve the clustering

performance compared to a single cluster separator. Therefore, we consider generating multiple clusterings of $\mathcal{X}$ using hierarchies of low-density separators from the proposed RP approaches, that search over different collections of random projections, and combine these partitions via an ensemble clustering. This final clustering incorporates information from all of the input partitions, and in our experiments, often produces a clustering of higher quality than the average performance from using a single hierarchy.

We apply the ensemble clustering approach of Dimitriadou et al. (2002) (implemented in the `clue` package for `R`), which takes a collection of m input cluster assignment matrices $\mathbf{M}^1, \ldots, \mathbf{M}^m$, and returns the fuzzy clustering assignment matrix $\mathbf{P}$, such that each entry p_{il} is the probability that observation $\mathbf{x}_i$ belongs to cluster l for $i = 1, \ldots, n$ and $l = 1, \ldots, k$. Since each partition obtained through the divisive RP methods is a hard clustering, we have $M^j_{il} = 1$ if observation $\mathbf{x}_i$ is assigned to cluster l in the j-th input clustering and equal to zero otherwise. If the input partitions contain different numbers of clusters, any input cluster assignment matrices with fewer columns than the maximum number of clusters located in the input clusterings, are augmented with columns of zeros to ensure that the input matrices have the same dimensionality. Further, the columns of the input assignment matrices may be trivially permuted so that the input clusterings are invariant to relabelling the clusters. Given the processed collection of input matrices, $\mathbf{M}^1, \ldots, \mathbf{M}^m, \mathbf{M}^j \in \{0,1\}^{n \times k}\ \forall j = 1, \ldots, m$, The fuzzy clustering assignment matrix $\mathbf{P}$ is the matrix whose rows $\mathbf{p}_i$ solve,

$$\min_{\mathbf{P}} \left(\min_{\mathbf{\Pi}} \left\{ \frac{1}{m} \sum_{j=1}^{m} \sum_{i=1}^{n} \| \mathbf{p}_i - \Pi^j(\mathbf{m}^j_i) \|_2^2 \right\} \right)$$

where $\mathbf{m}^j_i$ is the ith row of the input cluster assignment matrix $\mathbf{M}^j$ and Π^j is a function that permutes the columns of $\mathbf{M}^j$. Therefore, the fuzzy ensemble cluster assignment matrix $\mathbf{P}$

minimises the average squared Euclidean distance to the input clusterings. To locate a final hard clustering of $\mathcal{X}$, we take the final partition $\boldsymbol{\pi}^* \in \mathbb{N}^n$ such that $\pi_i^* = \arg\max \mathbf{p}_i$ for $i = 1, \ldots, n$.

6.2.6 Optimality Criteria to Select Random Projections

The optimality criteria which we consider for the selection of a set of univariate random projections that may be suitable for the location of a low-density separator are:

1. Maximum relative depth in the estimated density of the univariate projections. This optimality criterion retains projections exhibiting a strong multimodal structure in their estimated density, with a low minimiser between two large modes. Hence, this is consistent with the objective of locating low-density separators that assign observations in high-density regions around the modes of $\hat{p}_{\mathbf{x}}$ to different clusters. Therefore, this optimality criterion is related to the objective of MDH.

2. Maximum dip statistic (Hartigan and Hartigan, 1985) in the estimated density of the univariate projections. Like the relative depth, this criterion also considers the modality of the estimated projected density, and favours projections with a strongly multimodal structure. This was applied by Krause and Liebscher (2005) as an objective for projection pursuit clustering. Unlike the maximum relative depth criterion, the dip statistic only considers the extent to which the estimated density of the univariate projections is multimodal. Therefore, this criterion can permit projections that have a strongly multimodal distribution, but do not necessarily have a low minimiser between these modes.

3. Maximum variance in the univariate projections. Although there is no guarantee that directions of high variability are suitable for cluster detection (Kriegel et al., 2009), if the clusters are not heavily elongated, it is likely that projections which are highly dispersed are separable by a region of low density (Boley, 1998; Tasoulis et al., 2010). This optimality criterion is consistent with the objective of PCA.

4. Minimum kurtosis, which retains univariate projections with minimal Gaussianity, so as to avoid projections with a clear unimodal Gaussian density. Peña and Prieto (2001) show that locating univariate projections with minimal kurtosis corresponds to maximising the bi-modality in the estimated density of the projections. As such, this optimality criterion should permit a cluster separator that separates regions of high-density in $\hat{p}_x$. This is associated with the objective of ICA. However, since ICA maximises the absolute value of the excess kurtosis, it is possible that ICA will locate

a projection direction with a very slender non-Gaussian distribution, while selecting projections with the most negative excess kurtosis will always favour projections with a highly dispersed uniform-type or bi-modal distribution. Therefore, projections which minimise the kurtosis are arguably more consistent with locating cluster separators than projections that maximise the absolute excess kurtosis. We expect cases where RP with this optimality criterion and ICA locate drastically different projections to be rare in datasets with a clear clustering structure.

6.2.7 Computational Complexity

In this section, we discuss the computational complexity of locating hierarchies of low-density separators by the proposed RP approaches, and the alternative projection techniques considered. In our experiments, when extending these techniques to locate appropriate projections of feature vectors, we use the n-dimensional projections of the feature vectors onto an orthonormal basis. This requires the construction of the kernel matrix (for which we use the Gaussian kernel), with cost $\mathcal{O}(n^2 d)$. For the construction of an n-dimensional orthonormal basis of the feature vectors, we use KPCA, with computational cost $\mathcal{O}(n^3)$. Finally, the projections of the feature vectors onto the kernel principal components incurs a cost of $\mathcal{O}(n^3)$. This is the same for all the projection techniques considered, and is only computed once.

Hereafter, we consider the cost of locating the optimal projections of $\mathcal{X}$ with n observations and d dimensions, either as the original d-dimensional set of observations or the n-dimensional projections of the feature vectors. First we consider the computational complexity of locating an optimal (or approximately optimal) univariate projection of $\mathcal{X}$. The first principal component of $\mathcal{X}$ may be located by an iterative procedure such as the power method (Kuczyński and Woźniakowski, 1992), avoiding the computation and complete eigen-decomposition of the covariance matrix. The power method has a cost of $\mathcal{O}(nd^2)$

per iteration. The JADE algorithm for ICA iteratively computes the projection vector with minimal absolute excess kurtosis, with a computational cost of $\mathcal{O}(d^2 + n)$ per iteration. The location of the MDH is also an iterative procedure. For each iteration, $\mathcal{X}$ is projected onto $\mathbf{v}$ with computational cost $\mathcal{O}(nd)$, and then $\hat{p}_{\mathbf{v}^\top \mathbf{x}}$ is constructed at m points using the fast Gauss transform (Morariu et al., 2009), at a cost of $\mathcal{O}(m + n)$. Locating the minimiser of $\hat{p}_{\mathbf{v}^\top \mathbf{x}}$ to accuracy ϵ requires $\mathcal{O}(-\log_2 \epsilon)$ iterations. The subsequent update of $\mathbf{v}$ by BFGS requires a single gradient evaluation, with a cost of $\mathcal{O}(d^2 + nd)$. Therefore, the overall computational complexity of locating the MDH is $\mathcal{O}(d^2 + nd)$.

For an approximately optimal projection, located using RP, it is necessary to compute the projections of $\mathcal{X}$ onto the matrix of r random vectors, with cost $\mathcal{O}(ndr)$. This is the most significant cost for the proposed RP approach, so dominates the computational complexity. It is worth noting that this is a single multiplication, and not an iterative procedure, such as those required for the optimal projection techniques considered. For each of the r random univariate projections, it is necessary to compute the value of the optimality criterion of interest. For the maximum relative depth and maximum dip statistic criteria, this requires the construction of the estimated density of the projections at m points, and this has cost $\mathcal{O}(n + m)$. The maxima and minima in these densities can then be located to accuracy ϵ with cost $\mathcal{O}(-\log_2 \epsilon)$. Meanwhile, the maximum variance and minimum kurtosis criteria have a cost of $\mathcal{O}(n)$.

Once the projections of $\mathcal{X}$ onto the selected projection vector have been computed, the subsequent bi-partition of the projections requires the construction and minimisation of a single univariate density estimate with cost $\mathcal{O}(n + m)$ and $\mathcal{O}(-\log_2 \epsilon)$ respectively. Except for the computation of the projections of $\mathcal{X}$ onto the random vectors in the RP approach, all the above operations are performed at each level of the hierarchy. Locating op-

timal projections by the iterative procedures required for PCA, ICA and MDH becomes computationally expensive for very large and high-dimensional datasets, and re-computing this at each level of the hierarchy increases the computational time required further, making the proposed RP approach very attractive. We investigate the computational times to locate bi-partitions and divisive clusterings using the projection techniques considered in Section 6.3.2. For a representative real-world dataset, the location of a cluster hierarchy took approximately 30, 15 and 20 minutes for MDH, PCA and ICA respectively. Meanwhile, locating a cluster hierarchy with 1,000 random projections took approximately 5 minutes.

6.2.8 Notation for RP Approaches

In Sections 6.3.2 - 6.3.4 and 6.4.2, we use the following notation to refer to the RP approaches using varying numbers of random projections and different optimality criteria. RP-depth-r, RP-dip-r, RP-var-r and RP-kur-r correspond to locating a single hierarchy using a fixed collection of r random projections and selecting the set of univariate projections with maximum relative depth, maximum dip statistic, maximum variance and minimum kurtosis respectively to partition $\mathcal{X}$ at each level of the hierarchy. When using multiple hierarchies, generated from different random projections, and combining the resulting partitions with an ensemble clustering, we use the notation RP-depth-r-E-m, RP-dip-r-E-m, RP-var-r-E-m and RP-kur-r-E-m. These refer to combining m hierarchies, each of which use a collection of r random projections to search for the set of univariate projections with maximum relative depth, maximum dip statistic, maximum variance and minimum kurtosis respectively to partition $\mathcal{X}$ at each level of the hierarchy.

6.3 Experimental Results using Original Observations

In this section, we conduct an empirical analysis of the proposed divisive RP approach to clustering, when $\mathcal{X}$ is the original set of d-dimensional observations across high-dimensional simulated and real-world datasets with varying characteristics. We consider different numbers of random projections to search for an approximately optimal set of univariate projections for clustering, as well as investigating the suitability of the different optimality criteria suggested in Section 6.2.6 to quantify the appropriateness of a set of projections for cluster detection. We begin with a run time analysis to assess the computational saving of using RP with increasing numbers of observations and dimensions in $\mathcal{X}$. Later, we evaluate the clustering performance of the RP approach over simulated and real-world datasets. The run time and performance of low-density separators located with the proposed RP approach is compared to low-density cluster separators computed by PCA, ICA and MDH. For the simulated datasets, we found that the randomness in the dataset dominated any randomness in the clustering algorithms, so only a single run of each algorithm is included. For the real datasets, the RP approach was run 30 times, each with a different set of random projection vectors. For large numbers of random projections, the variability in clustering performance was typically low over different sets of random projections, and the NMI between partitions arising from different random projections was high (approximately 0.8 - 0.9).

For the performance evaluation, we also include the clustering performance of k-means++ (Arthur and Vassilvitskii, 2007), where the number of clusters is estimated using the Gap statistic (Tibshirani et al., 2001) as a standard benchmark, and to assess the performance of clustering using low-density separators by comparison to an alternative approach. The computational cost of evaluating the Gap statistic for large datasets meant that this method could not run within four weeks on a high performance computing cluster, for some of the

datasets. In this case, this method is omitted in the performance evaluation. The computational cost of the Gap statistic also made it infeasible to include k-means++ in the run time analysis.

6.3.1 Details of Implementation

Parameter Settings

As for any density-based approach, the choice of bandwidth used to construct the kernel density estimate of the univariate projections affects the performance of our approach. For the computation of the estimated density of the projections located by all projection techniques, except MDH, we use the standard rule of $h = \hat{\sigma}_{\mathbf{v}}(4/(3n))^{1/5}$ where $\sigma_{\mathbf{v}}$ is the standard deviation of the projections onto $\mathbf{v}$. Since when $\hat{p}_{\mathbf{x}}$ is multimodal, MDH almost always locates projections with a multimodal estimated density, we use $h = 0.9\hat{\sigma}n^{-1/5}$ where $\hat{\sigma}$ is a fixed parameter. Silverman (1986) recommend this rule as the optimal choice for multimodal densities. We select $\hat{\sigma}$ to be the standard deviation of the projections of the observations in the cluster of interest along their first principal component. For all other parameters for MDH, we take the same approach as given in Section 4.4.1.

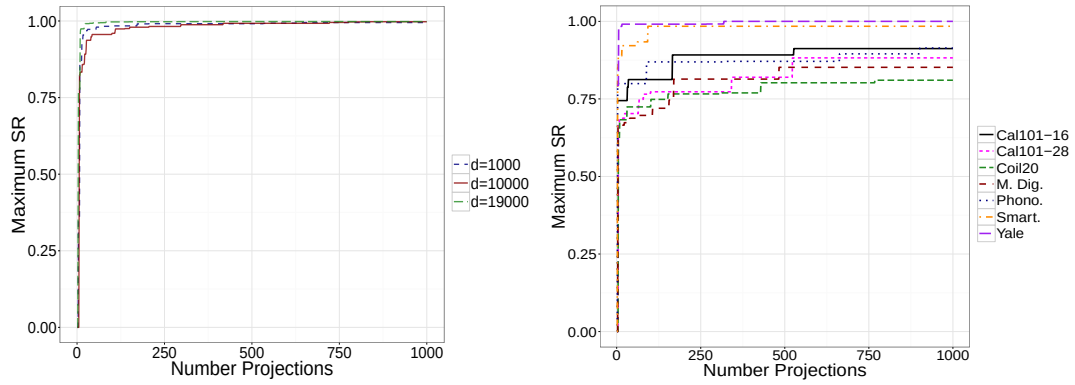
To obtain the estimated quantiles of the relative depth from the null distribution for our stopping rule proposed in Section 6.2.4, we use Monte-Carlo simulation with 1,000 null samples, each with the same number of observations as $\mathcal{X}$. We experimented with fixing this relative depth threshold at the start of the divisive procedure, and re-calculating this at each level, using null samples with the same number of observations as the cluster of interest, thus accounting for fewer observations per cluster at lower levels of the hierarchy (as applied in Chapters 4 and 5). For the experiments in this chapter, re-calculating this threshold at each level did not greatly improve clustering performance in the majority of cases,

and added extra computational cost, so we took the approach of a fixed threshold. For this threshold, we chose the 0.975 quantile of the relative depth of the estimated density of 1,000 uniform samples, each of size n although, we found that any threshold above the 0.9 quantile generally rejected poor-quality separators. This stopping rule was employed for all competing projection techniques, as well as our RP approaches to ensure that any difference in performance is due to the choice of projection vector and not different stopping rules.

Finally, for the RP approach with an ensemble clustering, we used 30 trees as input partitions. In our experiments, this was sufficient to locate a diverse set of input partitions, and offered a fair trade off between clustering performance of the ensemble and computational time. As in Chapters 4 and 5, we evaluate clustering performance with normalised mutual information (NMI) (Strehl and Ghosh, 2002). Alternative performance measures did not alter the relative performance of the competing approaches.

Number of Random Projections

For the proposed RP approach, we experimented using varying numbers of random projections over which to search for an appropriate cluster separator. Figure 6.1 shows the increase in clustering performance with an increasing number of random projections, for a bi-partition of the simulated datasets and real-world datasets which we use in Sections 6.3.3 and 6.3.4. The performance measure used here is the success ratio (SR) (Pavlidis et al., 2016), which is appropriate for assessing the quality of a bi-partition of a dataset with an arbitrary number of clusters. SR takes values in the range $[0, 1]$ with a value of 1 indicating that at least one cluster has been completely separated from the rest of the data. The simulated datasets used to produce Figure 6.1(a) were all generated from a Gaussian mixture model with 30 components (clusters). Results for different numbers of clusters were very similar so are omitted. The real datasets contain varying numbers of clusters and dimensions, which



(a) Simulated data (b) Real data

Figure 6.1: Increase in success ratio with increasing number of random projections for simulated datasets with 30 clusters in 1,000, 10,000 and 19,000 dimensions and real benchmark datasets summarised in Section 6.3.4.

are summarised in Table 6.1.

For each successive projection vector $\mathbf{r}_1, \ldots, \mathbf{r}_r$, the data were split (where possible) at the minimiser $b^\star$ of the estimated density of the projections of $\mathcal{X}$ onto $\mathbf{r}_j$ with maximum relative depth (as defined in Eqs. (6.7) - (6.8)), and the clustering performance of the bi-partition was recorded. If the performance for the partition $\boldsymbol{\pi}^j$ using the current set of random projections was better than for any previous set of projections, this was stored as follows,

$$S^j = \max\{\text{SuccessRatio}(\boldsymbol{\pi}^j, \boldsymbol{\pi}^\star), S^{j-1}\} \tag{6.9}$$

$$\pi_i^j = \begin{cases} 0 & \text{iff } \mathbf{r}_j^\top \mathbf{X} \leqslant b^\star \\ 1 & \text{iff } \mathbf{r}_j^\top \mathbf{X} > b^\star \end{cases}, \; i = 1, \ldots, n, \; j = 1, \ldots, r \tag{6.10}$$

where $\boldsymbol{\pi}^\star$ is the vector of true cluster labels and $\mathbf{X}$ is the data matrix associated with $\mathcal{X}$. Figure 6.1 indicates the rate of convergence to a high-quality separator if it is possible to select the most appropriate projection vector based on the quality of the resulting bi-partition.

For the datasets considered, a high-quality bi-partition was located with only a small number of random projections, and the improvement in performance when searching

over greater than 1,000 projections became negligible. Therefore, in Sections 6.3.3, 6.3.4 and 6.4.2, when computing complete cluster hierarchies using RP with the four optimality criteria considered, we experimented using 100, 500 and 1,000 random projections. The performance of the RP approaches using 500 random projections was always between the two more extreme cases so these are omitted for brevity.

6.3.2 Run Time Analysis

In this section we assess the computational time to produce a bi-partition and a divisive clustering of $\mathcal{X}$ using low-density separators located using our RP approach, PCA, ICA and MDH. All of the methods were coded using `R` and for computing projections with PCA and ICA we used the optimised `RSpectra` and `MASS` packages respectively. For each dataset, we considered using 100, 500 and 1,000 random projections, and selected the most appropriate projection using the relative depth criterion. The choice of optimality criterion did not noticeably affect the run time for RP, so we omit the results for alternative criteria. For the ensemble methods, the times quoted include the computation of 30 input cluster hierarchies and the subsequent ensemble clustering. In this section, the data were simulated from a d-dimensional Gaussian mixture model, with k very well separated components (clusters). This ensured that all the projection techniques were able to locate subspaces in which all the clusters were clearly identifiable. Therefore, the difference in run time is due to the cost of locating the projection vector, not as a result of different numbers of splits in the hierarchy.

Figure 6.2 provides the median CPU time in seconds, over 30 replications, to produce a single, bi-partition using low-density separators computed through RP, PCA, ICA and MDH with increasing numbers of observations and dimensionality in $\mathcal{X}$. Notice that we plotted the log-time due to the very high cost of PCA, ICA and MDH in high dimensions.

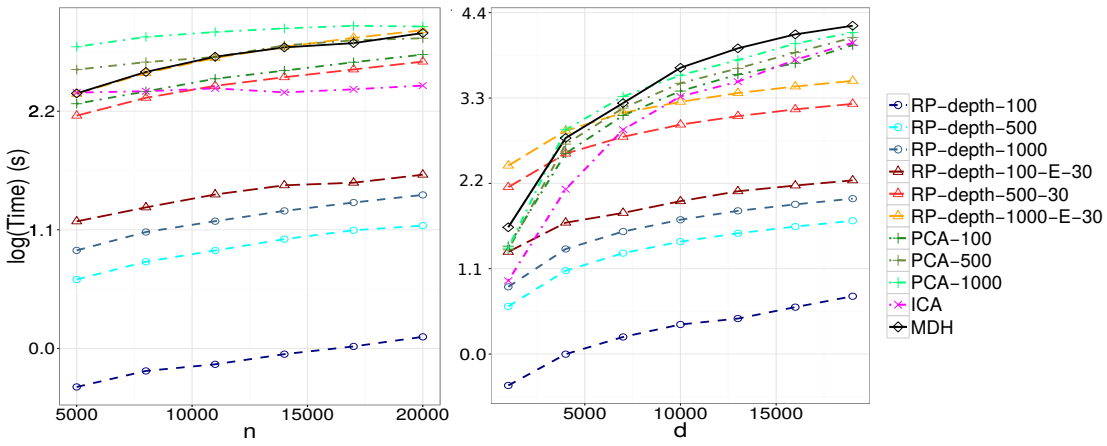


Figure 6.2: CPU time for a binary split with increasing numbers of observations and dimensionality for RP, PCA, ICA and MDH.

The datasets used to produce this figure consisted of two clusters of equal size. For increasing numbers of observations, dimensionality was fixed to $d = 5,000$, while for increasing dimensionality, the number of observations was fixed to $n = 20,000$. For PCA, we recorded the time to compute the same number of principal components as random projections, whereas the computational cost of ICA meant it was only feasible to compute the first component.

Figure 6.2 shows that all the projection techniques have a linear computational cost in the number of observations in $\mathcal{X}$. However, all but the most computationally expensive RP approach considered locate a bi-partition significantly faster than PCA, ICA or MDH. The benefit of RP is more apparent when considering its computational cost when $\mathcal{X}$ is very high-dimensional. For high-dimensional datasets, applying RP instead of PCA, ICA or MDH reduces the computational time to locate a bi-partition of $\mathcal{X}$ substantially.

Figure 6.3 shows the median CPU time, in seconds, over 30 replications for a complete divisive clustering using low-density separators located by our RP approach, PCA, ICA and MDH. Due to the cost of repeated calculations throughout the hierarchy, the times presented here only include a single component for PCA and ICA. For all the datasets gener-

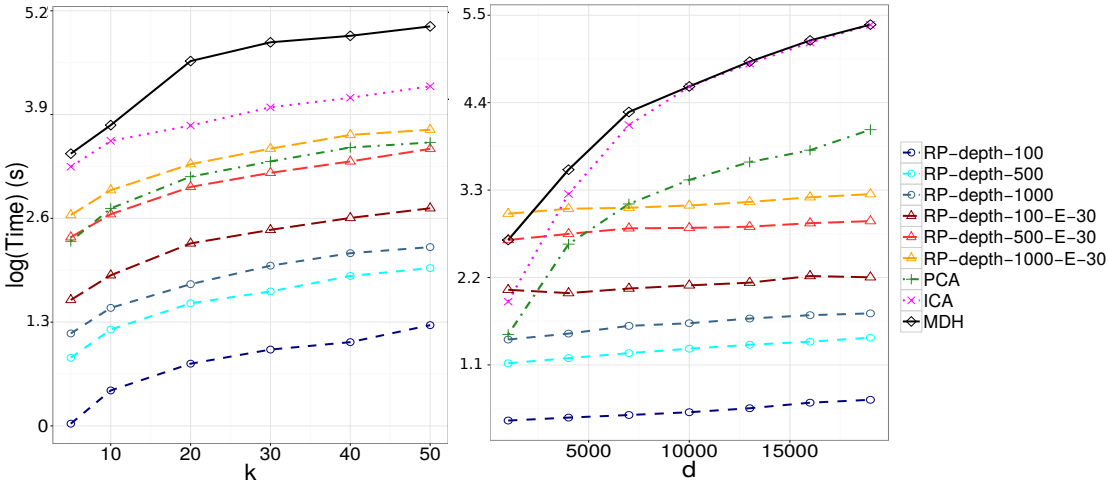


Figure 6.3: CPU time for a full clustering hierarchy with increasing numbers of clusters and dimensionality for RP, PCA, ICA and MDH.

ated, the number of points per cluster was 500. Increasing the number of clusters therefore increased the number of observations, as well as the number of splits needed in the hierarchy to identify all the clusters. When dimensionality was increased, the number of clusters was fixed to $k = 10$, and when the number of clusters was increased, dimensionality was fixed to $d = 5,000$.

For a full divisive clustering, the computational advantage of our RP approaches is more apparent than for a single bi-partition. This is due to the repeated calculations of globally optimal projection vectors required when applying PCA, ICA or MDH, while our RP approaches only computes $\mathcal{X}^r$ once. For increasing numbers of clusters, locating a single hierarchy using RP is significantly faster than using the alternative projection methods. As the dimensionality of $\mathcal{X}$ increases, the significantly lower computational cost of RP becomes highly attractive, while repeatedly computing optimal projections through PCA, ICA and MDH at each level of the hierarchy (at a cost which is quadratic in the dimensionality of $\mathcal{X}$) becomes infeasible practically. Therefore, when $\mathcal{X}$ is large and high-dimensional, the linear cost of RP with respect to both the number of observations and dimensions in $\mathcal{X}$, means that multiple RP hierarchies may be computed and combined with ensemble clustering in a

fraction of the time of locating a single hierarchy with the alternative projection techniques.

6.3.3 Performance Evaluation on Simulated Data

We now consider the clustering performance of low-density separators located using RP, with the four optimality criteria considered in Section 6.2.6, compared to the low-density separators located using PCA, ICA and MDH across simulated datasets. The performance of k-means++ using the Gap statistic to estimate the number of clusters is also included as a benchmark. For each of the simulated datasets, the data were drawn from a d-dimensional Gaussian mixture model with k components (clusters), each with $100k$ points as follows.

$$\mathbf{x} \sim \sum_{j=1}^{k} \frac{1}{k} N(\boldsymbol{\mu}_j, \sigma \mathbf{I})$$

$$\boldsymbol{\mu}_j \sim \text{Uniform}(0, 2)$$

$$\mathbb{P}(\sigma_i = s) = \begin{cases} 1/2, & s = 0.05 \\ 1/2, & s = 1 \end{cases} \qquad \forall i = 1, \ldots, d$$

The choice of σ in this generative model results in clusters which can be hard to detect along some projection directions, as seen in Figure 6.4, which shows pairwise plots of two-dimensional axes parallel projections along four dimensions of a dataset simulated from this model, and the univariate estimated density along these dimensions.

The effect of the high variability along some projection directions is reduced when $\mathcal{X}$ is high-dimensional, as illustrated by Figure 6.5, which provides the two-dimensional projections of two example simulated datasets with 1,000 and 10,000 dimensions onto their first two principal components. Evidently, the clusters should be identifiable along univariate projections which retain high variability, as long as the dimensionality of $\mathcal{X}$ is sufficiently high.

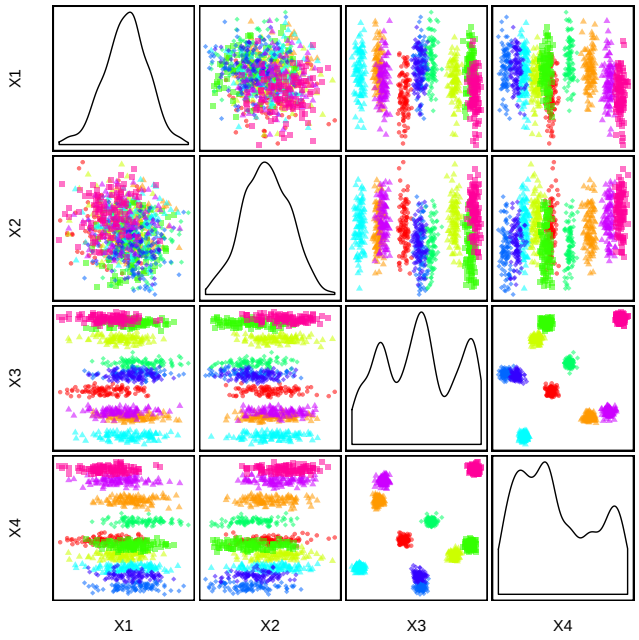
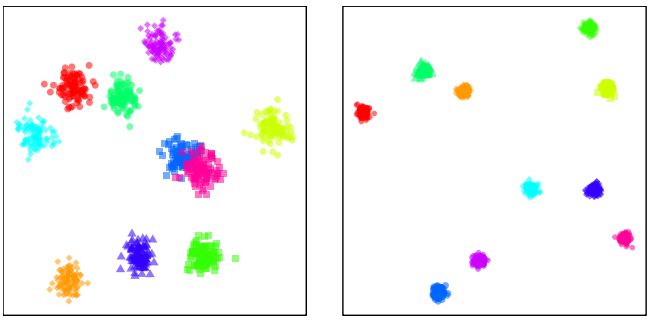


Figure 6.4: Four dimensions of a simulated dataset.

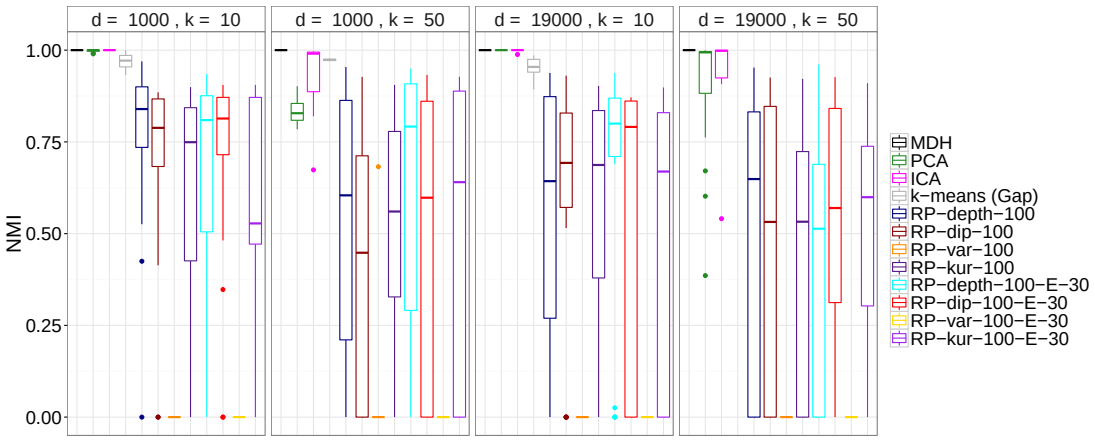


(a) d=1,000 (b) d=10,000

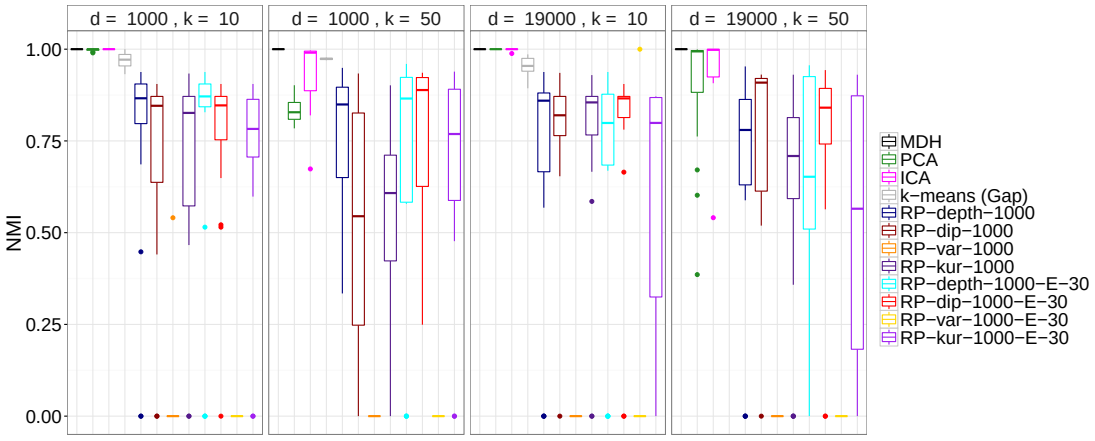
Figure 6.5: PCA projections of example simulated datasets with 10 clusters as dimensionality increases.

We simulated datasets with 10, 30 and 50 clusters and 1,000, 10,000 and 19,000 dimensions. The generative model induces higher levels of cluster overlap in datasets with larger numbers of clusters in fewer dimensions. Therefore, we expect the 1,000-dimensional datasets with 50 clusters to be the most challenging.

Figure 6.6 shows boxplots of the clustering performance of partitions from hierarchies of low-density separators located using the proposed RP approaches, PCA, ICA and MDH as well as k-means++, over 30 datasets, each with 10 and 50 clusters and 1,000 and 19,000



(a) 100 projections



(b) 1,000 projections

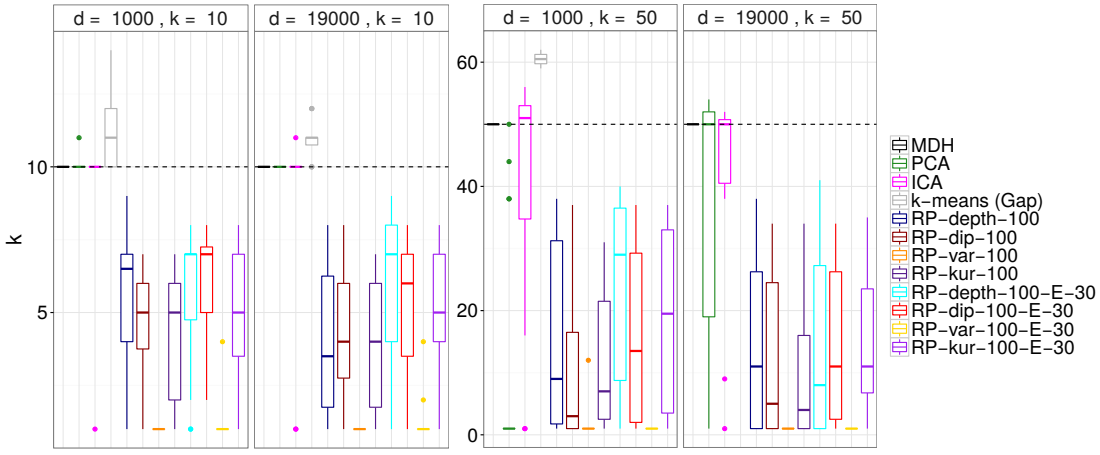
Figure 6.6: Boxplots of clustering performance from hierarchies of low-density separators located by RP approaches, PCA, ICA and MDH as well as k-means++ over 30 replications for simulated datasets with 10 and 50 clusters, 1,000 and 19,000 dimensions.

dimensions. The subfigures 6.6(a) and 6.6(b) correspond to using 100 and 1,000 random projections for each individual RP cluster hierarchy respectively. We have only included the most extreme numbers of clusters, dimensions and random projections for brevity. In general, we found that MDH was capable of almost perfect performance for all datasets. For the datasets with 10 clusters, where overlap is relatively low, PCA and ICA also correctly identified all the clusters. However, for higher numbers of clusters, inducing greater overlap, the 1,000-dimensional datasets were more challenging, particularly for PCA. The performance of k-means++ was relatively consistent across all the datasets, typically exhibiting clustering performance slightly below that of MDH. For the RP approaches, greater numbers of clus-

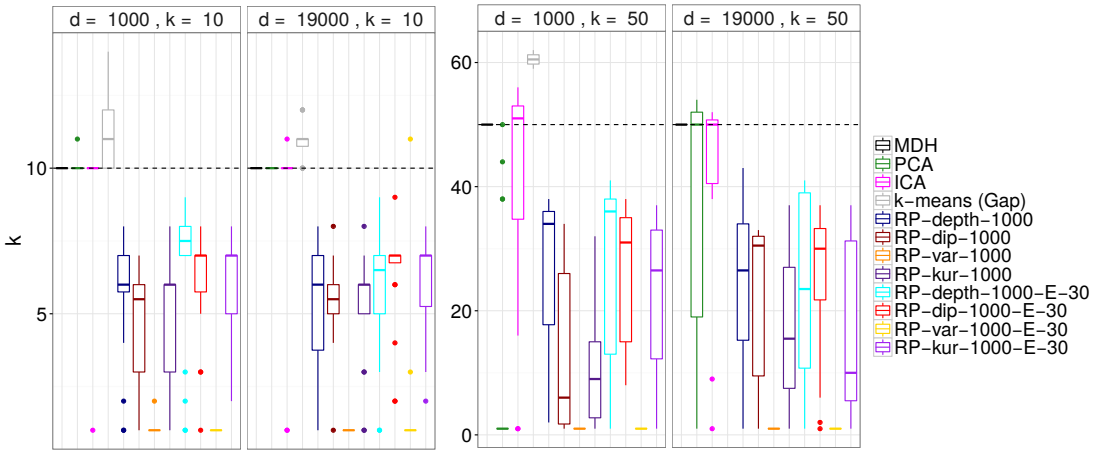
ters negatively affected performance, since higher overlap reduced the probability of locating projections along which the clusters were separable. Meanwhile, higher dimensionality resulted in a much larger search space for an appropriate projection direction. Therefore, a small, fixed number of random vectors were less likely to locate a set of projections that permit a low-density separator, unless the cluster overlap was very low. This results in less competitive performance for the RP approaches with 100 projections in datasets with large numbers of dimensions. However, taking more projections alleviates this problem.

For the RP approaches on the simulated datasets, all optimality criteria, except the maximum variance criterion, tend to locate higher quality partitions when searching over more random projections. The ensemble approach is effective for the 1,000-dimensional datasets, in some cases significantly improving performance compared to a single hierarchy. However, this is not the case for the 19,000-dimensional datasets, where the large search space for projection directions means the input partitions can be very varied, and in broad disagreement, resulting in inconsistent performance when an ensemble is used.

Over these simulated datasets, the RP approach has the most consistently competitive clustering performance when the relative depth optimality criterion is used. The maximum variance optimality criterion performs very poorly, even in the datasets with low cluster overlap, where PCA performs well. The relative performance of RP with the maximum dip statistic and minimum kurtosis optimally criteria are more varied. In the majority of cases, both of these criteria perform similarly to each other, and are generally competitive with the clustering performance of RP with the maximum relative depth criterion. RP with these optimality criteria perform well in the 19,000-dimensional datasets, however, using an ensemble does not improve the performance. By contrast, the individual partitions located through RP with the maximum dip statistic and the minimum kurtosis optimality crite-



(a) 100 projections



(b) 1,000 projections

Figure 6.7: Boxplots of estimated number of clusters from hierarchies of low-density separators located by RP approaches, PCA, ICA and MDH as well as *k*-means++ over 30 replications for simulated datasets with 10 and 50 clusters, 1,000 and 19,000 dimensions.

ria are not as competitive for the 1,000-dimensional datasets with 50 clusters, where cluster overlap is higher. However, if an ensemble is used, the performance of these approaches improves significantly.

Figures 6.7(a) and 6.7(b) provide boxplots of the estimated number of clusters located over 30 simulated datasets containing 10 and 50 clusters in 1,000 and 19,000 dimensions using hierarchies of low-density separators from the proposed RP approaches with 100 and 1,000 random projections respectively. This is compared to the number of clusters located from a hierarchy of low-density separators arising from PCA, ICA and MDH as

well as the Gap statistic for k-means++. The black dashed line indicates the true number of clusters. For the datasets with only 10 clusters where overlap is low, locating a hierarchy of low-density separators using PCA, ICA and MDH allows the number of clusters to be estimated almost perfectly. In the datasets with 50 clusters, where overlap is much higher, ICA and PCA are more susceptible to underestimating the number of clusters, indicating that for these datasets, it is necessary to actively seek projection directions which permit the lowest possible density separators to locate all the clusters. Meanwhile, the Gap statistic overestimates the number of clusters in all cases. All of the RP approaches underestimate the number of clusters, especially when only 100 random projections are used to search for an appropriate projection direction. In general, RP with the maximum relative depth and maximum dip statistic optimality criteria estimate the number of clusters more accurately than the other two RP approaches, particularly when an ensemble is used. This is a result of these approaches retaining projections which have a multimodal estimated density, allowing cluster separation based on our splitting rule. Although the minimum kurtosis optimality criterion favours projections with a bi-modal structure, this criterion also avoids projections with outliers. This can result in a tendency to locate a set of projections with a unimodal density that resembles a uniform distribution over a set of projections with one large mode and a smaller mode in the estimated density. Meanwhile, the maximum variance optimality criterion does not consider the modality of the estimated density of the projections so is also susceptible to selecting projections with a unimodal estimated density, which is not appropriate to partition $\mathcal{X}$ based on our splitting rule.

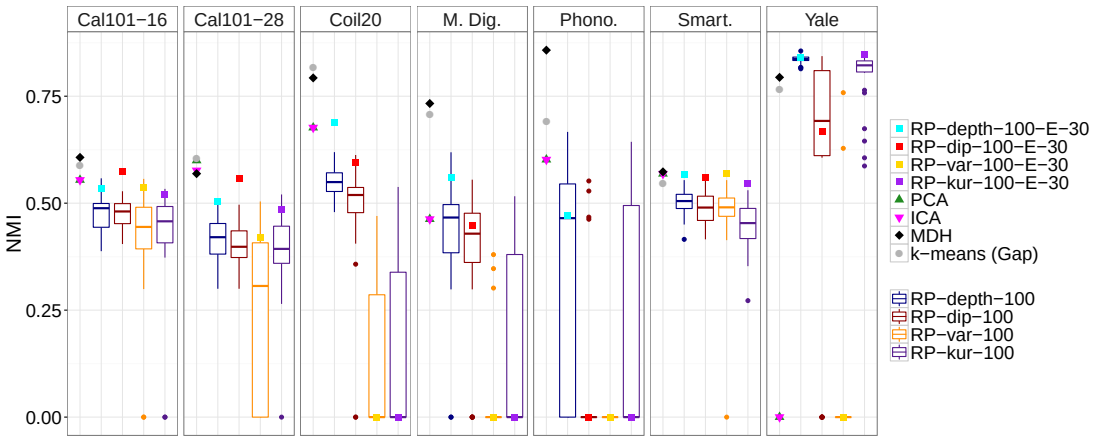
6.3.4 Performance Evaluation on Real Data

In this section, the quality of the partitions arising from hierarchies of low-density separators located using univariate projections computed by our proposed RP approaches, MDH,

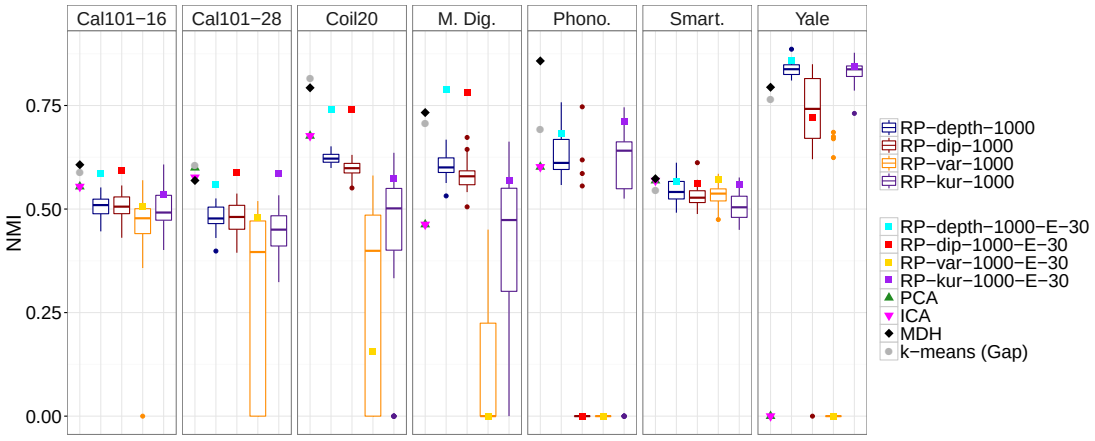
PCA and ICA are investigated across a variety of real benchmark datasets. The datasets considered are:

- Caltech-101-16 (silhouettes): binary images with 16 by 16 (256) pixels each, which are silhouettes of the classic Caltech-101 dataset. The original dataset consists of 1641 images from 101 categories. However, many of these categories only contain a small number of images relative to some larger categories. These small, sparse clusters are not consistent with our aim of locating dense clusters so we removed these from the dataset. The resulting dataset contains 2901 images from six categories.
- Caltech-101-28 (silhouettes): the same images as the Caltech-101-16 datasets but with 28 by 28 (784) pixels, processed in the same way as above.
- Coil-20 : Images of 20 grey-scale images of objects against a black background, with 128 by 128 (16,384) pixels. Each object is rotated around 360 degrees, with images taken at each 5 degree interval. Thus there are 1420 images, belonging to 20 categories in 16,384 dimensions.
- Multi Digits : A set of 649 features of handwritten digits from 0 to 9. Each digit has 200 occurrences, resulting in 2,000 observations belonging to 10 equally sized categories.
- Phoneme : This dataset is formed from 4,509 continuous speech recordings of five phonemes. Each speech recording is characterised by 512 samples (taken as the features for each instance). The number of recordings from each of the five categories ranges from 695 to 1,163.
- Smartphone : 10,929 sensor (accelerometer) signals recorded on smartphones from 30 volunteers performing 12 tasks (which define the clusters). Each signal is processed to have 516 features.
- Yale Faces : 5,850 images, each with 1,200 pixels of 10 people under 585 viewing conditions. The clusters are defined by the 10 individuals

Figures 6.8(a) and 6.8(b) show boxplots of the clustering performance of the proposed RP approaches (over 30 experiments) for the real datasets considered using 100 and 1,000 random projections respectively. We also include dots for the performance of an ensemble over these 30 partitions, and the performance of hierarchies of low-density separators arising from MDH, PCA and ICA, as well as k-means++ as a comparison.



(a) 100 projections



(b) 1000 projections

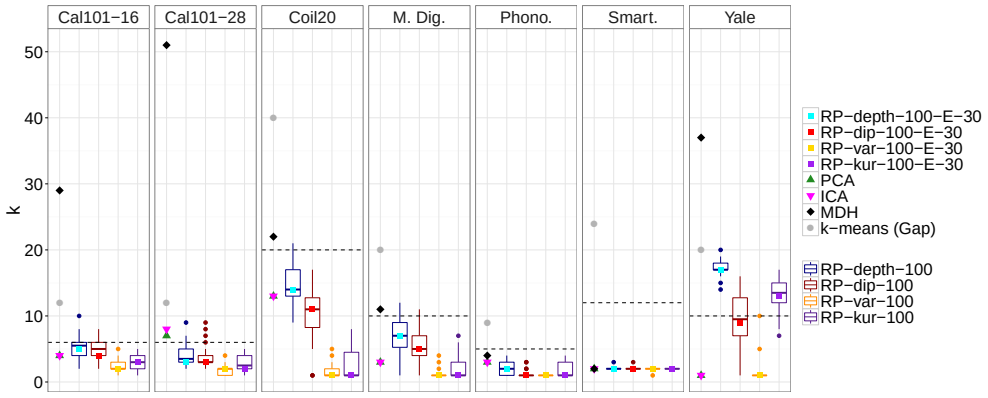
Figure 6.8: Boxplots of clustering performance from hierarchies of low-density separators located by RP approaches, PCA, ICA and MDH as well as k-means++ over real datasets.

Across the real datasets considered, locating projections which permit cluster separators that intersect regions of minimal density in $\hat{p}_{\mathbf{x}}$ is the most appropriate approach to correctly identify the clusters, with MDH performing better than PCA or ICA for all but one dataset, and similarly or better than k-means++ for all datasets. Therefore, the maximum relative depth is the best performing optimality criterion for RP, while the maximum variance criterion produces relatively poor partitions in general. For the Caltech101 datasets, RP with the minimum kurtosis and maximum dip statistic optimality criteria also locate competitive partitions, which have similar performance to RP with the maximum relative depth optimality criterion. For these two datasets, all the RP approaches locate a higher-quality

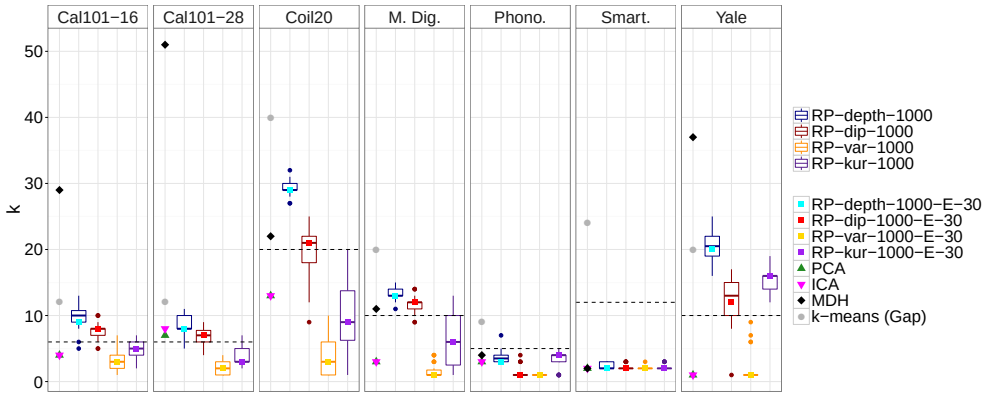
partition with an ensemble clustering, with a significant improvement over any single input partition in some cases. For the maximum relative depth and minimum dip statistic optimality criteria over 1,000 random projections, this ensemble clustering is competitive with the performance of MDH and *k*-means++, and exhibits better performance than the partitions arising from PCA and ICA projections. For the Coil20 dataset, RP with the maximum relative depth and maximum dip statistic optimality criteria locate partitions which are competitive with the partitions from the globally optimal projection techniques and *k*-means++, especially with an ensemble clustering, which allows the RP approach to locate a more accurate clustering than PCA or ICA. For this dataset, the minimum kurtosis optimality criterion is not as appropriate, however, this still performs significantly better than the maximum variance criterion. For the Multi Digits dataset, RP using 1,000 random projections with the maximum relative depth and maximum dip statistic optimality criteria and an ensemble clustering allows the RP approach to perform very well, locating a partition with higher clustering performance than any of the optimal projection methods or *k*-means++. Again RP with the minimum kurtosis and maximum variance optimality criteria are much less competitive. For the phoneme dataset, RP with the maximum dip statistic and maximum variance optimality criteria perform very poorly. For this dataset, the minimum kurtosis criterion produces the highest quality partition of the RP approaches, closely followed by the relative depth criterion. Both of these RP approaches perform better than PCA or ICA, and also perform similarly to *k*-means++ when the partitions are combined with an ensemble clustering, but MDH produces the best performing partition for this dataset. For the Smartphone dataset, all the optimality criteria for the RP approach locate partitions which are competitive with the partitions arising from hierarchies of low-density separators located by MDH, PCA and ICA or a centroid-based clustering using *k*-

means++. For this dataset, RP with the maximum variance criterion is competitive with the optimal projection techniques, and is close to the performance of RP with the maximum relative depth criterion. Due to the relatively low diversity in the performance of the input partitions, the advantage of using an ensemble is not as significant here. For the Yale Faces dataset, the clustering structure is not evident in directions of high variability, so PCA and RP with the maximum variance criterion fail to locate a meaningful partition. By contrast, RP with the minimum kurtosis and maximum relative depth criteria locate partitions which have higher clustering performance than any of the alternative projection techniques, including MDH or *k*-means++. RP with the maximum dip statistic optimality criterion performs relatively competitively for this dataset, but an ensemble clustering does not improve performance.

Figure 6.9 shows boxplots of the estimated number of clusters located by a hierarchy of low-density separators using the proposed RP approaches, PCA, ICA and MDH as well as the Gap statistic for *k*-means++. The dashed black line indicates the true number of clusters for each dataset, and the subfigures 6.9(a) and 6.9(b) correspond to using 100 and 1,000 random projections respectively. For all the datasets except Smartphone, using more random projections allows the RP approaches to identify more clusters. For the real datasets, the RP approaches estimate the number of clusters relatively accurately. In general, RP with the maximum relative depth and maximum dip statistic optimality criteria are susceptible to slightly overestimating the number of clusters, while RP with the maximum variance and minimum kurtosis optimality criteria tend to underestimate the number of clusters. This is expected since the maximum relative depth and maximum dip statistic criteria select projections which are more likely to locate a valid bi-partition using our splitting rule, while for reasons discussed in Section 6.3.3, the minimum kurtosis and maximum variance optimality



(a) 100 projections



(b) 1,000 projections

Figure 6.9: Boxplots of estimated number of clusters from hierarchies of low-density separators located by RP approaches, PCA, ICA and MDH as well as k-means++ over real datasets.

criteria do not always favour projections with a multimodal estimated density. This overestimation of the number of clusters is also the case for MDH, which locates significantly more than the true number of clusters for the Caltech101 and Yale Faces datasets. Hierarchies of low-density separators located using PCA and ICA generally underestimate the number of clusters, since these projection techniques are more likely to locate projections with a unimodal estimated density than MDH. Meanwhile, the Gap statistic overestimates the number of clusters for all datasets.

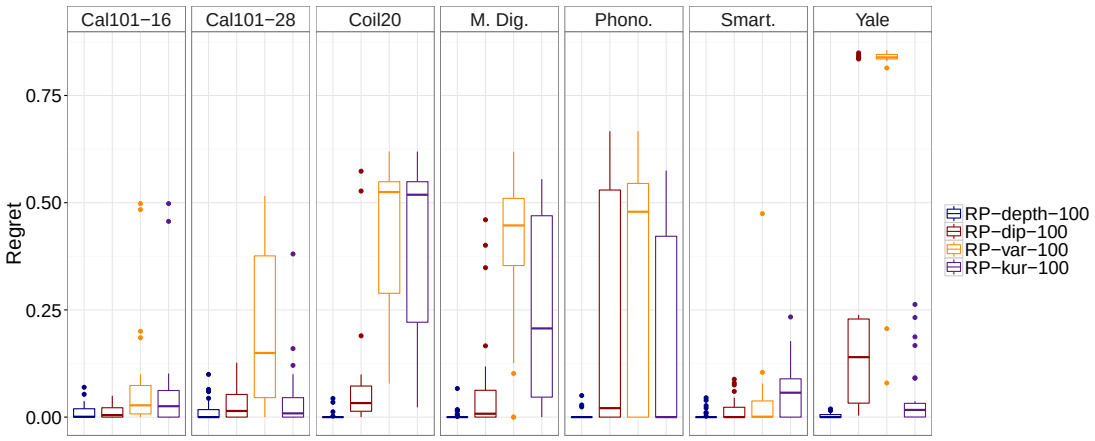
To compare the relative quality of the partitions produced using our RP approaches with the four optimality criteria considered, Figures 6.10(a) and 6.10(b) provide boxplots of regret associated with each of the optimality criteria over 30 experiments for each of the real

datasets, when 100 and 1,000 random projections are used. The regret is defined as the difference in performance of the best performing optimality criterion and the criterion in question,

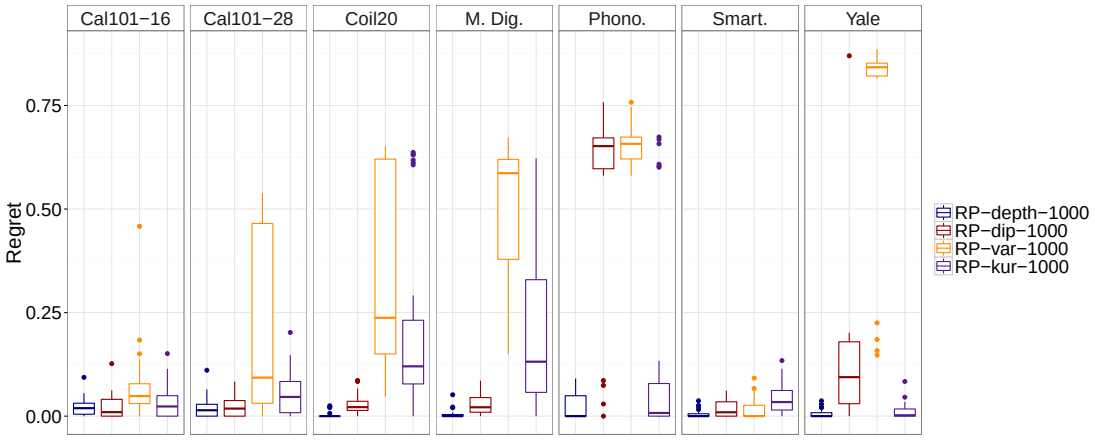
$$\text{Regret}(f) = \text{NMI}(\boldsymbol{\pi}_{f^\star}, \boldsymbol{\pi}^\star) - \text{NMI}(\boldsymbol{\pi}_f, \boldsymbol{\pi}^\star) \tag{6.11}$$

where f is the optimality criterion in question, $\boldsymbol{\pi}_f$ is the partition induced by the RP approach with this criterion and $\boldsymbol{\pi}_{f^\star}$ is the partition induced by the RP approach with the best performing optimality criterion. This difference is taken over each of the 30 cluster trees produced by each optimality criterion, for the same collection of random projections, so any difference in performance is a direct consequence of the different criteria, not different random projections. A regret close to zero indicates a consistently competitive performance relative to the alternative optimality criteria.

For all the datasets except Phoneme, the relative performance of the different optimality criteria is similar when using either 100 and 1,000 random projections. The relative depth criterion is the most competitive optimality criterion, with a median regret close to zero for all datasets. The maximum dip statistic is the second most competitive criterion for the Caltech101, Coil20 and Multi Digits datasets, while the minimum kurtosis criterion has the second best relative performance for the Phoneme and Yale Faces datasets. The maximum variance criterion has the worst relative performance except for the Smartphone dataset, as expected from the results in Figure 6.8. The relatively poor performance of the maximum variance optimality criterion compared to the other criteria suggests that using criteria that are more likely to select projections with a bi-modal or multimodal structure is beneficial to more accurately separate the clusters in these datasets.



(a) 100 projections



(b) 1,000 projections

Figure 6.10: Boxplots of regret with respect to NMI for the four optimality criteria for RP approaches.

6.4 Experimental Results using n-dimensional Projections of Feature Vectors

In this section, we conduct an empirical evaluation of the proposed RP approach when $\mathcal{X}$ is a set of non-linearly mapped feature vectors, which have been projected onto the n-dimensional basis defined by the kernel principal components. This non-linear mapping results in linear separators of $\mathcal{X}$ corresponding to non-linear separators of the original observations permitting the identification of non-linearly separable clusters. We consider varying numbers of random projections over which to search for an appropriate projection vec-

tor for cluster identification, and compare the performance of hierarchies of low-density separators located using the proposed RP approaches to alternative low-density separators located using PCA, ICA and MDH, over the mapped feature vectors of a variety of real-world benchmark datasets. The performance of k-means++, with the number of clusters determined using the Gap statistic is also included as a benchmark. However, similarly to Section 6.3, the computational cost of evaluating the Gap statistic was infeasible for some of the larger datasets, in which case the performance of this method is omitted. We do not conduct a simulation study as part of this investigation, since it is not possible to generate a set of observations with any guarantees about their structure after mapping them into the feature space.

6.4.1 Details of Implementation

Parameter Settings

For all algorithms considered, we used the same parameter settings as stated in Section 6.4.1 for the bandwidth, h, interval width, α (for MDH) and the relative depth threshold for the stopping rule proposed in Section 6.2.4. We also retained our choice of constructing 30 different trees using our RP approaches as input clusterings to an ensemble. For the construction of these trees we experimented with searching over varying numbers of random projections at each level of the hierarchy.

Figure 6.11 shows the increase in clustering performance of a bi-partition of the mapped feature vectors of some of the real datasets considered with increasing numbers of random projections. This was produced in the same way as Figures 6.1(a) and 6.1(b), by partitioning each successive set of univariate projections at the minimiser $b^\star$ of the estimated density with maximum relative depth, and recording the current best SR, as stated in Eqs. (6.9) -

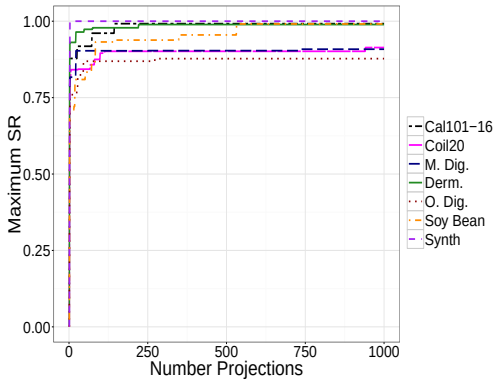


Figure 6.11: Increase in success ratio with increasing number of random projections for mapped feature vectors of real benchmark datasets summarised in Table 6.1.

(6.10). Some datasets are omitted for clarity, but these showed a similar rate of convergence to the best partition. For the mapped observations, the RP approach converges to a high-quality bi-partition faster than for the original observations, with approximately 100 random projections being sufficient to locate the best performing bi-partition for most datasets. For consistency with Section 6.3, we experimented with searching over 100, 500 and 1,000 random projection vectors at each level of the hierarchy for the divisive RP approaches in Section 6.4.2. However, since Figure 6.11 indicates a higher rate of convergence to a high-quality separator of the feature vectors, we expect the increase in clustering performance when searching over large numbers of random projections to be less significant than when partitioning the original observations.

As for all kernel-based approaches, the choice of kernel function, and any subsequent parameter values critically affect the clustering performance of all the algorithms applied in this section. This is a well-documented, open problem in the literature, and as such a robust approach to determine an optimal choice of kernel is beyond the scope of our work. We use the Gaussian kernel, since this is the most widely used in the literature. To tune the kernel

parameter, we use the local scaling approach proposed by Zelnik-Manor and Perona (2004),

$$\kappa(\mathbf{y}_i, \mathbf{y}_j) = \exp\left\{-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{s_i s_j}\right\}$$

where the $\mathbf{y}_i$ are the original observations (before the feature mapping) and s_i and s_j are the distances from the ith and jth original observations to their seventh nearest neighbours respectively. This can handle data on multiple scales and is very effective in our experience. We use the same kernel matrix for all algorithms considered to compute the projections of the feature vectors onto their kernel principal components, stored in $\mathcal{X}$, upon which univariate projections are computed.

6.4.2 Performance Evaluation on Real Data

In this section we conduct an empirical evaluation of the performance of our proposed RP approaches when locating low-density separators of the n-dimensional KPCA projections of the feature vectors of real-world benchmark datasets. In the feature space, the dimensionality of the search space for an appropriate projection vector for clustering is determined by the number of observations, and not the dimensionality of the original dataset. Therefore, when clustering the feature vectors, our RP approach is relevant for datasets with large numbers of observations, irrespective of the dimensionality of the original dataset. Hence, in this section we consider additional datasets to those presented in Section 6.3.4, the main characteristics of which are summarised in Table 6.1. The additional datasets are all available from the UCI machine learning repository (Lichman, 2013) where more detailed descriptions can be found, so we omit these here.

Figures 6.12 and 6.13 present boxplots of the clustering performance of hierarchies of low-density separators located by our proposed RP approaches when applied to the n-

Table 6.1: Main characteristics of real datasets considered.

Dataset	n	d	k
Cal101-16[1]	2901	256	6
Cal101-28[1]	2901	784	6
Coil20[2]	1420	16384	20
Dermatology[3]	366	34	6
Heart Disease[3]	294	13	5
Image Segmentation[3]	2309	19	7
Ionosphere[3]	351	33	2
Iris[3]	150	4	3
Isolet[3]	7797	617	26
Multi. Digits[3]	2000	216	10
Opt. Digits[3]	5620	64	10
Pen Digits[3]	10992	16	10
Phoneme [4]	4506	256	5
Satellite[3]	6435	36	6
Seeds[3]	210	7	3
Smartphone[3]	10929	561	12
Soy Bean[3]	682	35	19
Synth[3]	600	60	6
Votes[3]	435	16	2
Wine[3]	178	13	3
Yale Faces[5]	5850	1200	10

[1](Marlin, 2014) available from people.cs.umass.edu/~marlin/data.shtml

[2](Nene et al., 1996) available from cs.columbia.edu/CAVE/software/softlib/coil-20.php

[3]UCI machine learning repository (Lichman, 2013)

[4](Hastie et al., 1995) available from statweb.stanford.edu/tibs/ElemStatLearn/data.html

[5](Georghiades et al., 2001) available from cervisia.org/machine_learning_data.php

dimensional KPCA projections of the feature vectors of the real-world datasets considered. These plots correspond to searching over 100 and 1,000 random projections at each level of the hierarchy respectively. For each dataset, we located 30 hierarchies of low-density separators using the four RP optimality criteria suggested in Section 6.2.6, and then produced an ensemble clustering over these partitions. The clustering performance of the ensemble clusterings from the RP approaches are indicated by square dots. As a comparison, the performance of a divisive clustering using low-density separators located by MDH, PCA and ICA are included along with the clustering performance of k-means++, where the number of clusters was estimated using the Gap statistic.

For the majority of datasets, searching over more random projections permits the RP

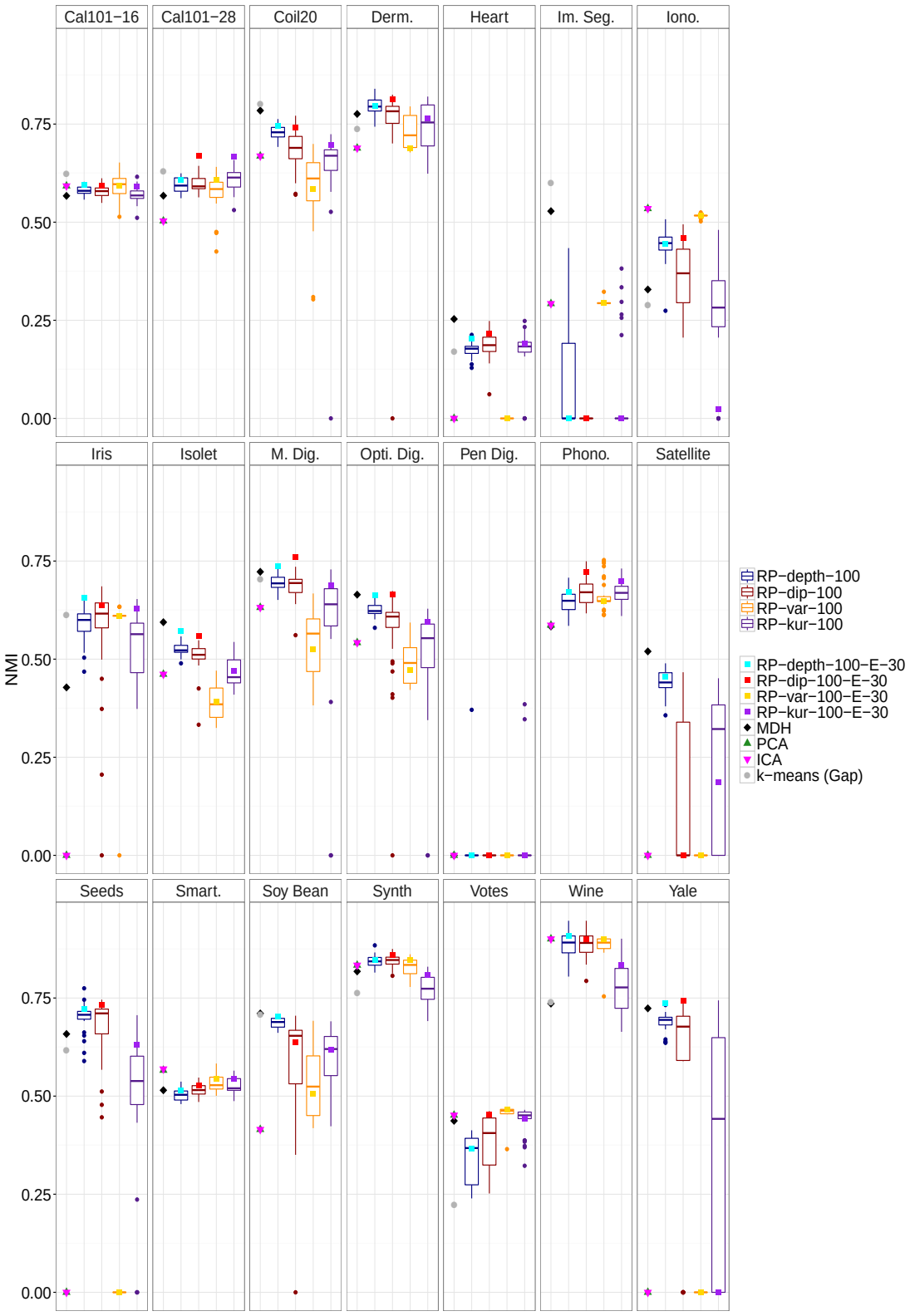


Figure 6.12: Boxplots of clustering performance from hierarchies of low-density separators located by RP approaches with 100 projections, PCA, ICA and MDH as well as *k*-means++ over mapped feature vectors of real datasets.

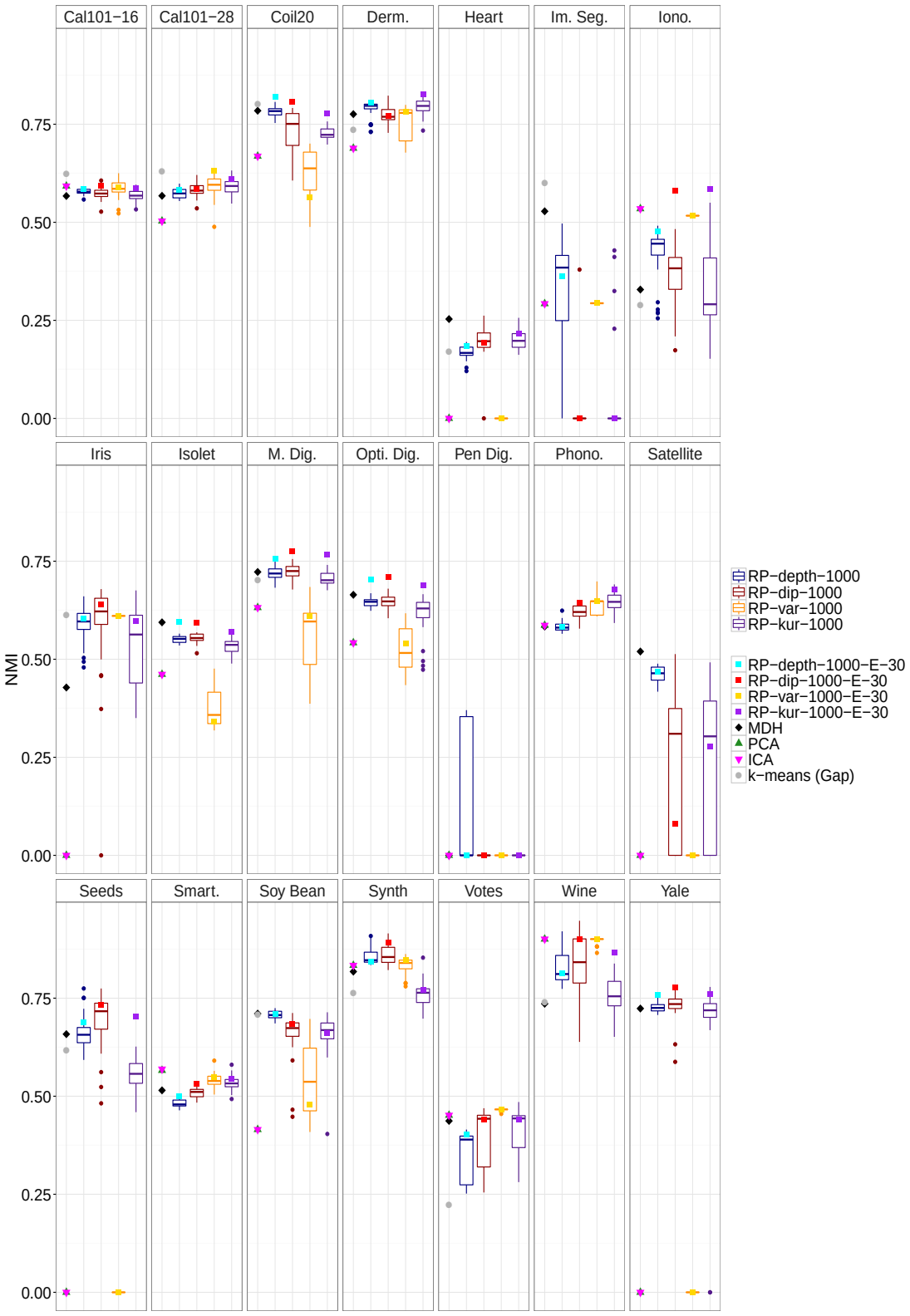


Figure 6.13: Boxplots of clustering performance from hierarchies of low-density separators located by RP approaches with 1,000 projections, PCA, ICA and MDH as well as *k*-means++ over mapped feature vectors of real datasets.

approaches to locate higher quality cluster separators with all the optimallity criteria considered. However, the increase in clustering performance when using 1,000 instead of 100 random projections is not as substantial as in the original data space (Section 6.3.4). For the mapped feature vectors, all the proposed RP approaches are capable of locating high-quality partitions. This performance is almost always improved by an ensemble over the 30 clusterings located by different collections of random projections. The use of this ensemble clustering allows a group of hierarchies of low-density separators located by the proposed RP approaches to perform better than *k*-means++ or a single hierarchy of low-density separators located by MDH, PCA or ICA for 17 of the 21 datasets considered.

For the mapped feature vectors, the most appropriate optimality criterion for locating univariate projections that allow accurate cluster separation is more varied than for the original observations. This is evident for the RP approaches as well as the competing projection techniques. MDH locates the best projections of the techniques that locate globally optimal projections, but PCA and ICA also offer similar or better performance regularly. This suggests that projection directions which allow the lowest possible density separator are often, but not always, the most appropriate for the separation of the true clusters in the feature space. For the RP approach, none of the optimality criteria consistently result in the highest clustering performance.

For most datasets, the best choice of optimality criterion for the RP approach is the criterion that most closely relates to the objective of the best performing globally optimal projection direction. Generally, the maximum relative depth and maximum dip statistic are the most competitive criteria. RP with the minimum kurtosis optimality criterion also performs well for most datasets, often performing similarly to the best RP approaches. Meanwhile, RP with the maximum variance optimality criterion is most susceptible to relatively poor

performance. The poor performance of RP with this optimality criterion tends to be for datasets where PCA also fails to locate appropriate projections for clustering. However, for datasets where directions of high variability are suitable for cluster separation, RP with this optimality criterion also performs competitively.

Figures 6.14 and 6.15 show boxplots of the number of clusters located by hierarchies of low-density separators located by the RP approaches, MDH, PCA and ICA as well as the Gap statistic for k-means++ when applied to the the mapped feature vectors of the real datasets considered. These figures correspond to using 100 and 1,000 random projections to search for an appropriate cluster separator respectively. For most datasets, and choices of optimality criterion for the RP approach, searching over more random projections results in the location of more clusters.

For the RP approach, the maximum relative depth and maximum dip statistic optimality criteria tend to locate more clusters than the maximum variance or minimum kurtosis optimality criteria, since projections with a strongly multimodal estimated density (favoured by the maximum relative depth and maximum dip statistic criteria) will permit cluster separation based on our splitting rule in Section 6.2.4. Actively seeking such projection directions tends to lead to an overestimation of the number of clusters in the feature space. This is a result of the high level of sparsity of the mapped observations over an n-dimensional space, increasing the probability of locating univariate projections whose estimated density is multimodal, even if a true low-density separator of high-density regions on $\hat{p}_{\mathbf{x}}$ does not exist. This is also evident for MDH, which is susceptible to locating significantly more than the true number of clusters. The tendency to overestimate the number of clusters as a result of the sparsity of the mapped observations is not unique to the minimum density separation approach, with the Gap statistic exhibiting the same problem. By contrast, PCA, ICA and

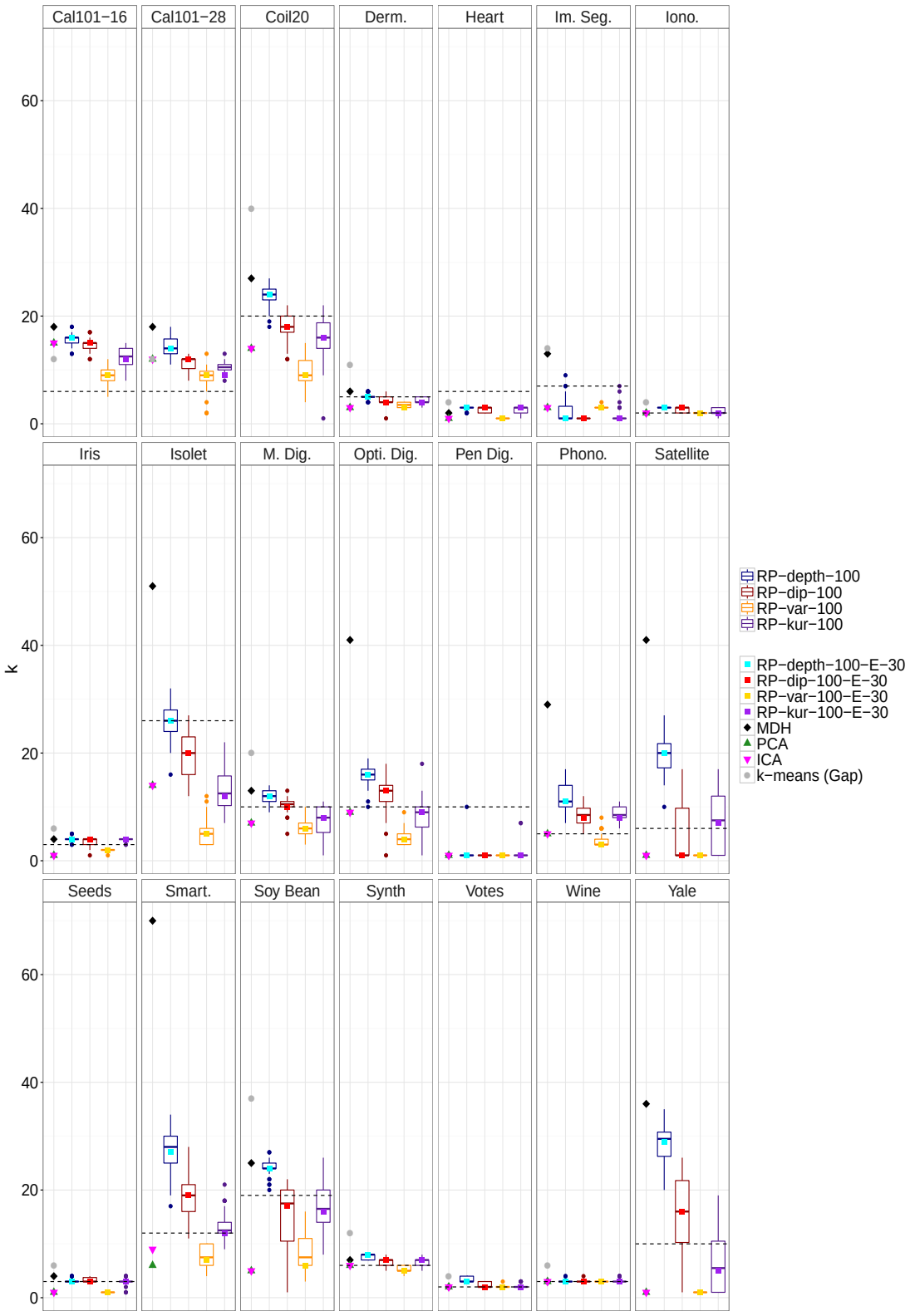


Figure 6.14: Boxplots of estimated number of clusters from hierarchies of low-density separators located by RP approaches with 100 projections, PCA, ICA and MDH as well as k-means++ over mapped feature vectors of real datasets.

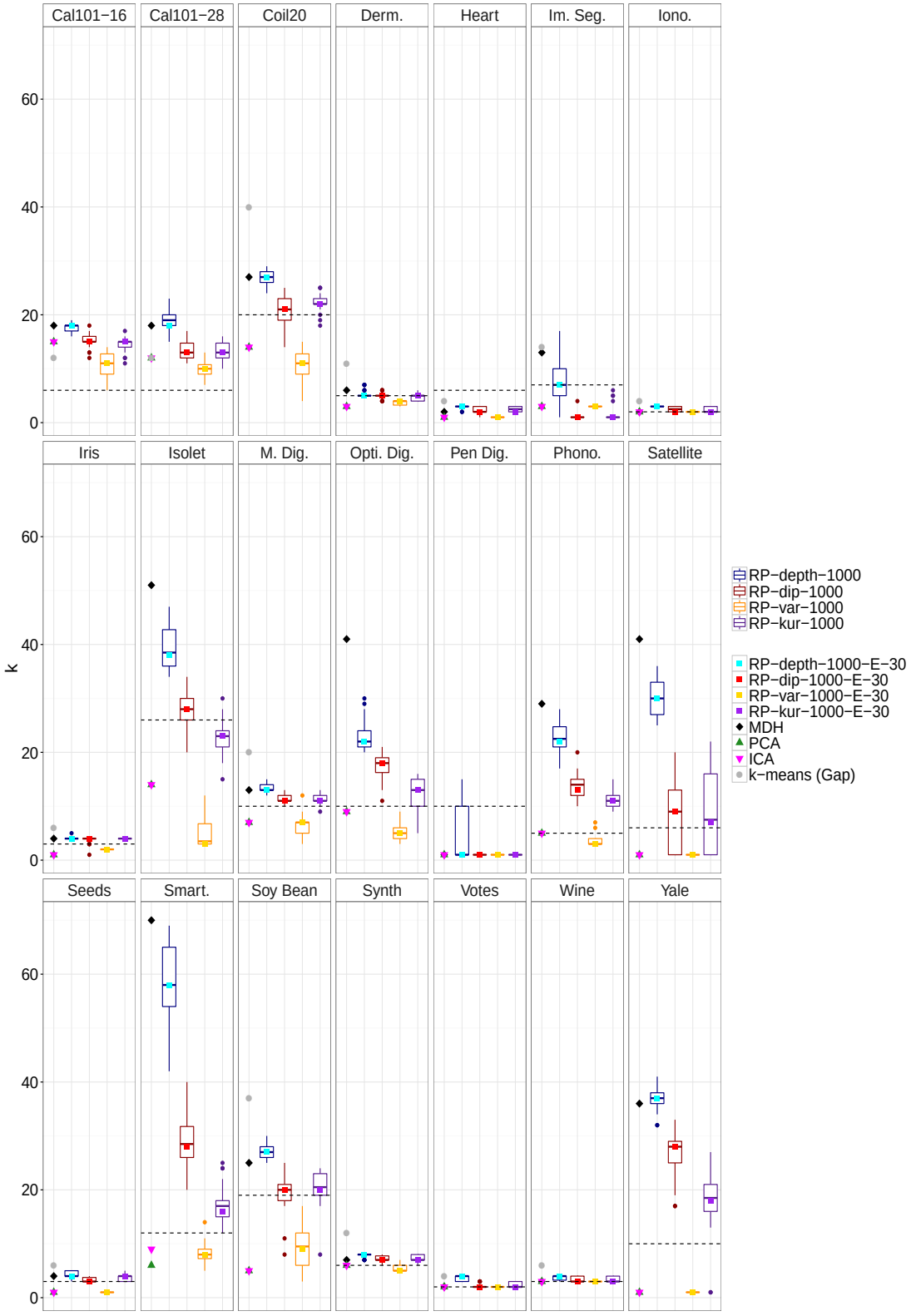


Figure 6.15: Boxplots of estimated number of clusters from hierarchies of low-density separators located by RP approaches with 1,000 projections, PCA, ICA and MDH as well as k-means++ over mapped feature vectors of real datasets.

RP with the maximum variance and minimum kurtosis optimality criteria are less likely to overestimate the number of clusters. This is due to directions of high variability not considering the modality of the projections at all, and directions with minimal kurtosis being more robust to outliers in the tails of the estimated density of the projections. However, these methods fail to identify all the clusters for some datasets.

The relative performance of the different optimality criteria for the RP approaches when applied to the mapped feature vectors of the real datasets is considered in Figures 6.16 and 6.17. These provide boxplots of the regret (defined in Eq. (6.11)) associated with selecting a random projection for cluster separation from 100 and 1,000 projections respectively, using each of the four optimality criteria considered. Each of the 30 experiments for each dataset used a fixed collection of random projections, so the difference in performance is a direct consequence of the choice of optimality criteria.

For the mapped feature vectors of the datasets considered, the use of more random projections does not significantly change the relative performance of the different optimality criteria. When clustering the mapped feature vectors, the most competitive optimality criterion is less clear than for the original observations, with no single criterion consistently achieving minimal regret. The relative performance of the maximum variance optimality criterion differs significantly across these datasets, sometimes achieving a regret of almost zero, but also having a very high regret for some datasets. The regret associated with the other three optimality criteria is less varied across the different datasets. The relative depth is the optimality criterion which is least likely to result in an unusually high regret, however, all four optimality criteria can achieve minimal regret on a number of datasets.

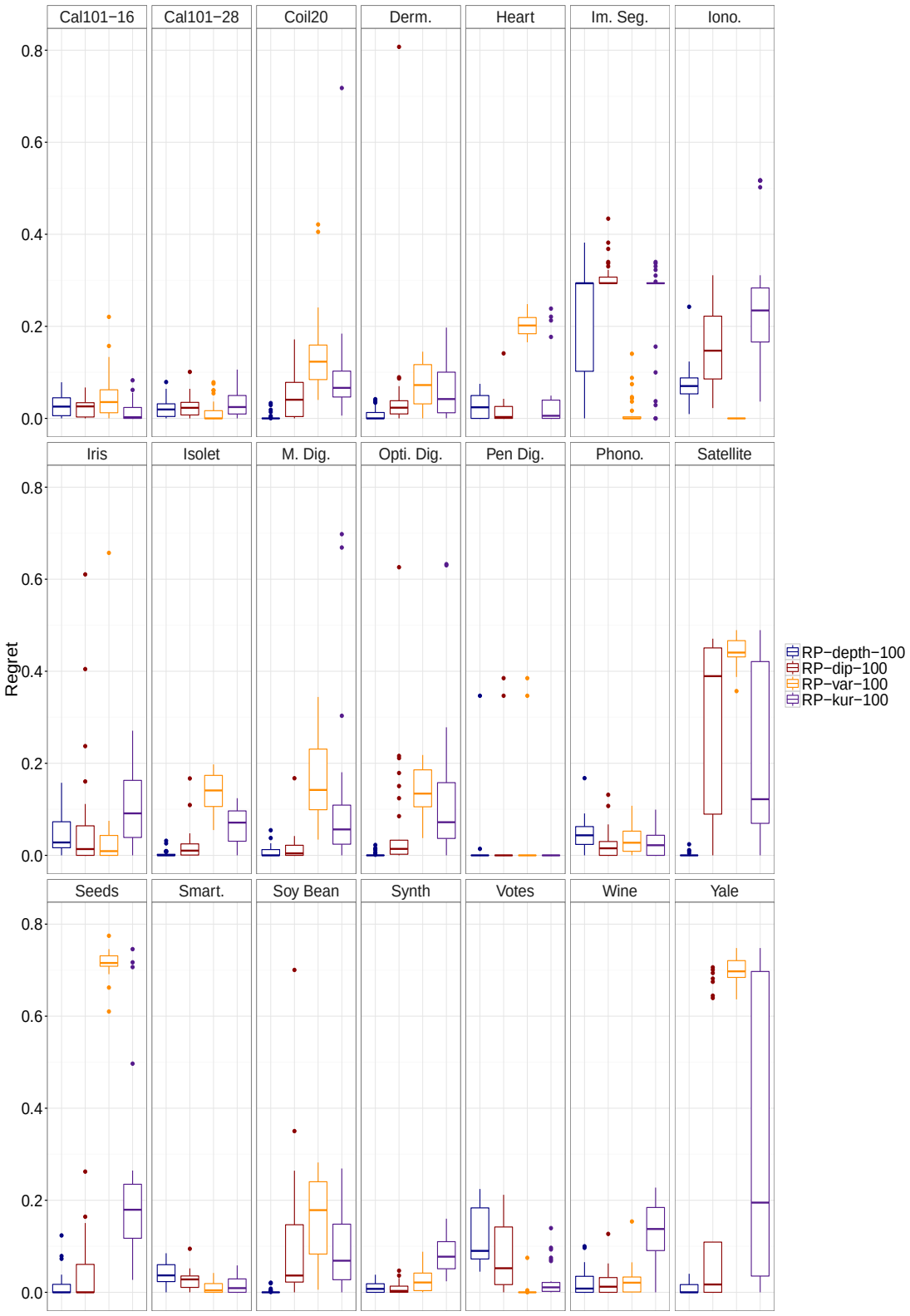


Figure 6.16: Boxplots of regret with respect to NMI for the four optimality criteria for RP approaches using 100 projections over mapped feature vectors of real datasets.

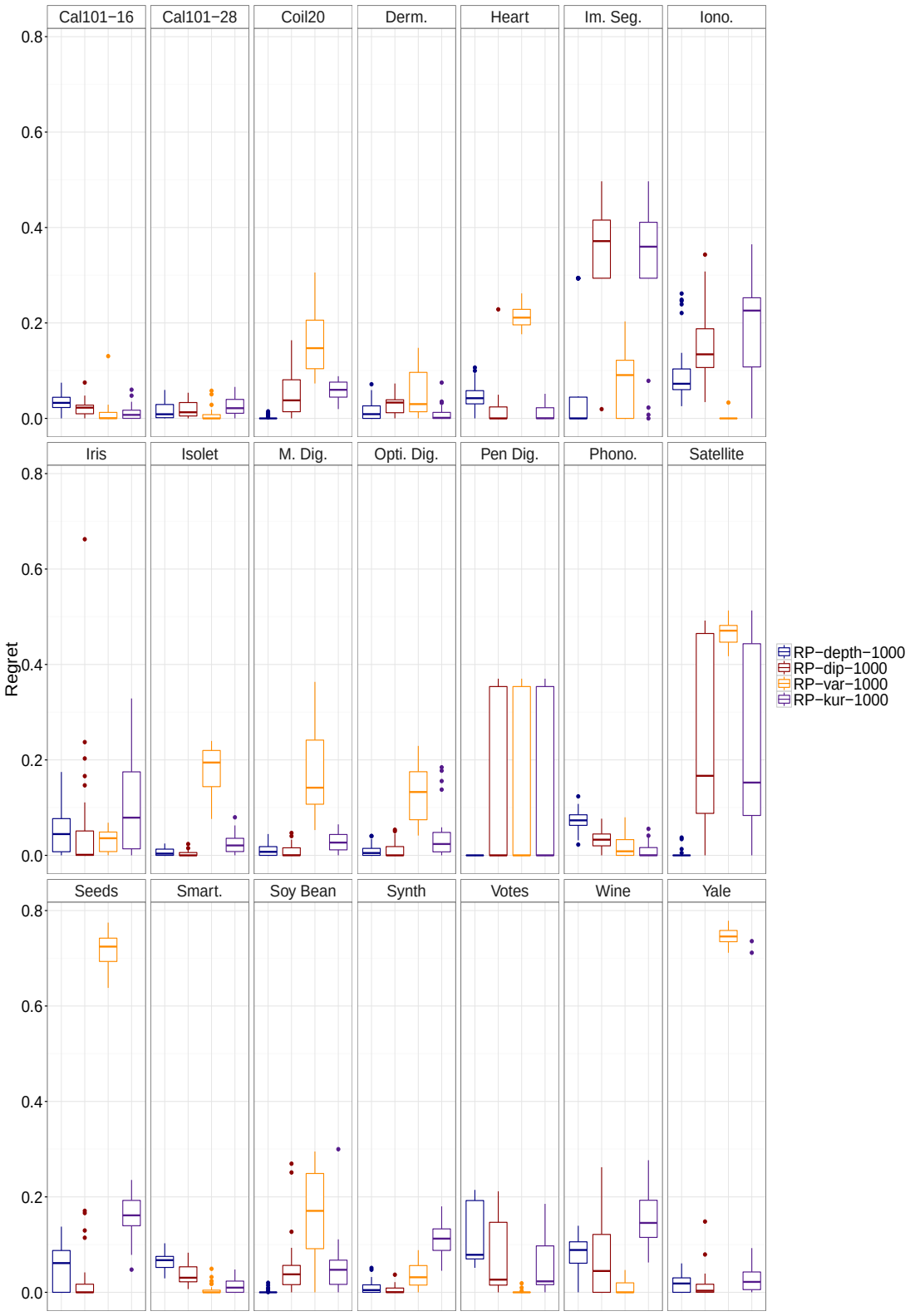


Figure 6.17: Boxplots of regret with respect to NMI for the four optimality criteria for RP approaches using 1,000 projections over mapped feature vectors of real datasets.

6.5 Summary of Experimental Results

In Sections 6.3 and 6.4, we found that searching over a finite collection of one-dimensional subspaces for an approximately optimal projection direction for clustering permits the location of low-density cluster separators at a significantly lower computational cost compared to locating globally optimal projections. Further, if it were possible to quantify the suitability of a projection vector based on the clustering accuracy of the resulting partition, a high-quality cluster separator is located by searching over a relatively small number or random subspaces. The convergence to a high-quality cluster separator was faster when locating a bi-partition of the feature vectors than the original observations. This is a result of the high level of sparsity in the feature space increasing the probability of generating a projection vector along which a suitable low-density cluster boundary can be located.

In clustering we require alternative criteria to quantify the suitability of a set of projections for cluster separation. Of the optimality criteria considered, the maximum relative depth criterion offered the most consistently competitive performance for the proposed RP approaches, although the difference in performance of the different optimality criteria was not as significant when clustering the feature vectors. In almost all cases, computing an ensemble clustering over partitions from different collections of random projections permitted a higher-quality clustering than considering an individual hierarchy of low-density separators.

In addition, we found that for our choice of feature mapping, the performance of a hierarchy of low-density separators of the feature vectors did not significantly improve performance compared to locating low-density separators of the original observations. It is likely that a more rigorous approach to tuning the kernel parameter, or an alternative kernel function, would result in higher-quality partitions of the resulting feature vectors, but this is

beyond the scope of this work.

6.6 Conclusions

We proposed an approach for the location of low-density cluster separators using univariate random projections. We search over a finite collection of one-dimensional random subspaces for a set of univariate projections that approximately optimise criteria that may be indicative of the suitability of a set of projections for cluster separation. These criteria are related to the objectives of alternative projection techniques such as PCA, ICA and MDH. Subsequently, linear cluster boundaries are identified by bi-partitioning the data $\mathcal{X}$ at the minimiser of the estimated density of their projections onto the selected random vector. These bi-partitions are combined in a divisive hierarchical algorithm to locate a complete clustering of $\mathcal{X}$ and, through an appropriate stopping rule, also estimate the number of clusters. We remove the restriction to linear cluster boundaries by considering a non-linear mapping of the original observations to a set of feature vectors, upon which a linear separator allows the identification of non-linear cluster boundaries in the original data space.

Our approach only requires a single matrix multiplication (with a linear computational cost with respect to the number of observations and dimensions in $\mathcal{X}$) to compute the projections of $\mathcal{X}$ into the collection of random vectors. Therefore, this has a significantly lower computational cost than locating optimal univariate projections by PCA, ICA or MDH, all of which have a computational cost that is at least quadratic in the dimensionality of $\mathcal{X}$, so become computationally infeasible when $\mathcal{X}$ is very large and high-dimensional. Our approach also avoids recomputing the projections at each level of the hierarchy, making the computational advantage more significant when producing a complete clustering.

Through an empirical evaluation of the clustering performance of the proposed RP ap-

proach across simulated and real-world benchmark datasets, we find that RP allows the location of high-quality cluster separators, which are competitive with the separators located through alternative projection techniques for a much lower computational expenditure, and that the number of clusters may be estimated accurately. Furthermore, this approach converges to a high-quality clustering solution with relatively few random projections. We find that seeking projections with a strongly multimodal estimated density with a low minimiser between larger modes is the most appropriate optimality criterion for selecting random projections. This permits a lowest possible density separator (using a given collection of random projections) that also partitions dense regions associated with clusters. However, if $\mathcal{X}$ is very sparse, with a susceptibility to outliers, (which is often the case if $\mathcal{X}$ is a set of mapped feature vectors), this can lead to an overestimation of the number of clusters.

7

Random Projections with Alternative Clustering Objectives

Abstract

We investigate how random projection may be applied to locate non-linear separators of a dataset, which are consistent with the clustering objectives of k-means and spectral clustering. Our approach relies on univariate random projections of a set of non-linearly mapped feature vectors. These projections are used to locate a linear separator of the feature vectors, which corresponds to a non-linear separator of the original observations. We compute multiple univariate random projections, and bi-partition the feature vectors using the set of projections that permits the best separator based on optimality criteria which are consistent with the clustering objectives of k-means and spectral clustering. These bi-partitions are combined in divisive algorithms to locate a complete clustering of the data. We compare the quality of the partitions located through random projection to bisecting kernel k-means and hierarchical spectral clustering across a variety of real-world benchmark datasets. Our results show that univariate random projections can locate high-quality partitions, which are competitive with alternative divisive algorithms that are appropriate for clustering in the feature space.

7.1 Introduction

In the clustering problem, the set of observations $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ is partitioned into disjoint subsets or *clusters* such that within cluster similarity is maximised, and between cluster similarity is minimised. There exist multiple approaches to clustering, which rely on different definitions of similarity. Some of the most commonly applied clustering algorithms rely on the centroid-based and graph cut definitions. Centroid-based clustering seeks subsets of $\mathcal{X}$ which minimise the sum of squared distances between the observations and the centroid of their assigned cluster. Meanwhile, the graph cut problem formulation views the observations as nodes of an undirected, weighted graph with edge weights proportional to the pairwise similarity between observations, and aims to partition the graph such that edges with minimal weight are cut. These two approaches to clustering are adopted by the k-means and spectral clustering algorithms respectively, which are discussed in Section 2.1.2.

Both of these algorithms require the pairwise distances between the observations to define similarity, which is intuitive in low-dimensional datasets. However, as dimensionality increases, a fixed number of observations become increasingly sparse and high levels of noise can be introduced along dimensions which do not contain meaningful information for clustering (Steinbach et al., 2004) . Therefore, spatial proximity is less meaningful in such datasets, and defining clusters solely on distances between observations in inappropriate for accurate cluster identification. This motivates the search for low-dimensional subsets of $\mathcal{X}$, in which the clustering structure is apparent.

In Chapter 6, we considered the computationally efficient location of one-dimensional subspaces that are appropriate for cluster detection under the density-based cluster definition using random projection (RP) (Achlioptas, 2001). This work showed that RP can locate one-dimensional projections that allow accurate bi-partitions of clusters, and that

these bi-partitions can be combined in a divisive clustering algorithm to locate a complete clustering of $\mathcal{X}$. In this chapter, we propose to apply this RP approach in divisive clustering algorithms that locate one-dimensional subspaces which permit successive bi-partitions of $\mathcal{X}$ that are consistent with the centroid-based and graph cut cluster definitions, assumed by k-means and spectral clustering respectively.

Since partitions located using univariate orthogonal projections of $\mathcal{X}$ only allow the correct identification of linearly separable clusters, throughout this chapter, we apply our algorithms over a set of non-linearly mapped feature vectors, contained in $\mathcal{X}$, which have been projected onto an n-dimensional orthonormal basis of the feature space. Therefore, a linear separator of these mapped feature vectors corresponds to a non-linear separator of the original observations, allowing our approaches to identify non-linearly separable clusters. This makes the approaches proposed in this chapter comparable to alternative divisive algorithms which cluster feature vectors.

We conduct an empirical evaluation of the performance of the proposed divisive RP algorithms across a variety of real-world benchmark datasets. The performance of the proposed approach is compared to the performance of bisecting kernel k-means and spectral clustering, where partitions are located using the kernel matrix of pairwise inner products between the feature vectors, and therefore separate clusters based on the pairwise separation of the feature vectors, computed over all dimensions of the feature space. Since divisive algorithms using centroid-based and graph cut clustering to recursively bi-partition $\mathcal{X}$ do not offer an intuitive stopping rule, we provide the true number of clusters as an input parameter for all algorithms considered in this chapter.

The remainder of this chapter is organised as follows. Section 7.2 outlines the methodology for the proposed approach. Next, Section 7.3 provides experimental results for the

clustering performance of the proposed RP approach compared to alternative divisive algorithms, which recursively bi-partition the feature vectors across real-world benchmark datasets with varying characteristics. Finally the work is concluded in Section 7.4.

7.2 Methodology

In this section, we present the methodology for the proposed RP approach. This is similar to the methodology in Sections 6.2.3 and 6.2.4, so we omit a complete discussion of RP. Throughout this chapter, we assume that $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^n$ is a set of non-linearly mapped feature vectors, which have been projected into the n-dimensional space spanned by their kernel principal components. This is justified in Section 5.2 since any meaningful projections for clustering must lie within the span of the feature vectors. We define $\mathbf{R} = [\mathbf{r}_i]$ to be a matrix whose columns $\mathbf{r}_i$ for $i = 1, ..., r$ are a set of r random vectors sampled uniformly over the n-dimensional unit sphere. The univariate random projections of $\mathcal{X}$ onto the vectors in $\mathbf{R}$ are given by the columns of $\mathcal{X}^r = \{\mathbf{R}^\top \mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^r$.

7.2.1 Divisive Clustering with Univariate Random Projections

Given a collection of univariate random projections of $\mathcal{X}$, stored in $\mathcal{X}^r$, we propose to locate a bi-partition of $\mathcal{X}$ using the projections which are most appropriate for cluster separation, based on the centroid-based and graph-cut cluster definitions. These bi-partitions may be combined in a divisive algorithm to produce a complete clustering of $\mathcal{X}$ into k clusters. For this to be possible, we require rules to determine which cluster to split at each level of the hierarchy (selection rule), how to split this cluster (splitting rule) and when to terminate this procedure (stopping rule).

Selection Rule

For the approaches proposed in this chapter, we adapt the selection rule suggested in Section 6.2.4. We select the cluster, $\mathcal{X}_C$, whose projections onto one of the random vectors $\mathbf{r}_i$ best satisfies our specified optimality criterion $f(\cdot)$. This is defined formally in Eq. (6.2). Our choices of optimality criteria, which result in RP approaches that are related to bisecting kernel k-means and hierarchical spectral clustering algorithms are discussed later in Section 7.2.3.

Splitting Rule

We propose to bi-partition the set of observations assigned to the selected cluster, $\mathcal{X}_C$ using the univariate random projections of $\mathcal{X}_C$ which approximately optimise our specified optimality criterion, as detailed in Eqs. (6.5) - (6.6). For centroid-based and graph-cut clustering, we bi-partition the projections using 2-means and spectral clustering respectively, so the clusters located in the one-dimensional subspace are consistent with the relevant clustering objective.

Stopping Rule

Locating bi-partitions of the selected univariate random projections using 2-means or spectral clustering does not offer an intuitive termination rule, therefore we specify the desired number of clusters as an input to determine when the divisive procedure should terminate.

7.2.2 Combining RP Trees by Ensemble Clustering

As discussed in Section 6.2.5, different clustering results produced by using different collections of random projections may be combined using an ensemble clustering to locate a

singe partition, which combines information from all of the individual input clusterings. For the RP approaches proposed in this chapter, we apply the ensemble clustering of Dimitriadou et al. (2002), which returns the fuzzy clustering that minimises the sum of squared Euclidean distances to each of the m input partitions. Full details of this method are provided in Section 6.2.5.

7.2.3 Optimality Criteria to Select Random Projections

The optimality criteria which we propose to select the most appropriate univariate random projections for clustering based on the centroid-based and graph cut approaches to clustering are:

1. Minimum sum of squared euclidean distances between the univariate projections and their assigned cluster centroid, located by 2-means. This criterion is equivalent to the k-means cost function, and therefore selecting univariate projections which satisfy this optimality criterion permits the best bi-partition based on the centroid-based cluster definition.
2. Minimum second smallest eigenvalue of the normalised graph Laplacian (Ng et al., 2002). The graph Laplacian, as defined in Section 2.1.2, always has a smallest eigenvalue equal to zero. The second smallest eigenvalue measures the connectivity of the graph, where small values indicate that the graph has two components which are nearly disconnected, and therefore suggest that there are two distinct clusters. Selecting univariate projections that minimise this optimality criterion therefore permits a bi-partition which is consistent with the graph-cut approach to clustering.

Divisive clustering algorithms which select the most appropriate set of random projections for clustering at each level of the hierarchy based on these optimality criteria and subsequently bi-partition these projections using 2-means and spectral clustering are related to bisecting kernel k-means and hierarchical spectral clustering respectively.

7.2.4 Notation for RP Approaches

In Section 7.3, we use the following notation to refer to the RP approaches using varying numbers of random projections and different optimality criteria. RP-sse-r and RP-ev2-r correspond to locating a single hierarchy using a set of r univariate random projections, and selecting the set of projections with minimum 2-means cost function and minimum second smallest eigenvalue of the graph Laplacian respectively. When using multiple hierarchies, generated from different collections of random projections, we use the notation RP-sse-r-E-m and RP-ev2-r-E-m to refer to using m hierarchies, each of which use r random projections to search for the set of univariate projections with minimum 2-means cost function and minimum second smallest eigenvalue of the graph Laplacian respectively.

7.3 Experimental Results

In this section, we investigate the performance of the RP approaches proposed in Section 7.2 across real benchmark datasets with varying characteristics. Since we only consider the scenario where the original observations have been mapped into the feature space, and projected onto the kernel principal components, the dimensionality of the problem is defined by the number of observations and not the dimensionality of the original observations. Therefore, we consider the datasets summarised in Table 6.1. The performance of the RP approaches for centroid-based and graph cut clustering are compared to:

1. Bisecting kernel k-means. This algorithm recursively partitions the feature vectors using kernel 2-means in a divisive algorithm. At each level of the hierarchy, we select the cluster that maximises the sum of squared Euclidean distances between the feature vectors and their assigned cluster centroid. This cluster is then bi-partitioned using kernel 2-means. The details of this algorithm are discussed in Section 2.1.2. We are not aware of any method to automatically terminate this procedure, and therefore provide the true number of clusters as an input parameter. For each bi-partition, we use the implementation of kernel k-means in the `R` package `kernlab` which operates

directly on the kernel matrix of pairwise inner products of the feature vectors, not their projections onto the kernel principal components.

2. Hierarchical spectral clustering, where the set of original observations is recursively bi-partitioned using spectral clustering (von Luxburg, 2007) with the normalised graph Laplacian (Ng et al., 2002). At each level of the hierarchy, we split the cluster with minimal graph connectivity, measured by the value of the second smallest eigenvalue of the graph Laplacian as defined in Section 2.1.2. Since the computation of the kernel matrix is an inherent part of the spectral clustering algorithm, this method is implemented using the kernel matrix and not the n-dimensional mapped observations $\mathcal{X}$. This algorithm requires the true number of clusters as an input parameter. For the each bi-partition using spectral clustering, we use the implementation in the `kernelab` package for `R`.

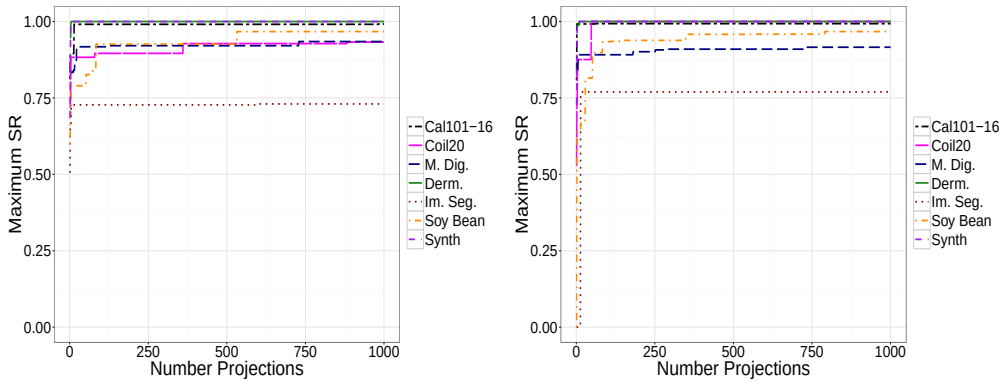
7.3.1 Details of Implementation

As for all kernel-based approaches, the choice of kernel function and any subsequent parameter values critically affect the performance of all the algorithms implemented in this chapter. This is a well-documented, open problem in the literature and as such a robust approach to determine an optimal choice of kernel is beyond the scope of our work. We use the Gaussian kernel, since this is the most widely used in the literature. To tune the kernel parameter, we use the local scaling approach proposed by Zelnik-Manor and Perona (2004),

$$\kappa(\mathbf{y}_i, \mathbf{y}_j) = \exp\left\{-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|^2}{s_i s_j}\right\}$$

where the $\mathbf{y}_i$ are the original observations (before the feature mapping) and s_i and s_j are the distances from the ith and jth original observations to their seventh nearest neighbours respectively. This can handle data on multiple scales and is very effective in our experience.

We use the same kernel matrix for all algorithms considered either directly (for bisecting kernel k-means and hierarchical spectral clustering) or to compute the projections of the



(a) 2-means (b) Spectral

Figure 7.1: Increase in success ratio for a bi-partition of univariate random projections of the feature vectors using 2-means and spectral clustering with increasing number of random projections for real benchmark datasets summarised in Table 6.1.

feature vectors onto their kernel principal components, stored in $\mathcal{X}$, which is used for the RP approaches. We also use this local scaling approach to construct the adjacency matrix of the graph for the RP approach with the minimal second smallest eigenvalue of the graph Laplacian optimality criterion, and for the selection rule in hierarchical spectral clustering.

For the RP approaches, we experimented using varying numbers of random projections to search for an appropriate set of projections for clustering at each stage of the divisive algorithm. Figure 7.1 provides the improvement in the clustering performance (measured by the success ratio) of a bi-partition of the mapped feature vectors for some the datasets considered with an increasing number of random projections, over which to search for an appropriate partition. Figures 7.1(a) and 7.1(b) correspond to bi-partitioning each successive random projection using 2-means and spectral clustering respectively and retaining the bi-partition with the current best success ratio, as defined in Eq. (6.9). Results for the other datasets showed a similar pattern so are omitted for clarity.

We found that in the feature space, a high-quality bi-partition was located with only a small number of random projections. For some of the datasets, the clustering performance converged very quickly, and in all cases the increase in clustering performance is negligible

for large numbers of random projections. For the empirical evaluation of the performance of complete divisive clustering results, located through the proposed RP approaches, we experimented using 100, 500 and 1,000 random projections, and found that the clustering performance when using 500 projections was always between the two more extreme cases, so we omit these results for brevity in Section 7.3.2. For each dataset, we located 30 complete clusterings, using different collections of random projections, and the ensemble results presented were computed using these as input partitions.

7.3.2 Performance Evaluation on Real Datasets

In this section, we evaluate the performance of the RP approaches using the minimum 2-means cost function and the minimum second smallest eigenvalue of the graph Laplacian optimality criteria. The clustering performance of divisive algorithms which partition $\mathcal{X}$ using these approximately optimal sets of univariate random projections is compared to the performance of bisecting kernel k-means and hierarchical spectral clustering, which locate splits at each level of the hierarchy using the kernel matrix of pairwise inner products between the feature vectors, and therefore consider information from all dimensions of the feature vectors for a bi-partition.

Figures 7.2 and 7.3 show boxplots of the clustering performance of the RP approaches proposed in this chapter for the real datasets considered over 30 cluster hierarchies, each of which use a different collection of 100 and 1,000 random projections respectively. For these datasets and our choice of feature mapping, there is not a substantial difference in performance between searching over 100 and 1,000 random projections for the most appropriate cluster separator, suggesting that high-quality separators may be located by only considering relatively small numbers of projections, as indicated by Figures 7.1(a) and 7.1(b).

For the majority of these datasets, taking an ensemble over the 30 hierarchies produced by

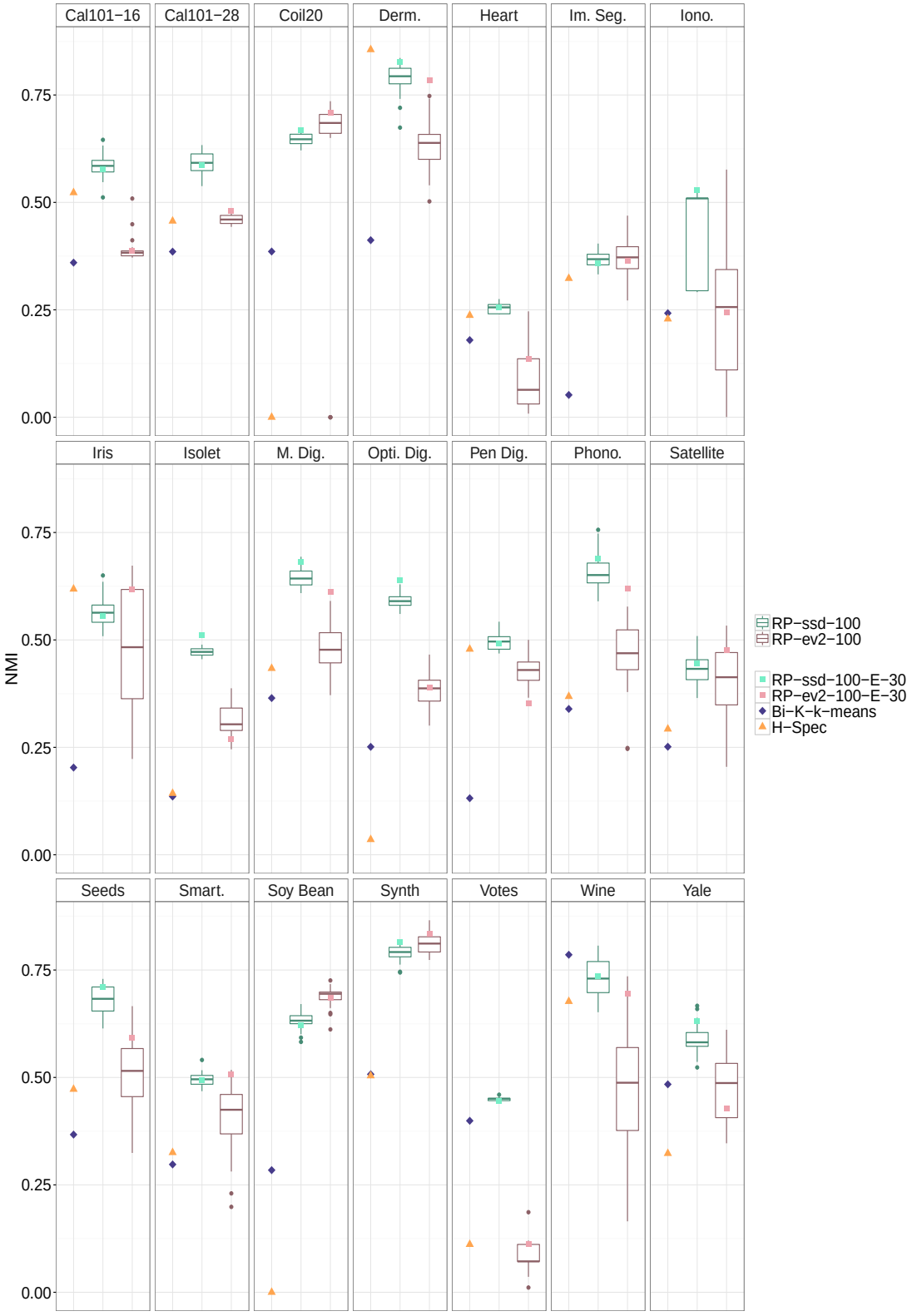


Figure 7.2: Boxplots of clustering performance of RP approaches using 100 projections, bisecting kernel k-means and hierarchical spectral clustering over mapped feature vectors of real datasets.

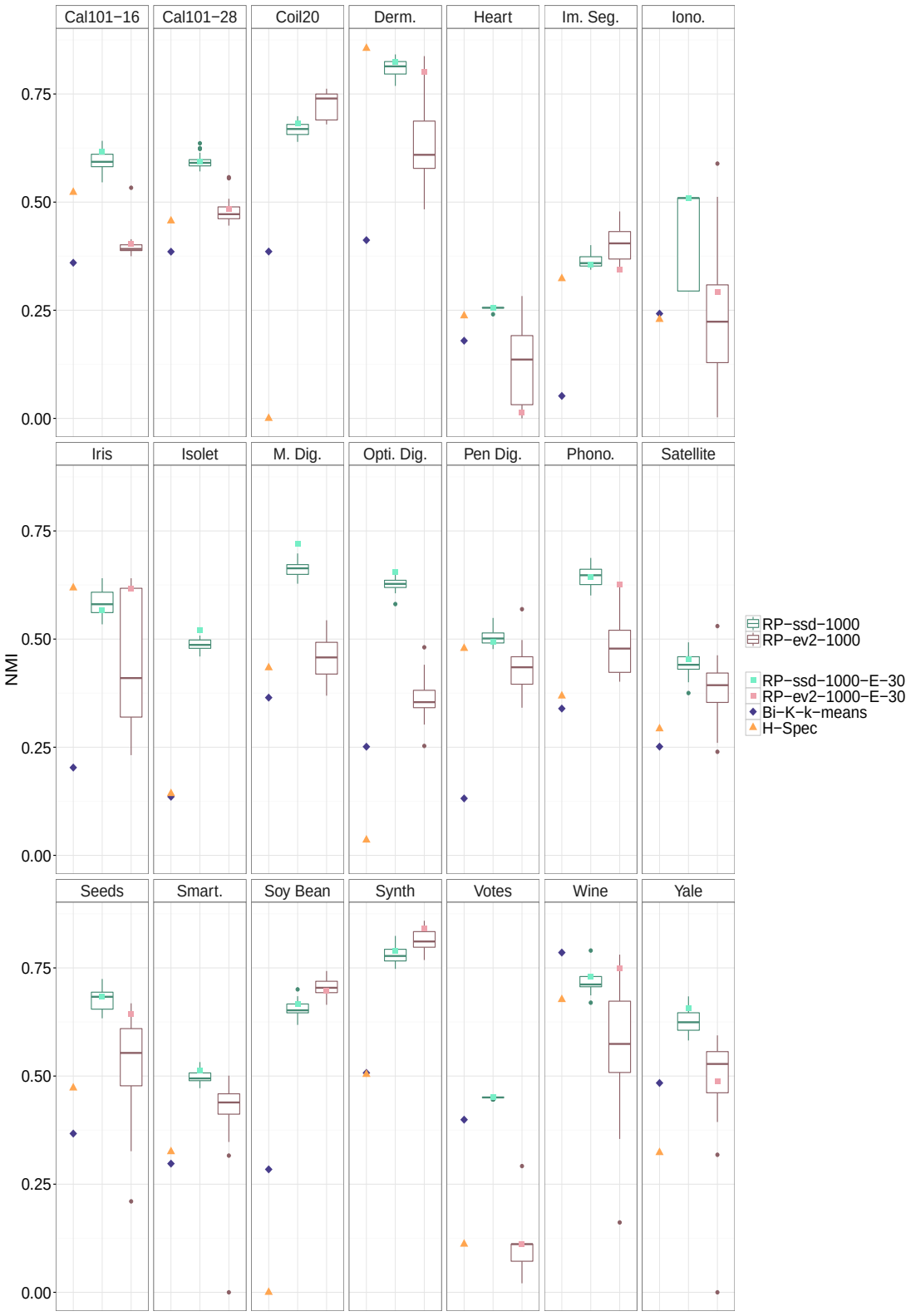


Figure 7.3: Boxplots of clustering performance of RP approaches using 1,000 projections, bisecting kernel k-means and hierarchical spectral clustering over mapped feature vectors of real datasets.

different random projections resulted in higher performance than the average performance of the input partitions. For these datasets, the RP approaches perform very well, frequently locating higher quality partitions than bisecting kernel k-means and hierarchical spectral clustering. RP with the minimum sum of squared distances to the 2-means centroids optimality criterion generally performs better than RP with the minimum second smallest eigenvalue of the graph Laplacian optimality criterion. In addition, the necessity to compute the graph Laplacian and its eigenvalues makes this optimality criterion more computationally expensive, and therefore less attractive than the minimum sum of squared distances to the 2-means centroids.

7.4 Conclusions

In this chapter, we investigated how univariate random projections may be applied to produce divisive hierarchical clustering algorithms, which locate cluster separators in one-dimensional subspaces of the feature space. Our proposed approach involves generating multiple random univariate projections and separating the set of mapped feature vectors $\mathcal{X}$ using the set of projections which is most appropriate for cluster identification based on optimality criteria that are related to the objectives of k-means and spectral clustering.

We compared the clustering performance of this RP approach to the clustering results from bisecting kernel k-means and hierarchical spectral clustering across the mapped feature vectors of a variety of real benchmark datasets. Our results indicate that cluster separators computed using approximately optimal one-dimensional subspaces, located by RP permits high-quality clustering results. The proposed approach often outperforms hierarchical algorithms that bi-partition the feature vectors using the kernel matrix directly, and therefore consider information from all dimensions of the feature space to locate a cluster separator.

8

Conclusion

8.1 Summary of Contributions

This thesis developed methodology for the identification of groups of similar objects (clusters) in datasets with large numbers of diverse features. Although clustering has a wide variety of application areas and a rich literature, high-dimensional, mixed datasets pose a significant challenge for the majority of clustering algorithms. The algorithms proposed in this thesis can locate, and estimate the number of clusters in high-dimensional datasets, whose features may contain mixed data types with non-linearly separable clusters. Our algorithms locate minimum density linear cluster separators using optimal one-dimensional projections of the data, and therefore avoid the challenges associated with cluster detection in high-dimensional spaces. For mixed datasets, we transform the original dataset to an appropriate continuous representation, upon which clustering is performed. The restriction to linear separators is lifted by considering a non-linear feature mapping of the original observations, such that a linear separator in the feature space can correctly identify non-linearly separable clusters in the original data space. The computation of optimal projections for clustering becomes expensive in very large, high-dimensional datasets, so we further propose techniques for the location of approximately optimal univariate projections for cluster separation using random projection.

In Chapter 4, a hierarchical divisive and a partitional clustering algorithm are proposed, both of which combine bi-partitions from minimum density hyperplane separators to lo-

cate an overall clustering of the data, while estimating the number of clusters. These algorithms rely on the density-based approach to clustering, and identify high-density clusters by defining low-density cluster boundaries, that separate regions of high probability density, associated with clusters. These low-density cluster separators are computed by globally or locally minimising the integral of the density on the hyperplane, which may be evaluated exactly using the estimated density of the one-dimensional projections of the data onto the vector normal to the hyperplane, making this approach applicable in high dimensions. For mixed datasets, an appropriate continuous representation is sought, extending the applicability of this approach to datasets with non-continuous attributes. The proposed algorithms can accurately identify clusters in arbitrarily oriented subspaces, and estimate their number. Of the two approaches, the divisive clustering algorithm provides the most competitive clustering performance, frequently producing higher quality partitions than alternative density-based and state-of-the-art clustering algorithms over simulated and real-world benchmark datasets.

The divisive clustering algorithm proposed in Chapter 4 is extended to feature spaces in Chapter 5. Through a non-linear mapping of the original data into the feature space, this extension permits a hyperplane separator to identify non-linear cluster boundaries in the space of the original observations. Since the density on a hyperplane is evaluated using the inner product between the data and the vector normal to the hyperplane, it is possible to formulate the problem of locating a hyperplane with minimal density in the feature space using the kernel matrix of pairwise inner products between the feature vectors, without explicit computation of the, potentially infinite-dimensional, mapped vectors. The search space for the minimum density hyperplane in the feature space is practically restricted to the n-dimensional space spanned by the feature vectors, where n is the number of observa-

tions. Therefore, the projections of the feature vectors onto the orthonormal basis formed by kernel principal component analysis are used to locate a minimum density separator of the feature vectors. For large datasets, searching over all n dimensions in the span of the feature vectors for the one-dimensional projection vector that permits the minimum density separator becomes computationally expensive, so the location of an approximate minimum-density separator is considered, by restricting the search to a lower-dimensional subspace, that excludes dimensions which are unlikely to contain meaningful information for clustering. A divisive clustering algorithm which locates successive bi-partitions of the feature vectors using a minimum density separator at each level of the hierarchy allows a complete clustering. The proposed approach has competitive performance to alternative clustering algorithms that are appropriate for clustering feature vectors across a variety of real-world benchmark datasets.

Chapter 6 presents an approach for the computationally efficient location of approximately optimal one-dimensional projections for low-density cluster separation. The computation of optimal projections for cluster identification through the minimum density hyperplane algorithms proposed in Chapters 4 and 5, or alternative projection techniques such as principal component analysis or independent component analysis have a high computational cost when applied to large, high-dimensional datasets (or their mapped feature vectors). Therefore, in Chapter 6, random projection is applied to compute a collection of univariate projections, which may be used to search for an projection that approximately optimises an appropriate criterion, quantifying the suitability of a set of univariate projections for cluster identification. The computation of these random projections only requires a single matrix multiplication, so is much more efficient than the alternative projection techniques considered. Therefore, the proposed approach locates a complete hierarchy of low-

density cluster separators significantly faster than techniques that seek globally optimal one-dimensional subspaces at each level of the hierarchy. Further, the clustering performance of the partitions located through random projection is competitive with the performance of the projections located through the aforementioned projection techniques across simulated and real-world datasets with varying characteristics.

8.2 Further Work

8.2.1 Tuning the Kernel

The methods proposed in Chapters 5 and 6 and Chapter 7, which locate clusters using non-linearly mapped feature vectors, all rely on the appropriate selection and tuning of the kernel function, used to construct the kernel matrix of pairwise inner products between the feature vectors. As for all kernel-based clustering algorithms, these choices change the structure present in the feature vectors, and therefore critically affect the clustering performance of the approaches proposed in this thesis, as well as the alternative algorithms considered as a comparison. Throughout this thesis, when construing a kernel matrix, the Gaussian (radial basis) kernel function is applied,

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right\}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two original observations and σ is a tuning parameter. This is the most widely applied kernel function in the literature. However, the choice of σ which is most suitable for cluster identification in the feature space is heavily dependent on the dataset of interest and tuning this parameter is a non-trivial problem. Throughout this thesis, we

apply the local scaling approach proposed by Zelnik-Manor and Perona (2004),

$$\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{s_i s_j}\right\}$$

where s_i and s_j are the distances from the ith and jth observations to their seventh nearest neighbours respectively. This approach permits clusters on multiple scales and in our experience is effective, with very poor clustering results in the associated feature space being rare. However, the selection of the number of nearest neighbours upon which s_i and s_j are computed is arbitrary, and impossible to justify theoretically.

More recently Huang et al. (2015) proposed an approach to tuning the scaling parameter, which is robust to noise and clusters with different densities. This approach employs a diffusion-based aggregated heat kernel to model the heat diffusion of the clusters and improve robustness in datasets with various types and levels of noise. Further, a local density affinity transformation is applied to model the local densities in each of the clusters, and therefore permit the location of clusters on different scales. The results presented by Huang et al. (2015) indicate that this approach is highly effective, and offers greater stability, robustness and superior clustering performance than alternative tuning techniques, when applied for spectral clustering. As further work, we would like to consider the application of this approach for the kernel-based algorithms proposed in this thesis, to investigate any possible improvements in clustering performance compared to the more straightforward scaling of Zelnik-Manor and Perona (2004).

As an extension to this, it may be appropriate to consider alternative choices of kernel function, such as non-parametric kernels, which aim to avoid the challenges associated with tuning parameters entirely. Examples of such kernels include Isomap (Tenenbaum et al., 2000) and the connectivity kernel (Fischer et al., 2004). Both of these approaches operate

on the graph $\mathcal{G}(\mathcal{X}, \mathcal{E})$ with nodes $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ and edge weights $\mathcal{E}$ proportional to the pairwise distances between the observations. For Isomap, the kernel matrix $\mathbf{K} \in \mathbb{R}^{n \times n}$ is defined such that each element K_{ij} is the shortest path between nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\mathcal{G}(\mathcal{X}, \mathcal{E})$. For the connectivity kernel, Fischer et al. (2004) define the effective dissimilarity along any possible path between nodes $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\mathcal{G}(\mathcal{X}, \mathcal{E})$ as the maximum edge weight between any two nodes along this path. Then, each element of the kernel matrix K_{ij} is the minimum effective dissimilarity along any of the possible paths between nodes $\mathbf{x}_i$ and $\mathbf{x}_j$. It is possible to show that this may be computed relatively efficiently using Kruskal's minimum spanning tree algorithm.

Initial experiments across small toy datasets indicate that these approaches can be effective, and permit high-quality cluster separators in the resulting feature spaces. However, the computational cost of locating shortest paths and minimum spanning trees over large graphs makes the computation of these kernel matrices expensive for large datasets, and approximation techniques would be required.

8.2.2 Multi-Objective Optimisation for Random Projection Selection

For the random projection algorithms proposed in Chapter 6 and Chapter 7, we considered a single optimality criterion to select the most suitable set of univariate projections for cluster separation at each stage of the divisive procedure. However, it is possible to simultaneously consider multiple optimality criteria to quantify the appropriateness of a set of projections for cluster identification. In this case, given a set of observations (or feature vectors) assigned to the cluster which is to be separated, $\mathcal{X}_C \subset \mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ and the matrix of columnwise random vectors $\mathbf{R} = [\mathbf{r}_i]$ for $i = 1, \ldots, r$, the set of projections upon which the

bi-partition of $\mathcal{X}_C$ is determined would be,

$$\mathbf{p}_C = \mathbf{X}_C \cdot \mathbf{r}^\star$$

$$\mathbf{r}^\star = \underset{\mathbf{r}_i \,;\, i \in \{1,\ldots,r\}}{\arg\max} \; \{f_1(\mathbf{X}_C \cdot \mathbf{r}_i), \ldots, f_m(\mathbf{X}_C \cdot \mathbf{r}_i)\}$$

where each of the $f_j(\cdot)$ for $j = 1, \ldots, m$ are different optimality criteria quantifying the suitability of a set of univariate projections for clustering and $\mathbf{X}_C$ is the data matrix associated with $\mathcal{X}_C$. It is highly unlikely that the m different optimality criteria will be in agreement as to the best set of univariate projections, and therefore, selecting a final projection vector is a non-trivial problem, for which many possible solutions will exist. A further extension may be to weight the different optimality criteria, depending on some measure of the suitability of the resulting univariate projections for cluster identification, such as a measure of compactness in the clusters or a measure of separation between the clusters.

8.2.3 Alternative Splitting Rule for Random Projections

In addition to the consideration of alternative optimality criteria to select appropriate random projections for clustering, it may also be advantageous for the RP approaches proposed in Chapter 6 and Chapter 7 to investigate alternative rules for splitting $\mathcal{X}$ based on the selected set of univariate random projections. Peña and Prieto (2001) propose a splitting rule that is related to our approach taken in Chapters 4, 5 and 6 in the sense that their approach splits a set of projections if there is significant evidence that they have a distribution with more than one mode. However, the approach proposed in Peña and Prieto (2001) avoids constructing an estimated density, and instead searches for a significant gap in the set of ordered univariate projections $p_1 \leqslant \ldots \leqslant p_n$. A gap is considered sufficiently large to indicate a valid partition if it has a very low probability of appearing in that position under the

assumption that the projections are sampled from a univariate normal distribution. If the projections (scaled to have zero mean and unit variance) do follow a univariate normal distribution, the transformed projections, $\Phi^{-1}(p_1), \ldots, \Phi^{-1}(p_n)$ where $\Phi^{-1}(\cdot)$ is the inverse normal distribution function will follow a Uniform$(0, 1)$ distribution. Therefore, the expected gap between any successive transformed projections is $\frac{1}{n+1}$. Hence any gaps between successive transformed projections that are significantly larger than this are indicative of a multi-modal structure in the projections, suggesting that the data should be separated.

Initial experiments indicate that this is an effective splitting criterion. This approach avoids the computation of an estimated density, and any potential sensitivity to the tuning of the bandwidth parameter or the relative depth threshold applied in our approaches. However, the specification of an appropriate threshold for a significantly large gap for cluster separation is not straightforward. Some potential values are suggested in Peña and Prieto (2001), although this would require further investigation.

8.2.4 Higher-Dimensional Subspaces for Random Projection

As an alternative to using RP to search over one-dimensional subspaces for an approximately optimal projection for low-density cluster separation, we could instead project $\mathcal{X} \subset \mathbb{R}^d$ into an r-dimensional subspace for $r \ll d$. Thereafter, a hierarchy of minimum density cluster separators can be sought over the projections of $\mathcal{X}$ into this subspace, $\mathcal{X}^r = \{\mathbf{R}^\top \mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^r$ where $\mathbf{R} \in \mathbb{R}^{d \times r}$ is a random orthogonal matrix. A new subspace could be generated at each level of the cluster hierarchy, or for further computational efficiency, the same random subspace could be retained throughout the divisive clustering. This approach is closely related to the work of Avogadri and Valentini (2009); Bingham and Mannila (2001); Goal et al. (2005); Fern and Brodley (2003); Tasoulis et al. (2012), where clusters are located, using different algorithms, in low-dimensional random subspaces.

References

Achlioptas, D. (2001). Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems*, pages 274–281. ACM.

Aggarwal, C. C., Wolf, J. L., Yu, P. S., Procopiuc, C., and Park, J. S. (1999). Fast algorithms for projected clustering. In *ACM SIGMoD Record*, volume 28, pages 61–72. ACM.

Aggarwal, C. C. and Yu, P. S. (2000). *Finding generalized projected clusters in high-dimensional spaces*, volume 29. ACM.

Agrawal, R., Gehrke, J., Gunopulos, D., and Raghavan, P. (1998). *Automatic subspace clustering of high-dimensional data for data mining applications*, volume 27. ACM.

Ahmad, A. and Dey, L. (2007). A k-means clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*, 63(2):503–527.

Ahmad, A. and Dey, L. (2011). A k-means type clustering algorithm for subspace clustering of mixed numeric and categorical datasets. *Pattern Recognition Letters*, 32(7):1062–1069.

Aitnouri, E., Wang, S., and Ziou, D. (2000). On comparison of clustering techniques for histogram pdf estimation. *Pattern Recognition and Image Analysis*, 10(2):206–217.

Arthur, D. and Vassilvitskii, S. (2007). *k*-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035.

Avogadri, R. and Valentini, G. (2009). Fuzzy ensemble clustering based on random projections for DNA microarray data analysis. *Artificial Intelligence in Medicine*, 45(2):173–183.

Azzalini, A. and Menardi, G. (2014). Clustering via nonparametric density estimation: The R package pdfCluster. *Journal of Statistical Software*, 57(11):1–26.

Azzalini, A. and Menardi, G. (2016). Density-based clustering with non-continuous data. *Computational Statistics*, pages 1–28.

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

Bach, F. R. and Jordan, M. I. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, 3:1–48.

Beeferman, D. and Berger, A. (2000). Agglomerative clustering of a search engine query log. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 407–416. ACM.

Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159.

Ben-David, S., Lu, T., Pál, D., and Sotáková, M. (2009). Learning low-density separators. In van Dyk, D. and Welling, M., editors, *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, JMLR Workshop and Conference Proceedings, pages 25–32, Florida, USA.

Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping Multidimensional Data*, pages 25–71. Springer.

Bezdek, J. C. (2013). *Pattern recognition with fuzzy objective function algorithms*. Springer Science & Business Media.

Bingham, E. and Mannila, H. (2001). Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 245–250. ACM.

Böhm, C., Kailing, K., Kröger, P., and Zimek, A. (2004a). Computing clusters of correlation connected objects. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, pages 455–466. ACM.

Böhm, C., Railing, K., Kriegel, H.-P., and Kröger, P. (2004b). Density connected clustering with local subspace preferences. In *Data Mining, 2004. ICDM'04. Fourth IEEE International Conference on*, pages 27–34. IEEE.

Boley, D. (1998). Principal direction divisive partitioning. *Data Mining and Knowledge Discovery*, 2(4):325–344.

Borg, I. and Groenen, P. J. (2005). *Modern Multidimensional Scaling: Theory and Applications*. Springer Science & Business Media.

Burke, J. V., Lewis, A. S., and Overton, M. L. (2005). A robust gradient sampling algorithm for nonsmooth, nonconvex optimization. *SIAM Journal on Optimization*, 15(3):751–779.

Campello, R. J., Moulavi, D., Zimek, A., and Sander, J. (2013). A framework for semi-supervised and unsupervised optimal extraction of clusters from hierarchies. *Data Mining and Knowledge Discovery*, 27(3):344–371.

Cardoso, J.-F. and Souloumiac, A. (1993). Blind beamforming for non-gaussian signals. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, pages 362–370. IET.

Cardoso, J.-F. and Souloumiac, A. (1996). Jacobi angles for simultaneous diagonalization. *SIAM Journal on Matrix Analysis and Applications*, 17(1):161–164.

Celeux, G. and Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational statistics & Data analysis*, 14(3):315–332.

Chacón, J. E. et al. (2015). A population background for nonparametric density-based clustering. *Statistical Science*, 30(4):518–532.

Chapelle, O., Schölkopf, B., and Zien, A. (2006). *Semi-Supervised Learning*. Adaptive computation and machine learning. MIT Press.

Chapelle, O. and Zien, A. (2005). Semi-supervised classification by low density separation. In Cowell, R. G. and Ghahramani, Z., editors, *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 57–64. Society for Artificial Intelligence and Statistics.

Cuevas, A., Febrero, M., and Fraiman, R. (2000). Estimating the number of clusters. *Canadian Journal of Statistics*, 28(2):367–382.

Cuevas, A., Febrero, M., and Fraiman, R. (2001). Cluster analysis: a further approach based on density estimation. *Computational Statistics and Data Analysis*, 36(4):441–459.

Dasgupta, S. (2000). Experiments with random projection. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, UAI'00, pages 143–151, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

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)*, pages 1–38.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 551–556. ACM.

Diaconis, P. and Freedman, D. (1984). Asymptotics of graphical projection pursuit. *The Annals of Statistics*, pages 793–815.

Dimitriadou, E., Weingessel, A., and Hornik, K. (2002). A combination scheme for fuzzy clustering. *International Journal of Pattern Recognition and Artificial Intelligence*, 16(07):901–912.

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 2nd International Conference on Knowledge Discovery and Data Mining (KDD '96)*, volume 96, pages 226–231. AAAI Press.

Estivill-Castro, V. and Yang, J. (2000). Fast and robust general purpose clustering algorithms. In *Pacific Rim International Conference on Artificial Intelligence*, pages 208–218. Springer.

Everitt, B. S. (1988). A finite mixture model for the clustering of mixed-mode data. *Statistics & Probability Letters*, 6(5):305–309.

Everitt, B. S., Landau, S., Leese, M., and Stahl, D. (2011). *Cluster Analysis*. Wiley Series in Probability and Statistics. John Wiley & Sons, 5th edition.

Fern, X. Z. and Brodley, C. E. (2003). Random projection for high-dimensional data clustering: a cluster ensemble approach. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 186–193.

Fischer, B., Roth, V., and Buhmann, J. M. (2004). Clustering with the connectivity kernel. In *Advances in Neural Information Processing Systems*, volume 16, pages 89–96.

Forgy, E. (1965). Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21:768––769.

Fowlkes, C., Belongie, S., Chung, F., and Malik, J. (2004). Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225.

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.

Fränti, P. and Virmajoki, O. (2006). Iterative shrinking method for clustering problems. *Pattern Recognition*, 39(5):761–775.

Friedman, J. and Stuetzle, W. (1981a). Projection pursuit classification. *Unpublished manuscript*.

Friedman, J. H. and Stuetzle, W. (1981b). Projection pursuit regression. *Journal of the American Statistical Association*, 76(376):817–823.

Friedman, J. H., Stuetzle, W., and Schroeder, A. (1984). Projection pursuit density estimation. *Journal of the American Statistical Association*, 79(387):599–608.

Friedman, J. H. and Tukey, J. W. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, 100(9):881–890.

Gan, G., Ma, C., and Wu, J. (2007). *Data Clustering: Theory, Algorithms, and Applications*. SIAM.

Georghiades, A. S., Belhumeur, P. N., and Kriegman, D. J. (2001). From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):643–660.

Goal, N., Bebis, G., and Nefian, A. (2005). Face recognition experiments with random projection. In *Proceedings SPIE*, volume 5779, pages 426–437.

Gower, J. C. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, pages 857–871.

Guha, S., Rastogi, R., and Shim, K. (1998). CURE: an efficient clustering algorithm for large databases. In *ACM SIGMOD Record*, volume 27, pages 73–84. ACM.

Guha, S., Rastogi, R., and Shim, K. (1999). ROCK: a robust clustering algorithm for categorical attributes. In *Data Engineering, 1999. Proceedings., 15th International Conference on*, pages 512–521. IEEE.

Hagen, L. and Kahng, A. B. (1992). New spectral methods for ratio cut partitioning and clustering. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 11(9):1074–1085.

Hameed, M. A., Ramachandram, S., and Al Jadaan, O. (2012). Clustering dependant recommender systems. *International Journal of Simulation–Systems, Science & Technology*, 13(6).

Hartigan, J. (1977). Distribution problems in clustering. *Classification and Clustering*, pages 45–72.

Hartigan, J. A. (1975). *Clustering Algorithms*. Wiley Series in Probability and Mathematical Statistics. Wiley, New York.

Hartigan, J. A. and Hartigan, P. M. (1985). The dip test of unimodality. *The Annals of Statistics*, 13(1):70–84.

Hastie, T., Buja, A., and Tibshirani, R. (1995). Penalized discriminant analysis. *The Annals of Statistics*, pages 73–102.

Hecht-Nielsen, R. (1994). Context vectors: general purpose approximate meaning representations self-organized from raw data. *Computational Intelligence: Imitating Life*, pages 43–56.

Hinneburg, A. and Keim, D. A. (1999). Optimal grid-clustering: towards breaking the curse of dimensionality in high-dimensional clustering.

Hruschka, H. (1986). Market definition and segmentation using fuzzy clustering methods. *International Journal of Research in Marketing*, 3(2):117–134.

Huang, H., Yoo, S., Yu, D., and Qin, H. (2015). Density-aware clustering based on aggregated heat kernel and its transformation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 9(4):29.

Huang, Z. (1997). Clustering large data sets with mixed numeric and categorical values. In *Proceedings of the 1st Pacific-Asia Conference on Knowledge Discovery and Data Mining,(PAKDD)*, pages 21–34. Singapore.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 2(3):283–304.

Huber, P. J. (1985). Projection pursuit. *The Annals of Statistics*, 13(2):435–475.

Hubert, L. (1973). Monotone invariant clustering procedures. *Psychometrika*, 38(1):47–62.

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

Hyvärinen, A., Karhunen, J., and Oja, E. (2004). *Independent Component Analysis*, volume 46. John Wiley & Sons.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys (CSUR)*, 31(3):264–323.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *Proceedings of International Conference on Machine Learning (ICML)*, volume 99, pages 200–209, Bled, Slowenien.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254.

Johnson, W. B. and Lindenstrauss, J. (1984). Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 26(189-206):1.

Jones, M. C. and Sibson, R. (1987). What is projection pursuit? *Journal of the Royal Statistical Society. Series A (General)*, pages 1–37.

Kailing, K., Kriegel, H.-P., and Kröger, P. (2004). Density-connected subspace clustering for high-dimensional data. In *Proceedings of SIAM International Conference on Data Mining*, pages 246–257.

Karatzoglou, A., Smola, A., Hornik, K., and Zeileis, A. (2004). kernlab: an S4 package for kernel methods in R.

Kärkkäinen, I. and Fränti, P. (2002). *Dynamic Local Search Algorithm for the Clustering Problem*. University of Joensuu.

Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley-Interscience, 9th edition.

Khan, M. E., Bouchard, G., Murphy, K. P., and Marlin, B. M. (2010). Variational bounds for mixed-data factor analysis. In *Advances in Neural Information Processing Systems 23*, pages 1108–1116.

King, B. (1967). Step-wise clustering procedures. *Journal of the American Statistical Association*, 62(317):86–101.

Krause, A. and Liebscher, V. (2005). Multimodal projection pursuit using the dip statistic. *Preprint-Reihe Mathematik*, 13.

Kriegel, H.-P., Kröger, P., Sander, J., and Zimek, A. (2011). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240.

Kriegel, H.-P., Kröger, P., and Zimek, A. (2009). Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(1):1.

Krishna, K. and Murty, M. N. (1999). Genetic k-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3):433–439.

Kuczyński, J. and Woźniakowski, H. (1992). Estimating the largest eigenvalue by the power and lanczos algorithms with a random start. *SIAM Journal on Matrix Analysis and Applications*, 13(4):1094–1122.

Lewis, A. and Overton, M. (2013). Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, 141:135–163.

Lichman, M. (2013). UCI machine learning repository. `http://archive.ics.uci.edu/ml`.

Lloyd, S. (1957). Least square quantization in PCM. Bell Telephone Laboratories Paper.

Lutkepohl, H. (1997). Handbook of matrices. *Computational Statistics and Data Analysis*, 2(25):243.

Macnaughton-Smith, P., Williams, W., Dale, M., and Mockett, L. (1964). Dissimilarity analysis: a new technique of hierarchical sub-division.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281 – 297.

Marlin, B. M. (2014). Machine learning. https://people.cs.umass.edu/~marlin/index.shtml.

Masulli, F. and Schenone, A. (1999). A fuzzy clustering based segmentation system as support to diagnosis in medical imaging. *Artificial Intelligence in Medicine*, 16(2):129–147.

McLachlan, G. and Peel, D. (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley-Interscience, 1st edition.

Menardi, G. (2016). A review on modal clustering. *International Statistical Review*, 84(3):413–433.

Menardi, G. and Azzalini, A. (2014). An advancement in clustering via nonparametric density estimation. *Statistics and Computing*, 24(5):753–767.

Morariu, V. I., Srinivasan, B. V., Raykar, V. C., Duraiswami, R., and Davis, L. S. (2009). Automatic online tuning for fast gaussian summation. In *Advances in Neural Information Processing Systems*, pages 1113–1120.

Muller, K.-R., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201.

Nene, S. A., Nayar, S. K., and Murase, H. (1996). Columbia object image library (coil-20). http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

Ng, A., Jordan, M. I., and Weiss, Y. (2002). On spectral clustering: analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849 –856. MIT Press, Cambridge.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer Science & Business Media.

Parsons, L., Haque, E., and Liu, H. (2004). Subspace clustering for high-dimensional data: a review. *ACM SIGKDD Explorations Newsletter*, 6(1):90–105.

Patané, G. and Russo, M. (2001). The enhanced LBG algorithm. *Neural Networks*, 14(9):1219–1237.

Pavlidis, N., Hofmeyr, D., and Tasoulis, S. (2016). Minimum density hyperplanes. *Journal of Macnine Learning Research*, 17(156):1–33.

Peña, D. and Prieto, F. J. (2001). Cluster identification using projections. *Journal of the American Statistical Association*, 96(456):1433–1445.

Polito, M. and Perona, P. (2002). Grouping and dimensionality reduction by locally linear embedding. In *Advances in Neural Information Processing Systems*, pages 1255–1262.

Portnoy, L., Eskin, E., and Stolfo, S. (2001). Intrusion detection with unlabeled data using clustering. In *In Proceedings of ACM CSS Workshop on Data Mining Applied to Security (DMSA)*. Citeseer.

Puzicha, J., Hofmann, T., and Buhmann, J. (1999). A theory of proximity based clustering: structure detection by optimization. *Pattern Recognition*, 33(4):617–634.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850.

Rinaldo, A. and Wasserman, L. (2010). Generalized density clustering. *The Annals of Statistics*, 38(5):2678–2722.

Roberts, S. and Everson, R. (2001). *Independent Component Analysis: Principles and Practice*. Cambridge University Press.

Rosenberg, A. and Hirschberg, J. (2007). V-measure: a conditional entropy-based external cluster evaluation measure. In *Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, volume 7, pages 410–420.

Roth, V., Laub, J., Kawanabe, M., and Buhmann, J. M. (2003). Optimal cluster preserving embedding of nonmetric proximity data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(12):1540–1551.

Roux, M. (1991). Basic procedures in hierarchical cluster analysis. In *Applied Multivariate Analysis in SAR and Environmental Studies*, pages 115–135. Springer.

Roux, M. (1995). About divisive methods in hierarchical clustering, data science and its applications.

Rubinstein, R. (1982). Generating random vectors uniformly distributed inside and on the surface of different regions. *European Journal of Operational Research*, 10(2):205–209.

Saidi, S. A., Holland, C. M., Kreil, D. P., MacKay, D. J., Charnock-Jones, D. S., Smith, S. K., et al. (2004). Independent component analysis of microarray data in the study of endometrial cancer. *Oncogene*, 23(39):6677.

Schölkopf, B., Smola, A., and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel Methods for Pattern Analysis*. Cambridge university press.

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*, volume 26. CRC press.

Sneath, P. H., Sokal, R. R., et al. (1973). *Numerical Taxonomy. The Principles and Practice of Numerical Classification.*

Steinbach, M., Ertöz, L., and Kumar, V. (2004). The challenges of clustering higdimensional data. In *New Directions in Statistical Physics*, pages 273–309. Springer.

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.

Tasoulis, S., Plagianakos, V., and Tasoulis, D. (2011). Independent component divisive clustering of gene expression data. In *Proceedings of Eighth International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics, Gargnano Lago di Garda, Italy*, pages 1–9.

Tasoulis, S. K., Tasoulis, D. K., and Plagianakos, V. P. (2010). Enhancing principal direction divisive clustering. *Pattern Recognition*, 43(10):3391–3411.

Tasoulis, S. K., Tasoulis, D. K., and Plagianakos, V. P. (2012). Random direction divisive clustering. *Pattern Recognition Letters*, 34(2):131–139.

Tenenbaum, J. B., De Silva, V., and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323.

Theodoridis, S. and Koutroumbas, K. (2008). *Pattern Recognition*. Academic Press, 4th edition.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the Gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423.

Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622.

Topchy, A., Jain, A. K., and Punch, W. (2005). Clustering ensembles: models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1866–1881.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Wagner, D. and Wagner, F. (1993). Between min cut and graph bisection. In *International Symposium on Mathematical Foundations of Computer Science*, pages 744–750. Springer.

Walther, G. (1997). Granulometric smoothing. *The Annals of Statistics*, 25(6):2273–2299.

Xu, L., Neufeld, J., Larson, B., and Schuurmans, D. (2004). Maximum margin clustering. In *Advances in Neural Information Processing Systems*, pages 1537–1544.

Zahn, C. T. (1971). Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 100(1):68–86.

Zelnik-Manor, L. and Perona, P. (2004). Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1601–1608.

Zentner, A. R., Berlind, A. A., Bullock, J. S., Kravtsov, A. V., and Wechsler, R. H. (2005). The physics of galaxy clustering. A model for subhalo populations. *The Astrophysical Journal*, 624(2):505.

Zhang, R. and Rudnicky, A. I. (2002). A large scale clustering scheme for kernel k-means. In *Proceedings. 16th International Conference on Pattern Recognition*, volume 4, pages 289–292. IEEE.

Zhang, T., Ramakrishnan, R., and Livny, M. (1996). BIRCH: an efficient data clustering method for very large databases. In *ACM Sigmod Record*, volume 25, pages 103–114. ACM.

Zhang, Y. J. (1996). A survey on evaluation methods for image segmentation. *Pattern Recognition*, 29(8):1335–1346.

Zhao, Y. and Karypis, G. (2005). Data clustering in life sciences. *Molecular Biotechnology*, 31(1):55–80.

Zhuang, X., Huang, Y., Palaniappan, K., and Zhao, Y. (1996). Gaussian mixture density modeling, decomposition, and applications. *IEEE Transactions on Image Processing*, 5(9):1293–1302.