Chapter 3

Minimum Density Hyperplanes

3.1 Introduction

There is an abundance of density-based clustering methods available to analysts. This chapter focuses on a recently developed technique, known as Minimum Density Hyperplane clustering, created by Pavlidis *et al.* (2016). This approach is based on the problem of learning an optimal low-density linear separator, as proposed by Ben-David *et al.* (2009). This method adopts the density-based clustering definition given by (Hartigan, 1975), whereby a *high-density* cluster is defined as a connected region surrounding a mode of a probability density function. Thus, any points that fall within the same connected high density region are considered to belong to one cluster. A direct consequence of this definition is that clusters are separated by regions of low-density, in accordance with the *low-density separation assumption* (Ben-David *et al.*, 2009).

Minimum Density Hyperplane clustering directly identifies low-density separators by learning the equation of a hyperplane which will partition the modes associated with high-density regions, as well as possible using projection pursuit. Projection pursuit is a class of statistical techniques which seeks an optimal linear transformation, from all possible transformations, that identifies an *interesting* low-dimensional projection of a dataset (Huber, 1985). MDH initialises projection pursuit using a well known method, Principal Component Analysis (PCA). The computation for PCA reduces to the eigen-decomposition problem for a positive semi-definite matrix (*e.g.* covariance matrix) whereby the first principal component is defined along the direction explaining the most variation within the data (Jolliffe, 2011).

The MDH objective seeks an optimal univariate projection upon which a linear decision boundary separates at least one dense region from all others (Hofmeyr and Pavlidis, 2018). Principal components have been used in an attempt to solve this problem (Tasoulis *et al.*, 2010). MDH seeks to improve upon this solution by utilising projection pursuit to further enhance the

separation of at least one dense region from others. A key advantage of using univariate projections is that it renders the density separation problem along the projection almost trivial. Projection pursuit can thus be performed efficiently, making this approach applicable to much larger datasets than is common for most density-based methods (Pavlidis *et al.*, 2016).

The probability density function of a univariate projection within MDH is estimated using a Gaussian kernel density estimator (Pavlidis *et al.*, 2016). A benefit from using Gaussian kernels in the full dimensional space is that it allows one to evaluate the density on a hyperplane exactly using univariate Gaussian kernels. Also, one can reduce time complexity by applying MDH to a subsample of the dataset. The decision rule obtained from the sample of the data can then be applied to the full dataset to obtain a complete clustering solution.

Central to MDH is the choice of a kernel bandwidth to use when estimating the density of the univariate projections. As with most density-based clustering methods, the choice of kernel bandwidth influences the solution and is not trivial. Larger bandwidth values can cause modes to merge, increasing the possibility of not locating an optimal low-density separator. Alternatively, smaller values generate additional *shallow* modes, increasing the possibility of locating a solution plane that does not effectively separate clusters. Whether h is set relatively large or small, the probability density estimate will always exhibit low densities along the boundaries. To mitigate against the possibility of defining a low-density separating hyperplane that is located near the edge of the density function, MDH applies a penalty term during the optimisation process which restricts the hyperplane solution's distance from the mean.

The main limitation of MDH is that it defines a cluster based on a linear hyperplane. While some real-world applications involve datasets which are well separated by linear hyperplanes, there are instances in which clusters cannot be separated linearly (Hofmeyr and Pavlidis, 2018). Yates and Pavlidis (2016) proposed a method to remove the limitation of a linear hyperplane by embedding the data, non-linearly, into a high-dimensional feature space using Kernel Principal Component Analysis. MDH is then applied to the embedded data, where the low-density separator in the feature space corresponds to a non-linear hyperplane in the input space. We propose an alternative approach to remove the limitation imposed by a linear separator. We suggest collecting observations in neighbourhood around a hyperplane solution and then reassigning these observations with a more flexible clustering method. It is thought that by removing the limitations of a linear separator, we can improve hyperplane solutions when the data are non-linearly separable. This is the topic of Chapter 4, wherein two techniques are presented to perform such a task, Mean Shift and a single step gradient heuristic.

When a dataset is thought to consist of more than two clusters, then a single binary partition is not ideal. To overcome this, one can combine several hyperplane solutions in a hierarchical way. This approach is embodied in a

method known as Minimum Density Divisive Clustering (MDDC) (Hofmeyr and Pavlidis, 2018). While the focus of this paper is on the refinement of a single hyperplane solution, these enhancements can also be applied to each hyperplane during the divisive partitioning of MDDC. Details of MDDC are beyond the scope of this paper.

The remainder of this chapter is organized as follows. Formulation and notation are established in Section 2 before applying MDH to the four distinct cluster type datasets (Figure 2.2) in Section 3. The effects of various MDH parameter settings are explored within Section 4 before summarising the chapter in Section 5.

3.2 Formulation

The formulation of MDH (Hofmeyr and Pavlidis, 2018; Pavlidis *et al.*, 2016) sets out to define a hyperplane which bi-partitions a finite dataset, $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^d$, that is assumed to be independent and identically distributed with an unknown probability density function $p(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^+$. A hyperplane is defined as $\mathrm{H}(\boldsymbol{v}, b) := \{\mathbf{x} \in \mathbb{R}^d | \boldsymbol{v}^\top \mathbf{x} = b\}$. The hyperplane $\mathrm{H}(\boldsymbol{v}, b)$ partitions $\mathbf{X}$ into two clusters to which each element is assigned according to the following rule:

$$\mathbf{X} = \mathbf{X}^+_{\boldsymbol{v},b} \cup \mathbf{X}^-_{\boldsymbol{v},b}, \tag{3.2.1}$$

$$\mathbf{X}^-_{\boldsymbol{v},b} := \Big\{\mathbf{x} \in \mathbf{X} \big| \boldsymbol{v}^\top \mathbf{x} < b\Big\}, \tag{3.2.2}$$

$$\mathbf{X}^+_{\boldsymbol{v},b} := \Big\{\mathbf{x} \in \mathbf{X} \big| \boldsymbol{v}^\top \mathbf{x} \geq b\Big\}, \tag{3.2.3}$$

where the decision boundary between clusters is defined by the linear equation $\boldsymbol{v}^\top \mathbf{x} = b$. Furthermore, the *projection vector* ($\boldsymbol{v}$) defining the hyperplane, is restricted to have unit norm, where $\boldsymbol{v}^\top \mathbf{X} = \{\boldsymbol{v}^\top \mathbf{x}_i\}_{i=1}^n$ denotes the projection of $\mathbf{X}$ onto $\boldsymbol{v}$.

The *density on the hyperplane* is defined as the integral of $p(\mathbf{x})$:

$$I(\boldsymbol{v}, b) := \int_{H(\boldsymbol{v},b)} p(\mathbf{x}) dx, \tag{3.2.4}$$

where $p(\mathbf{x})$ is approximated by an isotropic Gaussian kernel density estimator:

$$\hat{p}(\mathbf{x}\big|\mathbf{X}, h^2\mathbf{I}) = \frac{1}{n(2\pi h^2)^{\frac{d}{2}}} \sum_{i=1}^n \exp\left\{ - \frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2h^2} \right\}. \tag{3.2.5}$$

One of the benefits of estimating $p(\mathbf{x})$ using isotropic Gaussian kernels is that it allows one to evaluate the density on the hyperplane exactly from one-dimensional projections (Pavlidis *et al.*, 2016). The density on the hyperplane

is evaluated by:

$$\hat{I}(\boldsymbol{v}, b|\mathbf{X}, h^2\mathbf{I}) := \int_{H(\boldsymbol{v},b)} \hat{p}(x|\mathbf{X}, h^2\mathbf{I})dx, \tag{3.2.6}$$

$$= \frac{1}{nh\sqrt{2\pi}} \sum_{i=1}^{n} \exp\left\{ - \frac{(b - \boldsymbol{v}^\top \mathbf{x}_i)^2}{2h^2} \right\}, \tag{3.2.7}$$

$$= \hat{p}\Big(b|\{\boldsymbol{v}^\top \mathbf{x}_i\}_{i=1}^n, h^2\mathbf{I}\Big). \tag{3.2.8}$$

To elaborate, consider that the density on $\mathrm{H}(\boldsymbol{v}, b)$ is estimated using Gaussian kernels with a given bandwidth (h) then the density on a hyperplane is given by $\hat{I}(\boldsymbol{v}, b)$. The density on a hyperplane can be evaluated by; projecting $\mathbf{X}$ onto $\boldsymbol{v}$, estimating the kernel density using the same bandwidth (h) and evaluating the hyperplane at b.

It is inevitable that the surface integral, $\hat{I}(\boldsymbol{v}, b)$, of $\hat{p}(\mathbf{x})$ on $\mathrm{H}(\boldsymbol{v}, b)$ will approach zero given a relatively large absolute value for b, resulting in a solution that partitions all but a few observations to one cluster (Pavlidis *et al.*, 2016). That is to say, for any $\boldsymbol{v}$, $\lim_{b\to\pm\infty} \hat{I}(\boldsymbol{v}, b) = 0$. To guard against obtaining a solution hyperplane near the boundaries of the data, a penalty term is introduced to $\hat{I}(\boldsymbol{v}, b)$ which constrains the hyperplane's distance from the mean of the data. Given $\Phi\big(\boldsymbol{v}|\mathbf{X}\big)$ is the *projection index*, the optimisation is defined as:

$$\min_{\boldsymbol{v}} \Phi\big(\boldsymbol{v}|\mathbf{X}\big) = \min_{b\in\mathbb{R}} \Big\{ \phi(\boldsymbol{v}, b|\mathbf{X}) \Big\}, \tag{3.2.9}$$

$$\phi(\boldsymbol{v}, b|\mathbf{X}) = \hat{I}(\boldsymbol{v}, b) + C \max\big\{0, -\alpha\sigma_{\boldsymbol{v}} - b, b - \alpha\sigma_{\boldsymbol{v}}\big\}^{1+\epsilon}, \tag{3.2.10}$$

for any constant value C and $\epsilon \in (0, 1)$, where $\sigma_{\boldsymbol{v}}$ represents the standard deviation of $\boldsymbol{v}^\top\mathbf{X}$ and α is a factor which manipulates the overall size of the feasible region. As a final step, one can further restrict the location of the hyperplane solution, post-optimisation, by requiring it to reside between *prominent* modes. This effectively diminishes the possibility of obtaining a poor hyperplane solution due to a relatively large α value. Note that, since this constraint is done after the optimisation process, the differentiability properties of MDH are maintained. The resulting point on $\boldsymbol{v}$, b_w is defined as:

$$b_w = \arg\min_{b\in\mathbb{M}^w} \Big\{ \hat{I}(\boldsymbol{v}, b|\mathbf{X}, h^2\mathbf{I}) \Big\}, \tag{3.2.11}$$

where $\mathbb{M}^w$ is an interval defined by w *prominent* modes on $\boldsymbol{v}$. The value w represents the number of largest modes to consider as *prominent*. When $w = 2$, only the two largest modes are considered, with their location on $\boldsymbol{v}$ defining the interval within which b_w is defined. If $w = 4$, then the location of the four largest modes define the interval boundary for b_w, with the lowest associated location on $\boldsymbol{v}$ setting the lower bound and the highest associated modal location setting the upper bound. If w is greater than the number of modes identified within the estimated density, then w is truncated to the observed number of modes. If the estimated density is uni-modal then w is truncated to one and

the interval will consist of only one point on $\boldsymbol{v}$ representing the location of the mode, in which case we consider b_w to be undefined. When b_w is undefined over all projected density estimates, the solution reverts to the hyperplane solution along the initial projection vector. In situations when b_w is undefined for some of the projections, then the solution reverts to the last known defined b_w solution. Setting w excessively high, will result in considering lower valued probability modes and possibly lead to a solution that clusters all but a few observations into a single cluster. Figure 3.1 illustrates how the choice of w can affect the final location of the hyperplane with respect to relatively large α values.

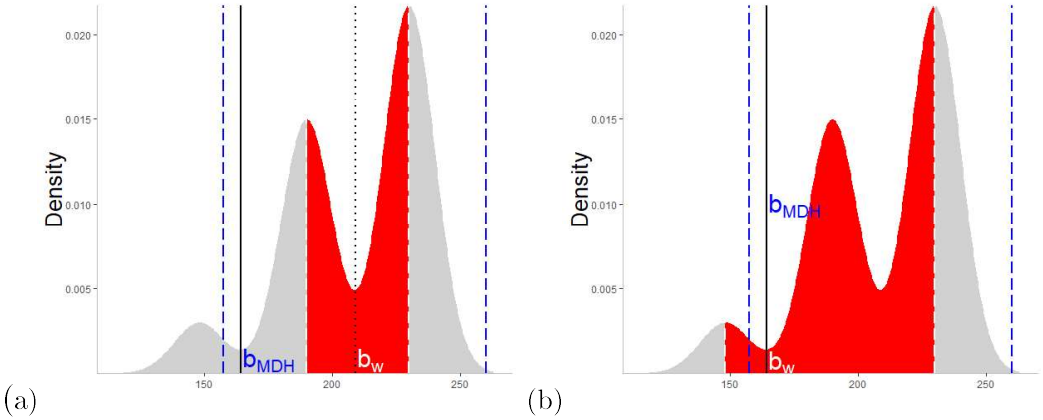

Figure 3.1: Example when setting $\alpha = 2$ and $w = 2$ with hyperplane solution b_w (a) and the setting which $w = 3$ with hyperplane solution $b_w = b_{MDH}$ (b). The blue lines represent the MDH maximum feasible region and the red shaded area represents the $\mathbb{M}^w$ interval.

Figure 3.1(a) illustrates the setting in which α is set relatively high (MDH feasible region indicated by blue lines). In this setting, MDH defines the optimal hyperplane at b_{MDH} (solid black line), clustering all but a relatively small number of objects to one group. However, implementing an additional constraint that restricts the final solution to a location between the two largest modes ($w = 2$) results in a final solution hyperplane at b_w. Increasing w to three results in an interval which contains the standard MDH's separator (Figure 3.1(b)) and thus both methods return the same hyperplane solution. The location of the hyperplane associated with $w = 2$ is arguably better, in the sense that the $\mathbb{M}^{w=2}$ interval contains the location of a hyperplane that results in a more balanced clustering solution. This simple example illustrates the motivation for applying an additional constraint upon the MDH solution. Additionally, bandwidth size should also be considered in concert with the selection of w since h influences the number of modes that an estimated density

will contain. The joint effect of w and bandwidth size are illustrated later on within Section 3.3.2. With the formulation of MDH defined, attention now turns to applying MDH within R.

3.3 Application in R

MDH clustering as defined within this text is applied using an augmented version of the `mdh` function from the `PPCI` R-package (Hofmeyr, 2018). The only required input to apply MDH within R is the dataset being clustered. There are several optional inputs a user can define when applying the standard `mdh` function from the `PPCI` package: the scale of the penalty term (α), an initial projection direction (defaulted to the first principal component) and a bandwidth to estimate the probability density functions. The original `mdh` function is adjusted to account for the constraint, dictated in Equation 3.2.11, which allows for the additional input for a number of *prominent* modes to consider when constructing the $\mathbb{M}^w$ interval. For the purpose of this text, focus is restricted to the choice of α, bandwidth and w parameters. Interested readers are directed to Hofmeyr (2018) for further details regarding the various inputs that can be specified within the `mdh` function.

MDH is applied to each of the data structure types displayed in Figure 2.2. Figure 3.2 illustrates how MDH assigned clusters for each of these distinct data structure types. As expected, MDH performs well when segmenting the linearly separable dataset (Figure 3.2(a), *Type A*) and does not correctly cluster the remaining non-linearly separable data types (Figure 3.2(b-c), *Type B-C*). However, projection pursuit enables a solution for *Type B* which is reasonable, resulting in relatively few errors compared to the other non-linearly separable datasets.

From each of the MDH solutions the projection vector ($\boldsymbol{v}$), allowable distance from mean (α), hyperplane location (b) and bandwidth (h) can be extracted. Transforming the data via $\boldsymbol{v}^\top \mathbf{X}$, estimating the density using h and plotting the hyperplane at b allows for a visual inspection of the final MDH solution (Figure 3.3). *Type A* and *Type B* datasets exhibit bi-modal distributions with a hyperplane solution that correctly identified the minimum integrated density within the feasible region. *Type C* and *Type D* solutions also locate the minimum density within the feasible region. The estimated density for *Type C* contains more than two modes and as such it is not possible to locate a point on $\boldsymbol{v}$ that will successfully cluster each object to its true class. For MDH to accurately cluster densities which are multimodal, each class would have to be positioned entirely to one side of the hyperplane location on $\boldsymbol{v}$. MDH was unable to identify more than one mode for the *Type D* dataset. Ultimately there is no location on $\boldsymbol{v}$ that will result in successfully grouping all points for the *Type D* dataset.

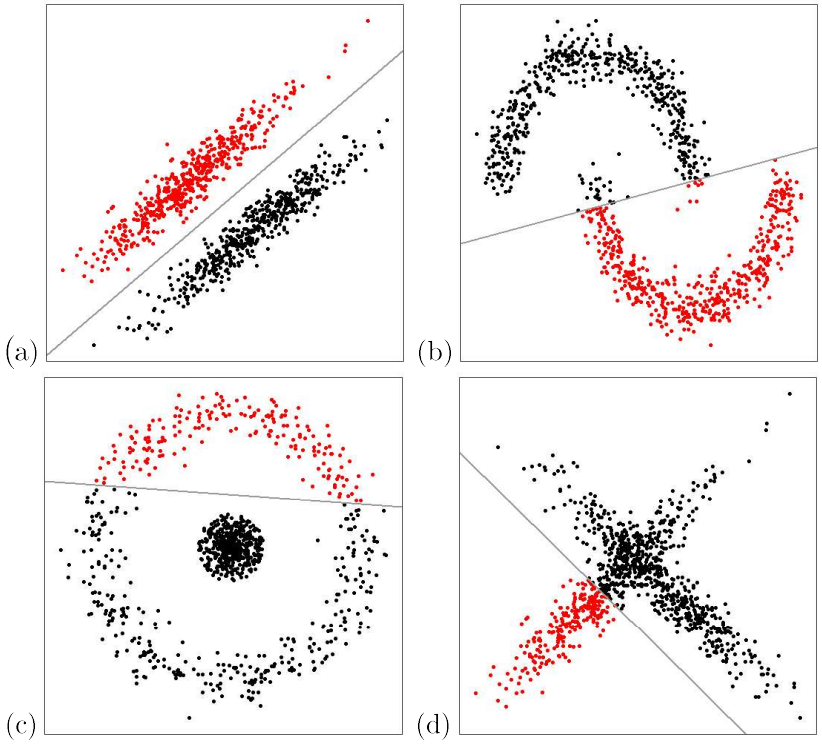

Figure 3.2: Minimum Density Hyperplane clustering of distinct (a-c) and overlapping (d) group structures.

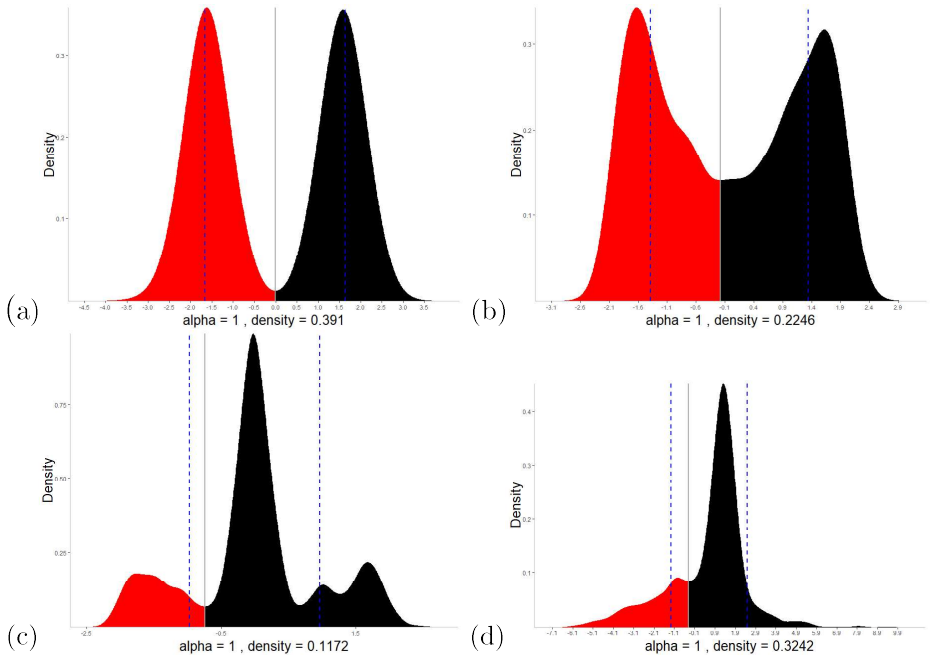

Figure 3.3: Univariate density estimates used for final results from clustering distinct (a-c) and overlapping (d) cluster structures.

3.3.1 Effect of bandwidth

As with all density-based methods, the choice of kernel bandwidth is crucial. Silverman (1986) recommends a bandwidth selection rule, $h = 0.9\hat{n}^{-1/5}\hat{\sigma}$, where 0.9 is a heuristic for univariate data, assumed to be multimodal and $\hat{\sigma}$ is the standard deviation of the data. MDH adopts this bandwidth selection rule but replaces the estimated standard deviation of a dataset with the square root of the variance explained within the first principal component. The bandwidth selection rule is defined as:

$$h = 0.9\hat{n}^{-1/5}\hat{\sigma}_{pc_1}, \tag{3.3.1}$$

where $\hat{\sigma}_{pc_1}$ is the estimated standard deviation of the data projected on the first principal component (Pavlidis *et al.*, 2016). This bandwidth will be referred to as the *heuristic* kernel bandwidth in this text. The multivariate version of the heuristic is defined as:

$$h^* = \left\{\frac{4}{2+d}\right\}^{-1/(4+d)} n^{-1/(4+d)}\hat{\sigma}_{pc_d}, \tag{3.3.2}$$

where d represents the number of dimensions and $\hat{\sigma}_{pc_d}$ is the average estimated standard deviation of the data projected onto d principal components. The h^* bandwidth will be referred to as the *full* kernel bandwidth for the remainder of this text. Bear in mind that given the definition in Equation 3.3.2, it cannot be said that the heuristic applies greater kernel smoothing compared to the full bandwidth. The relative difference is dependent on the dataset used for clustering and in some instances, one will apply relatively more smoothing than the other. Unless otherwise stated, the heuristic bandwidth is utilised when clustering a dataset. Applying MDH to *Type A* and *Type C* data using various bandwidths illustrates the influence that the bandwidth has on the final solution (Figure 3.4, 3.5 respectively).

Applying smaller bandwidth values when clustering *Type A* further increases the disparity between each of the classes' associated densities. Increasing the bandwidth results in merging both modes and a solution which poorly clusters the observations. While MDH should easily find the optimal separating plane for data which are linearly separable, choosing a bandwidth that is relatively large will produce a poor solution. For data which are non-linearly separable, as with *Type C*, the choice of bandwidth has little impact on the final solution. No level of smoothing can achieve a density upon which a linear separator will successfully assign class labels of this data type.

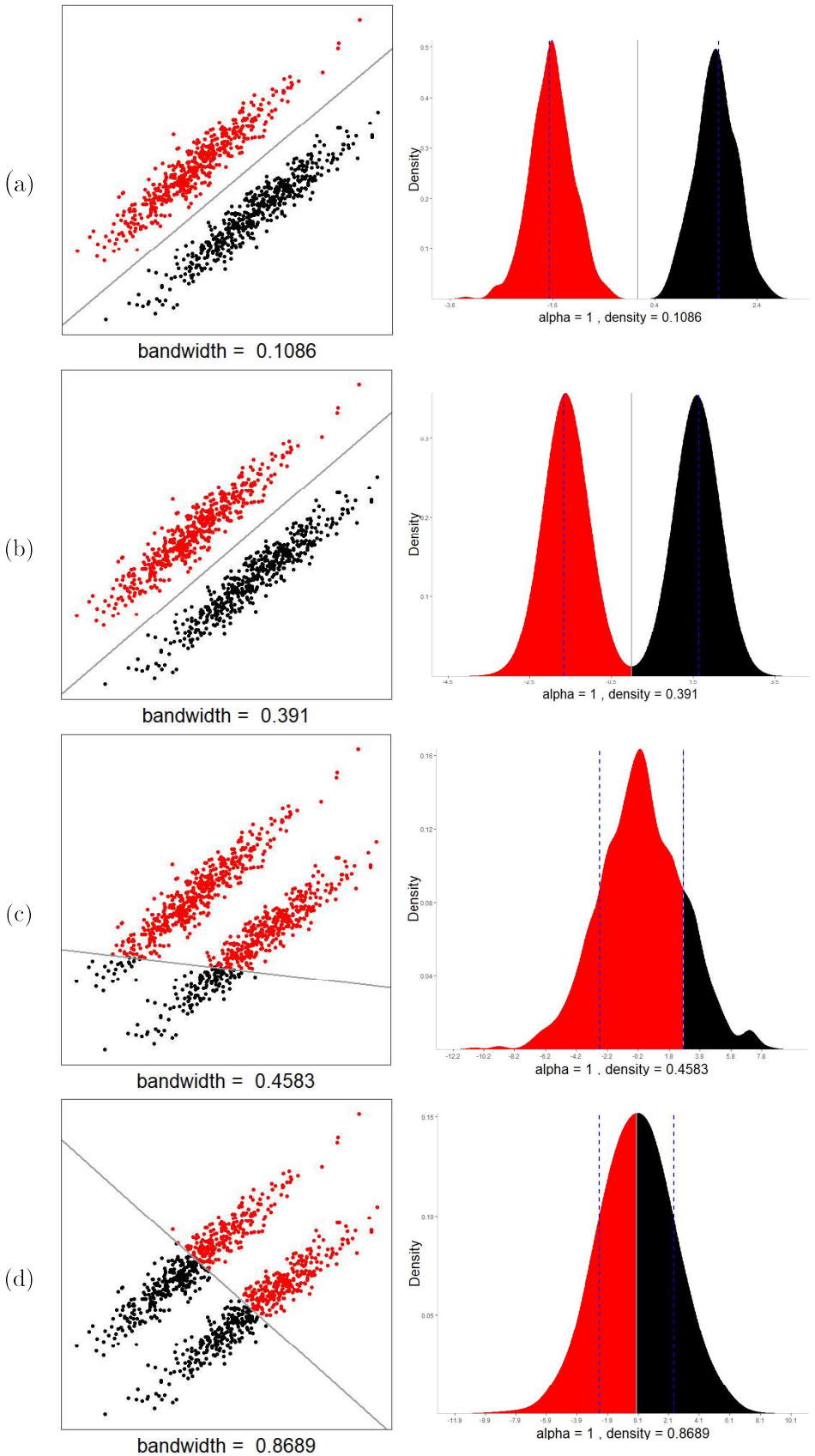

Figure 3.4: Effect of kernel bandwidth on MDH solution for distinct cluster *Type A*, using: a relatively small bandwidth(a), heuristic bandwidth (b), full bandwidth (c) and large bandwidth (d).

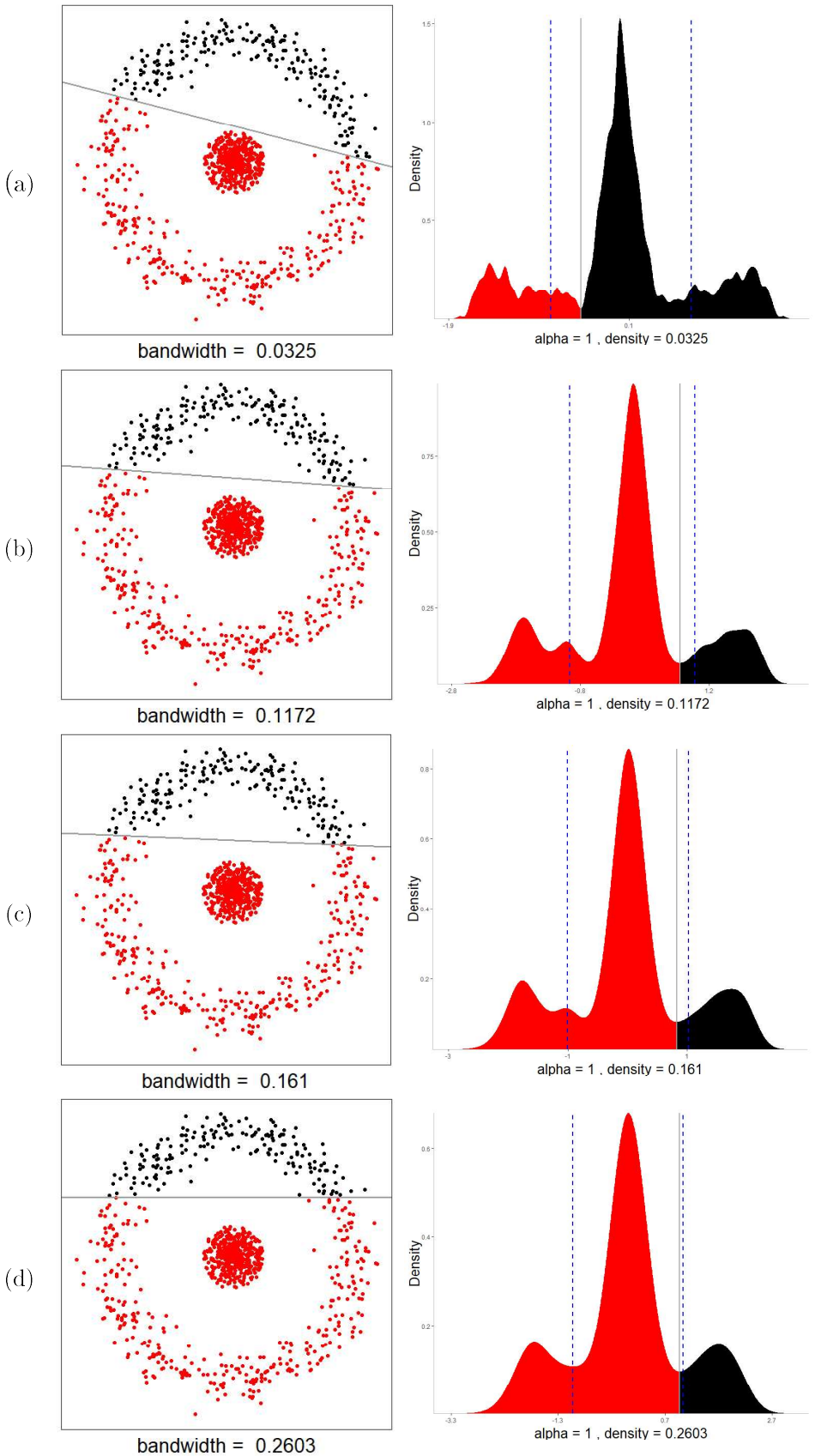

Figure 3.5: Effect of kernel bandwidth on MDH solution for distinct cluster *Type B*, using: a relatively small bandwidth(a), heuristic bandwidth (b), full bandwidth (c) and large bandwidth (d).

3.3.2 Constraints on hyperplane

As previously mentioned, the MDH solution plane is restricted to be within a given distance from the mean of the data. This reduces the possibility of assigning all but a few observations into one cluster. While the penalty does assist with reducing the possibility of returning this solution, it also serves another meaningful purpose. At each incremental α value, projection pursuit seeks an optimal transformation to obtain a low-density linear separating plane. Setting a larger range for α, increases the overall search region and iterations for projection pursuit. This increases the possibility of locating an optimal separating hyperplane.

The partial path to the final MDH solution for the *Type B* data is illustrated in Figure 3.6. At the first iteration, MDH utilises the first principal component as a projection vector and estimates the associated density function. Each subsequent projection's density is estimated by projecting the data onto the axis with maximum variance orthogonal to the projection vector. The points within each plot represent the projection of the data, where the y-axis represents the maximum variability orthogonal to the projection vector, associated with the x-axis (Pavlidis *et al.*, 2016). The red line within each plot indicates $\mathrm{H}(\boldsymbol{v}, b)$ and the constrained optimisation region is indicated as black dotted lines. As α incrementally increases, projection pursuit rotates the data. Iteration 16 represents the final, optimal hyperplane solution.

The solution associated with iteration 27 is rejected since it produces a hyperplane which passes through a point which is not a minimiser. This solution would have grouped all of the data into one cluster, as evident from the location of b relative to the scatter plot of all transformed data points. Based on the formulation of MDH, iteration 27's solution was rejected and MDH reverts back to the last known acceptable solution.

Previously, it was stated that w should be chosen in concert with the kernel bandwidth. Consider again the linearly separable *Type A* dataset. With a relatively large bandwidth and $w = 2$ (indicating that b must lay between the interval defined by the two largest modes), the result clusters all but a few objects to one cluster (Figure 3.7). While it is reasonable to expect a quality separating hyperplane resides between the two largest modes, this reasoning was diminished due to a poor choice of bandwidth. If the added constraint on the feasible region was not implemented, the hyperplane location on $\boldsymbol{v}$ would have been placed near -8, which would have resulted in one cluster containing all but a few observations.

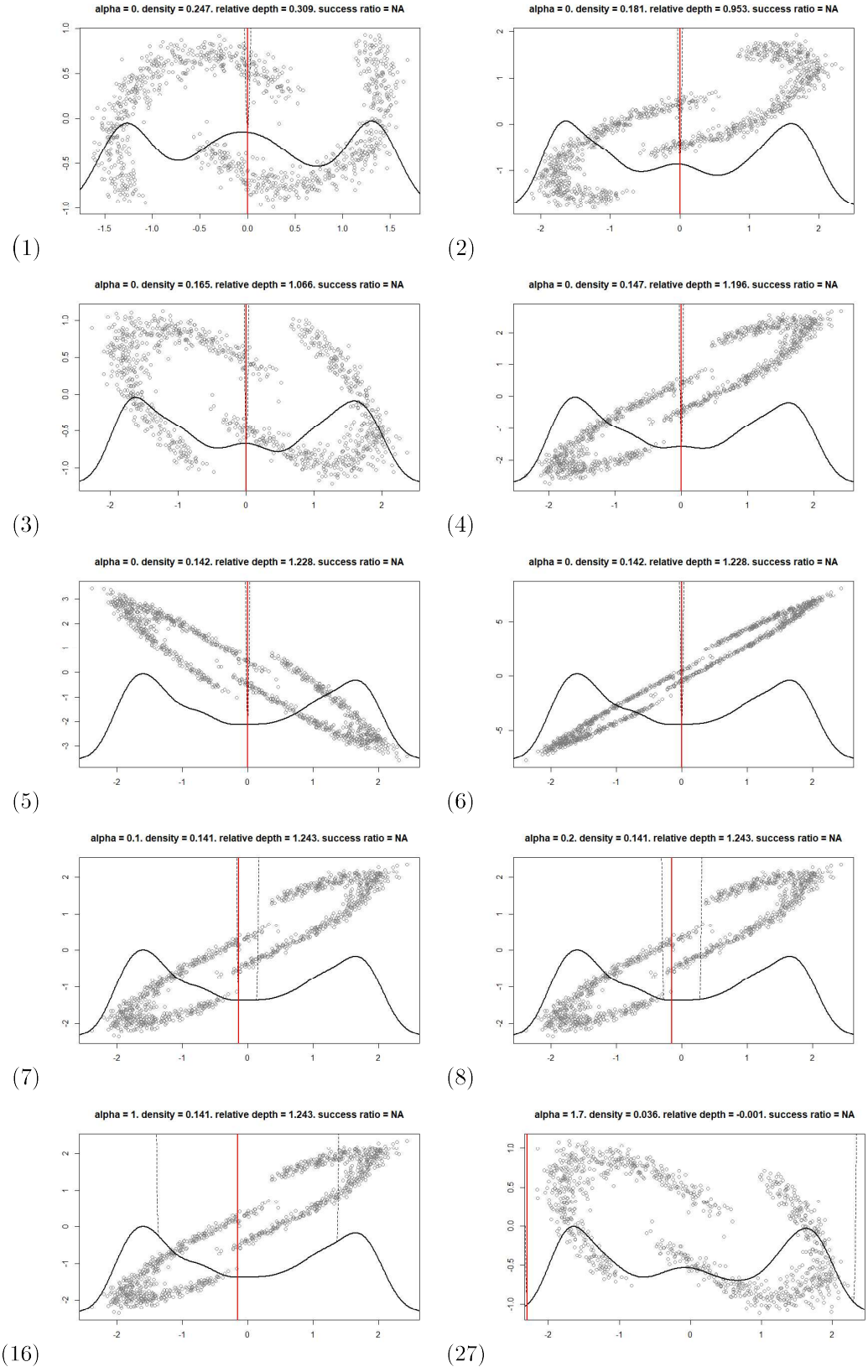

Figure 3.6: Illustration of Minimum Density Hyperplane estimation through different iterations utilising incremental α values. Each figure is accompanied by a number representing the overall iteration within the MDH solution.

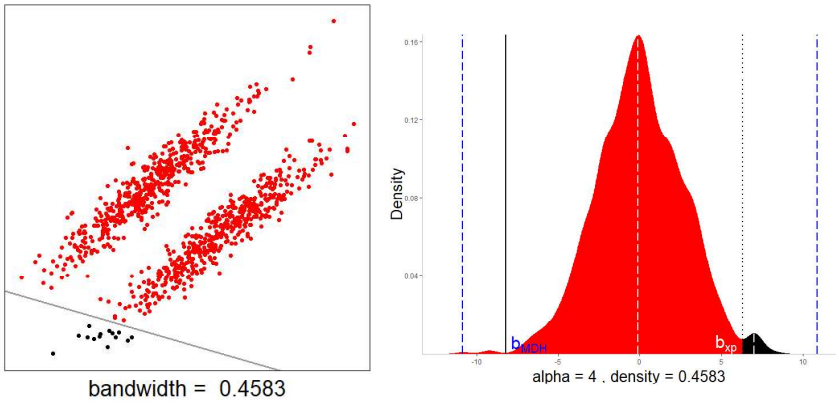

Figure 3.7: MDH solution of linearly separable data using a relatively large bandwidth and setting $w = 2$. The blue dashed lines represent the maximum feasible region, scaled be α and the white dashed lines represent the $\mathbb{M}^{w=2}$ interval

3.4 Summary

Minimum Density Hyperplane clustering was presented and details regarding how the algorithm solves the clustering problem were discussed. MDH seeks the minimum integrated density of an estimated probability distribution function. One of the challenges involved with locating a low-density hyperplane is to avoid solutions which groups all but a few observations to one cluster. Restricting the hyperplane distance to the mean of the data greatly reduces the likelihood of obtaining such a solution. An additional constraint was presented, whereby the final position of the hyperplane solution must reside within an interval bounded by the smallest and largest of the w *prominent* mode locations on $\boldsymbol{v}$. With this newly presented constraint, α can be set high, increasing the overall search area and projection pursuit iterations. This increases the possibility of locating an optimal separating hyperplane without the consequence of an insignificant solution. While this additional constraint guards against a solution which places all but a few objects into one cluster, it is still susceptible to the choice of bandwidth size.

As with all clustering methods the choice of bandwidth is not trivial. Clustering the linearly separable data, *Type A* with large h values produced poor solutions while relatively smaller values resulted in correctly identifying the underlying class structure. Regardless of the size of the kernel smoother, when data are non-linearly separable MDH cannot learn the true underlying cluster structure. Herein lies a limitation of an MDH solution, the linear decision boundary. Since MDH learns a linear decision boundary, it is incapable of segmenting complex cluster structures which are non-linearly separable. The following chapter details our approach to improve upon hyperplane solutions.