A generalized single linkage method for estimating the cluster tree of a density

Werner Stuetzle [*]
Department of Statistics
University of Washington

Rebecca Nugent [*]
Department of Statistics
Carnegie Mellon University

April 17, 2007

Abstract

The goal of clustering is to detect the presence of distinct groups in a data set and assign group labels to the observations. Nonparametric clustering is based on the premise that the observations may be regarded as a sample from some underlying density in feature space and that groups correspond to modes of this density. The goal then is to find the modes and assign each observation to the domain of attraction of a mode. The modal structure of a density is summarized by its cluster tree; modes of the density correspond to leaves of the cluster tree. Estimating the cluster tree is the primary goal of nonparametric cluster analysis. We adopt a plug-in approach to cluster tree estimation: estimate the cluster tree of the feature density by the cluster tree of a density estimate. For some density estimates the cluster tree can be computed exactly, for others we have to be content with an approximation. We present a graph-based method that can approximate the cluster tree of any density estimate. Density estimates tend to have spurious modes caused by sampling variability, leading to spurious branches in the graph cluster tree. We propose excess mass as a measure for the size of a branch, reflecting the height of the corresponding peak of the density above the surrounding valley floor as well as its spatial extent. Excess mass can be used as a guide for pruning the graph cluster tree. We point out mathematical and

[*] Supported by NSF grant DMS-9803226 and NSA grant 62-1942.

algorithmic connections to single linkage clustering and illustrate our approach on several examples.

Keywords: Cluster analysis, level set, single linkage clustering, excess mass, nearest neighbor density estimation, runt test.

1 Introduction

The goal of clustering is to identify distinct groups in a data set and assign a group label to each observation. We use the term "distinct groups" in the sense of Carmichael, George, and Julius (1968): contiguous, densely populated areas of feature space, separated by contiguous, relatively empty regions. This definition, while quite general, admittedly is not all-encompassing: Figures 1(a) - 1(c) would be regarded as showing two groups, while Figure 1(d) would not.

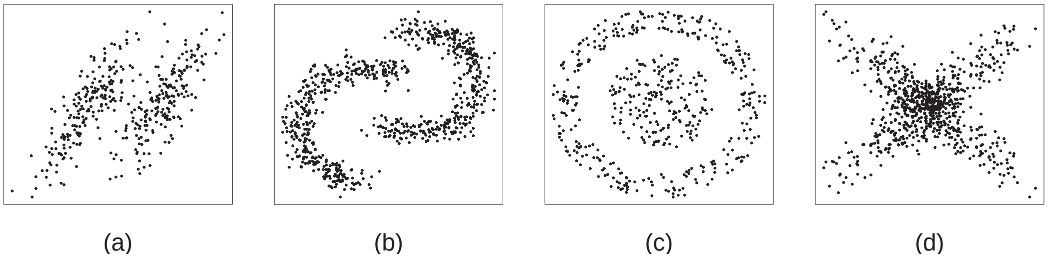


Figure 1: (a)-(c) Distinct groups in the sense of Carmichael, George, and Julius; (d) Groups that would not be considered distinct.

To cast clustering as a statistical problem we regard the data $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset R^m$ as a sample from some unknown probability density $p(\mathbf{x})$. There are two statistical approaches to clustering. The parametric approach (Fraley and Raftery 1998, 1999; McLachlan and Peel 2000) is based on the assumption that each group g is represented by a density $p_g(\mathbf{x})$ that is a member of some parametric family, such as the multivariate Gaussian distributions. The density $p(\mathbf{x})$ then is a mixture of the group densities, and the number of mixture components and their parameters are estimated from the data. Observations can be labeled using Bayes' rule.

In contrast, the nonparametric approach adopted in this paper is based on the premise that groups correspond to modes of the density $p(\mathbf{x})$. The goal then is to find the modes and assign each observation to the "domain of attraction" of a mode. Searching for modes as a manifestation of the presence of groups was first advocated in D. Wishart's (1969) paper on *Mode Analysis*. According to Wishart, clustering methods should be able to detect and "resolve distinct data modes, independently of their shape and variance".

Hartigan (1975, Section 11; 1981) expanded on Wishart's idea and made it more precise by introducing the notion of *high density clusters*. Define the level set $L(\lambda; p)$ of a density p at level λ as the subset of the feature space for which the density exceeds λ:

$$L(\lambda; p) = \{\mathbf{x} \mid p(\mathbf{x}) > \lambda\}.$$

Hartigan defined the high density clusters at level λ as the connected components of $L(\lambda; p)$.

Hartigan also pointed out that the collection of high density clusters has a hierarchical structure: for any two clusters A and B (possibly at different levels) either $A \subset B$ or $B \subset A$ or $A \cap B = \emptyset$. This hierarchical structure is summarized by the *cluster tree* of p. Each node N of the tree represents a subset $D(N)$ of the support $L(0; p)$ of p — a high density cluster of p — and is associated with a density level $\lambda(N)$. The cluster tree is easiest to define recursively (Stuetzle 2003). The root node represents the entire support of p and has associated density level $\lambda(N) = 0$. To determine the descendants of a node N we find the lowest level λ_d for which $L(\lambda; p) \cap D(N)$ has two or more connected components. If there is no such λ_d then p has only one mode in $D(N)$, and N is a leaf of the tree. Otherwise, let $C_1, \ldots, C_k$ be the connected components of $L(\lambda_d; p) \cap D(N)$. If $k = 2$ (the usual case) we create daughter nodes representing the connected components C_1 and C_2, both with associated level λ_d, and apply the definition recursively to the daughters. If $k > 2$ we create daughter nodes representing C_1 and $C_2 \cup \cdots \cup C_k$ and recurse. (Our terminology is different from the one used by Hartigan (1975); Hartigan refers to the connected components of all level sets as high density clusters, while we reserve this term for connected components of level sets associated with the nodes of the cluster tree.)

Figure 2 shows a density and the corresponding cluster tree. It is worth noting that the topology of the cluster tree of a density is invariant under nonsingular affine transformations of feature space; only the levels of the nodes change. In particular, the topology does not depend on the choice of units of measurement.

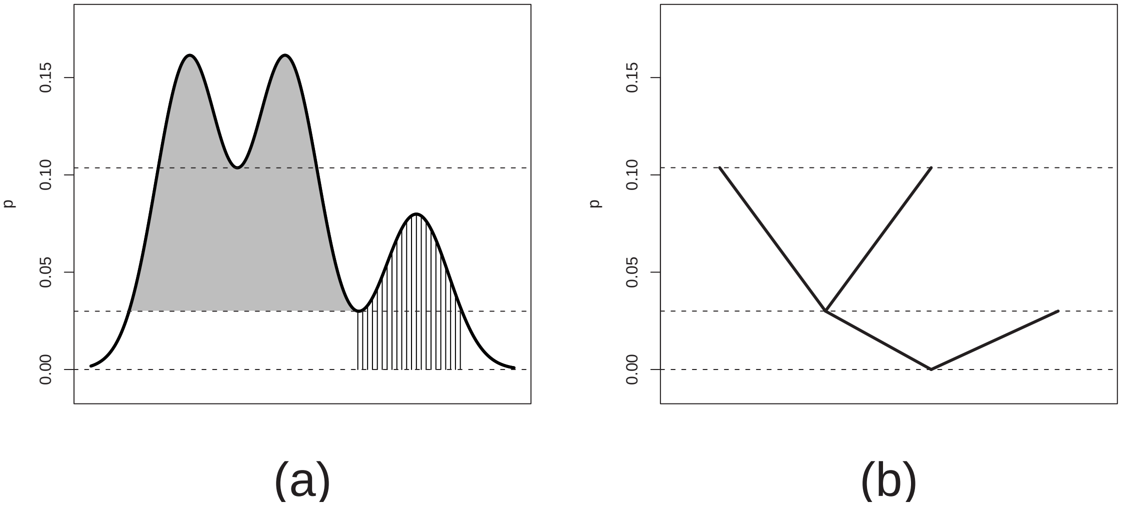


Figure 2: (a) Density; (b) cluster tree. The shaded area represents the excess mass of the left daughter of the root node. The hashed area represents the size of the right daughter of the root node.

We regard estimating the cluster tree as the fundamental goal of nonparametric cluster analysis. In this paper we propose a graph-based approach to cluster tree estimation which we call *generalized single linkage clustering.* In Section 2 we review previously suggested clustering methods that can be described in terms of level sets. In Section 3 we present the basic idea of our graph-based approach, and in Section 4 we illustrate our algorithm on a simple example. In Section 5 we describe a way of measuring the "prominence" of modes of a density, motivating a method for pruning branches of a cluster tree likely to correspond to spurious modes caused by sampling variability. In Section 6 we point out mathematical and algorithmic connections between our approach and single linkage clustering. In Section 7 we show additional examples. Section 8 with a summary and ideas for future work concludes the paper.

2 Previous work

Several previously suggested clustering methods can be described in terms of level sets and high density clusters.

Probably the earliest such method is Wishart's (1969) *one level mode analysis.* The goal of one level mode analysis is to find the connected components of $L(\lambda; p)$ for a given density level λ chosen by the user. The idea is to first compute a kernel density estimate $\hat{p}$ (Silverman 1986, Chapter 4) and set aside all observations with $\hat{p}(\mathbf{x}_i) \leq \lambda$, i.e., all observations not in the level set $L(\lambda; \hat{p})$. If the connected components of $L(\lambda; p)$ are well separated then the remaining high density observations should fall into clearly separated groups. Wishart suggests using single linkage clustering of the high density observations to identify the groups. One level mode analysis anticipates some of the "sharpening" ideas later put forth by P.A. Tukey and J.W. Tukey (1981).

A reincarnation of one level mode analysis is the DBSCAN algorithm of Ester, Kriegel, Sander, and Xu (1996). DBSCAN consists of four steps: (a) for each data point calculate a kernel density estimate using a spherical uniform kernel with radius r; (b) choose a density threshold λ and find the observations with $\hat{p}(\mathbf{x}_i) > \lambda$; (c) construct a graph connecting each high density observation to all other observations within distance r; (d) define the clusters to be the connected components of this graph. All observations not within distance r of a high density observation are considered "noise".

DBSCAN is closely related to Walther's (1997) method for estimating level sets and to the clustering algorithm of Cuevas, Febrero, and Fraiman (2000, 2001). Walther's level set estimator consists of two steps: (i) compute an estimate $\hat{p}$ of the underlying density p; (ii) approximate the level set of $\hat{p}$ by a union of balls B_r with suitably chosen radius r. In his theoretical development Walther uses a kernel density estimate for step (i), although operationally any density estimate could be substituted. The approximation $L^*(\lambda; \hat{p})$ of $L(\lambda; \hat{p})$ is constructed by first partitioning the data $\mathcal{X}$ (or a larger Bootstrap sample generated from $\hat{p}$) into subsets $\mathcal{X}^+ = \{\mathbf{x}_i \mid \hat{p}(\mathbf{x}_i) > \lambda\}$ and $\mathcal{X}^- = \{\mathbf{x}_i \mid \hat{p}(\mathbf{x}_i) \leq \lambda\}$, and then forming the union of those balls around points in $\mathcal{X}^+$ that do not contain any points in $\mathcal{X}^-$:

$$\hat{L}(\lambda; p) = L^*(\lambda; \hat{p}) = \left[(\mathcal{X}^- \oplus B_r) \cap \mathcal{X}^+\right]' \oplus B_r\,.$$

Here the $\oplus$ operator denotes dilation:

$$\mathcal{X} \oplus B_r = \{\mathbf{x} + \mathbf{y} \mid \mathbf{x} \in \mathcal{X}, \mathbf{y} \in B_r\}\,.$$

The representation of $\hat{L}(\lambda; p)$ as a union of balls is computationally very convenient; for example, it is easy to find the connected components. The crux of the matter, of course, is the choice of r. Walther gives a formula for r in terms of the behavior of p on the boundary $\partial L(\lambda; p)$ of the level set $L(\lambda; p)$: If

$$\begin{aligned} \|\nabla p(\mathbf{x})\| &> m \qquad \text{and} \\ \|\nabla p(\mathbf{x}) - \nabla p(\mathbf{y})\| &< k\,\|\mathbf{x} - \mathbf{y}\| \end{aligned}$$

for all $\mathbf{x}, \mathbf{y}$ on $\partial L(\lambda; p)$ then we should choose $r = m/k$. Of course, m and k for the true density p will be unknown in practice. However, the goal is to approximate $L(\lambda; \hat{p})$, which suggests substituting the corresponding quantities for $\hat{p}$. To our knowledge this approach has not been explored.

The level set estimator of Cuevas *et al.* (2000, 2001) differs from Walther's estimate in that $L^*(\lambda; \hat{p})$ is taken to be the union of balls B_r around all points in $\mathcal{X}^+$:

$$\hat{L}(\lambda; p) = L^*(\lambda; \hat{p}) = \mathcal{X}^+ \oplus B_r\,.$$

Walther claims that his estimator performs better asymptotically. It certainly makes more sense: an approximation to $L(\lambda; \hat{p})$ should not contain points for which $\hat{p}(\mathbf{x}) < \lambda$. Cuevas *et al.* propose several methods for choosing r, based solely on interpoint distances of points sampled from $\hat{p}$. They do not make use of the fact that an analytic expression for $\hat{p}$ is available and function values, derivatives, etc can be obtained, and they have a somewhat ad-hoc flavor.

A weakness of one level mode analysis or any method that attempts to find clusters based on a level set for a single level λ is apparent from Figure 2. The degree of separation between connected components of $L(\lambda; p)$, and therefore of $L(\lambda; \hat{p})$, depends critically on the choice of the cut level λ, which is left to the user. Moreover, there might not be a single value of λ that reveals all the modes.

Citing the difficulty in choosing a cut level, Wishart (1969) proposed *hierarchical mode analysis*, which can be regarded as a heuristic for computing the cluster tree of a kernel density estimate $\hat{p}$, although it appears that Wishart did not view it thus. (The word "tree" does not occur in the section of his paper on hierarchical mode analysis.) We use the term "heuristic" because there is no guarantee that hierarchical mode analysis will indeed correctly compute the cluster tree of $\hat{p}$ as defined above. Wishart's (1969) algorithm constructs the tree by iterative merging (i.e., is an agglomerative algorithm). It is quite complex, probably because its iterative approach is not well matched to the tree structure it is trying to generate.

The basic weakness of one level mode analysis was also noted by Ankerst, Breuning, Kriegel, and Sander (1999) who proposed OPTICS, an algorithm for "Ordering Points to Identify the Clustering Structure". OPTICS generates a data structure that allows one to calculate efficiently the result of DBSCAN for any desired density threshold λ. It also produces a graphical summary of the cluster structure. The idea behind their algorithm is hard to understand.

Stuetzle's (2003) *runt pruning* method estimates the cluster tree of the feature density by computing the cluster tree of the nearest neighbor density estimate and then pruning branches believed to correspond to spurious modes. In this paper we present a generalization of runt pruning to other density estimates.

Klemelae (2004, 2005) proposed tools for visualizing the level sets of density estimates that are piecewise constant over (hyper)-rectangles, such as histograms or discretized kernel estimates. Level sets and their connected components for such density estimates are easy to obtain. Klemelae defines a "level set tree" which is different from the cluster tree in that it has nodes at every one of the (finitely many) levels occuring as values of $\hat{p}$.

For sake of completeness we also mention an alternative method for estimating level sets based on the concept of "excess mass" put forth by Hartigan (1987). The excess mass of a set C at level λ is defined as

$$\begin{aligned} \mathcal{E}(\lambda, C; P) &= \int_C (p(\mathbf{x}) - \lambda)\, d\mathbf{x} \\ &= P(C) - \lambda \mu(C)\,. \end{aligned}$$

Here $P(C)$ and $\mu(C)$ denote probability content and Euclidean volume of C, respectively. It is easy to see that $L(\lambda; p)$ maximizes $\mathcal{E}(\lambda, C; P)$ among all Borel sets. This observation suggests estimating $L(\lambda; p)$ by maximizing empirical excess mass $\mathcal{E}(\lambda, C; P_n)$ over a collection $\mathcal{C}$ of sets (Mueller and Sawitzki 1991; Polonik 1995). The Borel sets are too large a class for the estimator to be consistent; $\mathcal{C}$ has to be a V-C class, or at least a Glivenko-Cantelli class for the true density. An example is the class of all ellipsoids. The need to specify a class of sets that are supposed to contain the true level set can be regarded as a strength or a weakness, depending on one's point of view. It allows one to incorporate prior knowledge about the shape of the level set into the estimation procedure, but on the other hand such information may not be available. Another problem with the approach is its computational complexity.

3 A graph-based approach to estimating the cluster tree of a density

An obvious way of estimating the cluster tree of a density p from a sample is to first compute a density estimate $\hat{p}$ and then use the cluster tree of $\hat{p}$ as an estimate for the cluster tree of p. This plug-in approach works for histograms or binned kernel density estimates which, however, are only viable in low dimensions (Nugent 2006). For many other estimates suitable for high-dimensional data, like Gaussian mixtures, kernel estimates, or projection pursuit estimates, computing level sets, and therefore computing the cluster tree, seems intractable. Instead we define and solve a closely related, but much simpler graph problem.

Let $\hat{p}_{ij}$ be the minimum value of the density estimate $\hat{p}$ over the line segment connecting observations $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$\hat{p}_{ij} = \min_{t \in [0,1]} \hat{p}((1-t)\,\mathbf{x}_i + t\,\mathbf{x}_j)\,.$$

Let G be the complete graph over the observations with edge weights $\hat{p}_{ij}$ and vertex weights $\hat{p}_{ii}$. Define the *threshold graph* $G(\lambda)$ as the subgraph of G consisting of the edges and vertices with $\hat{p}_{ij} > \lambda$. By construction, the vertices of $G(\lambda)$ are exactly the observations in $L(\lambda; \hat{p})$.

There is also a link between the connected components of $L(\lambda; \hat{p})$ and the connected components of the threshold graph $G(\lambda)$: Two observations in the same connected component of $G(\lambda)$ are guaranteed to lie in the same connected component of $L(\lambda; \hat{p})$ because they are connected by a path in G along which $\hat{p}_{ij} > \lambda$. Note that the reverse is not necessarily true: there might be a curve $\mathbf{x}(t) : [0,1] \to R^m$ with $\mathbf{x}(0) = \mathbf{x}_i$, $\mathbf{x}(1) = \mathbf{x}_j$ and $\hat{p}(\mathbf{x}(t)) \geq \lambda$ for all $t \in [0,1]$, even if there is no path in the graph G with this property. Therefore, observations in the same connected component of $L(\lambda; \hat{p})$ may lie in different connected components of $G(\lambda)$. However, erroneous splits are rare if $\hat{p}$ is smooth; we present some evidence for this assertion in Section 7. We will altogether miss connected components of $L(\lambda; \hat{p})$ that do not contain any observations, but those are probably artifacts of the density estimate and not of interest. In any case, our real target are the level sets $L(\lambda; p)$ of the feature density and their connected components; occasional mistakes in identifying connected components of $L(\lambda; \hat{p})$ are but one component of the overall estimation error.

The connected components of $G(\lambda)$ for different values of λ have a tree structure, just like the connected components of $L(\lambda; \hat{p})$. We call this tree the *graph cluster tree*; it is our approximation to the cluster tree of $\hat{p}$. Like the cluster tree of a density, the graph cluster tree is easiest to define recursively. Each node N of the graph cluster tree represents a subgraph $\tilde{D}(N)$ of G and is associated with a density level $\lambda(N)$. We refer to the vertex set of $\tilde{D}(N)$ as the *graph high density cluster* associated with N. The root node represents the entire graph G and has associated density level $\lambda(N) = 0$. To determine the descendants of a node N we find the lowest level λ_d for which $G(\lambda) \cap \tilde{D}(N)$ has two or more connected components. If there is no such λ_d then N is a leaf of the tree. Otherwise, let $C_1, \ldots, C_k$ be the connected components of $G(\lambda) \cap \tilde{D}(N)$. If $k = 2$ (the usual case) we create daughter nodes representing the connected components C_1 and C_2, both with associated level λ_d, and apply the definition recursively to the daughters. If $k > 2$ we create daughter nodes representing C_1 and $C_2 \cup \cdots \cup C_k$ and recurse.

Remark 1: In our graph-based approach the observations play two conceptually different roles. First, they are used to compute the density estimate $\hat{p}$. Second, they form the vertices of the graph G. The graph is merely a tool for approximating the structure of the level sets of $\hat{p}$. We could use a different set of "test" points as the graph vertices. For example, we could generate a large sample from $\hat{p}$, which would reduce the likelihood of erroneously splitting connected components of level sets of $\hat{p}$. We would end up with cluster labels for the test points and could then label the original observations using any classification method.

4 Computing the graph cluster tree

The recursive definition of the graph cluster tree given at the end of Section 3 translates directly into a recursive cluster tree algorithm for its computation. Note that when we we apply the splitting procedure to the subgraph $\tilde{D}(N)$ associated with a node N the only values for the threshold λ we have to consider are the weights of the edges in $\tilde{D}(N)$. However, we can simplify the algorithm and its visualization and expose similarities to other clustering methods (Section 6) by making use of a connection between the threshold graphs of the graph G and of its maximal spanning tree T:

Proposition 1: Let G be an edge weighted graph and T its maximal spanning tree. Then two vertices belong to the same connected component of $G(\lambda)$ iff they belong to the same connected component of $T(\lambda)$.

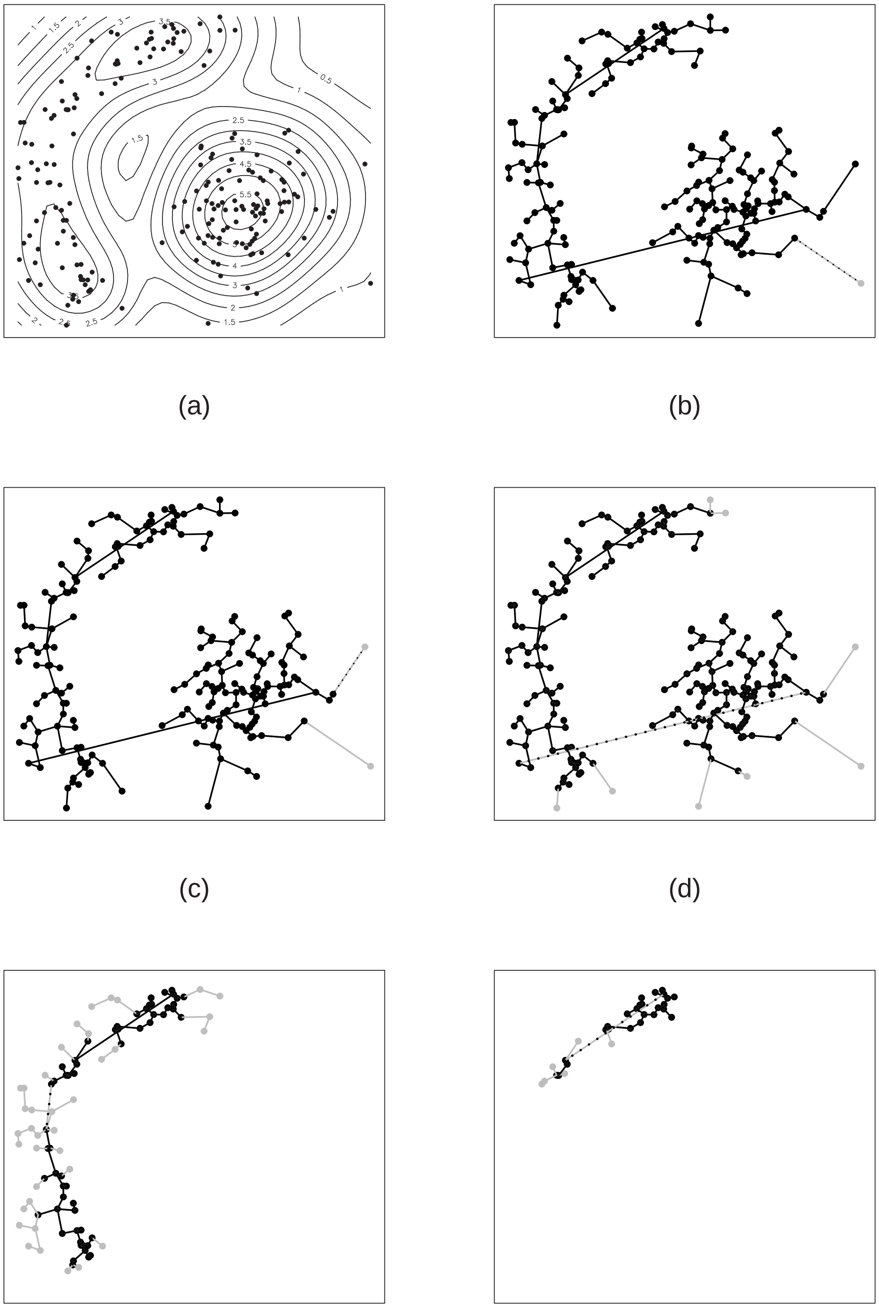
A proof of Proposition 1 can be found in the Appendix. Proposition 1 implies that we can apply the recursive cluster tree algorithm to the maximal spanning tree of G instead of G itself.

We still face an operational problem: the edge weights

$$\hat{p}_{ij} = \min_{t \in [0,1]} \hat{p}((1-t)\,\mathbf{x}_i + t\,\mathbf{x}_j)$$

of G for $j \neq i$ are not known explicitly but are solutions of an optimization problem. One way of dealing with this problem is to approximate the $\hat{p}_{ij}$ using a numerical optimizer, most simply a grid search. We used this method used to generate the examples presented in the paper.

We do have a more principled but computationally more demanding approach that can be shown to produce the correct tree. It is based on two observations: (i) to compute the maximal spanning tree of G and the graph cluster tree we only need the order of the edge weights of G; their exact values are not important; (ii) if the density estimate $\hat{p}$ is smooth (for example a kernel estimate with a smooth kernel or a Gaussian mixture estimate) then we can obtain upper and lower bounds on the $\hat{p}_{ij}$ using Taylor expansions. These bounds can be made arbitrarily tight at the cost of additional evaluations of $\hat{p}$ and its derivatives. This approach is described in Nugent (2006). In the examples we have tried, grid search and exact computation produce very similar results.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3: (a) Data set and isopleths of kernel density estimate; (b) maximal spanning tree of G with "shortest" edge dashed; (c) maximal spanning tree with second shortest edge dashed; (d) first split resulting in two connected components; (e) first split of banana; (f) second split of banana.

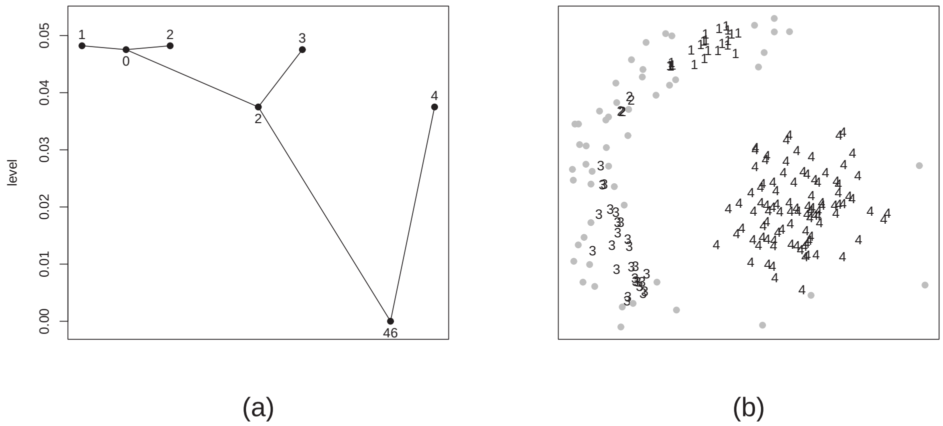


Figure 4: (a) Graph cluster tree; (b) clusters corresponding to leaves of graph cluster tree. Numbers above leaves are labels, numbers below interior nodes are runt excess masses.

We now illustrate the cluster tree algorithm on a simple two-dimensional example. Figure 3(a) shows a data set consisting of two obvious groups, which we will refer to as the "lump" and the "banana". Superimposed are the isopleths of a kernel density estimate.

Figure 3(b) shows the maximal spanning tree of G. The "shortest" edge, the one with lowest edge weight $\hat{p}_{ij}$ and the first one to be broken during the recursive thresholding process, is dashed. The minimum of the density along this edge is assumed at one of the end points, drawn in grey. Therefore, thresholding eliminates this edge and the end point, leaving us with one connected component.

Figure 3(c) illustrates the second step of the algorithm. The second shortest edge, the second one to be broken, is dashed; edges and vertices below the current threshold are drawn in grey. Again, the minimum of the density along the edge is assumed at an end point, and thresholding leaves us with one connected component.

The thresholding process progressively peels off edges and vertices until we reach the stage shown in Figure 3(d), where for the first time thresholding results in two connected components, essentially the lump and the banana, with a few low density points removed.

Applying the thresholding process to the lump does not result in any more splits - edges and vertices are removed until we are left with an empty graph. We therefore focus on the banana. Figure 3(e) shows the first split of the banana. There are no further splits of the lower part of the banana, while there is one additional split of the upper part, shown in Figure 3(f).

The graph cluster tree shown in Figure 4(a) has four leaves, corresponding to the lump and the three fragments of the banana. In Figure 4(b) observations in the high density clusters corresponding to the leaves of the tree are indicated by numbers; the remaining observations (the *fluff*) are drawn in grey. The numbers below the interior nodes of the tree are their runt excess masses (Section 5).

Remark 2: The density estimate has at least one additional mode, visible in Figure 3(a) between the lump and the banana, that does not manifest itself in the cluster tree because there are no observations in its vicinity. The valley between the two modes in the upper part of the banana is shallow and not visible in Figure 3(a) due to the choice of contour levels.

Remark 3: The maximal spanning tree edge connecting the lump and the banana in Figure 3(d) might seem implausible. Note, however, that there are many edges of the complete graph G with very similar edge weights crossing the density valley separating the lump from the banana. Which of those has the largest edge weight and therefore ends up in the maximal spanning tree depends on minor details of the density estimate and the locations of the grid points along the edges.

Remark 4: For the data in this example we would hope to obtain a graph cluster tree with two leaves corresponding to the lump and the banana, respectively. However, density estimates are inherently noisy, and the occurrence of spurious modes is unavoidable. Note, though, that the two valleys separating the three spurious modes in the banana are shallow, and the separation between them is not nearly as clear as the separation between the lump and the banana. This fact is not apparent from the graph cluster tree in Figure 4(a) because the tree only indicates the levels of the valleys, not the heights of the peaks. In Section 5 we propose a measure for the "prominence" of a high density cluster incorporating both its spatial extent and the rise of its peak (or peaks) above the valley separating it from the rest of feature space. Given such a measure, we can then prune branches of the graph cluster tree corresponding to clusters with low prominence.

Remark 5: As illustrated in Figure 4(b), the graph high density clusters corresponding to the leaves of the graph cluster tree do not form a partition of the data. If we want a partition, we need a way of assigning the fluff to the clusters. In keeping with the recursive nature of the clustering process, it is natural to make this assignment recursively. Consider Figure 3(d) where we make the first split. The graph high density clusters corresponding to the daughters of the root node are the solid black points in the lump and the banana, respectively. The grey points are fluff, and the picture suggests a way of assigning the fluff to the graph high density clusters: Breaking the dashed edge splits the maximal spanning tree into two subtrees, and we assign each fluff point to the high density cluster in its subtree. The same recipe can be applied at any stage of the algorithm. A problem with this method is that it occasionally results in counter-intuitive assignment of outliers. The minimum densities along the edges in G connecting an outlier to the rest of the observations will all be small, and which of them is the smallest will depend, for example, on the locations of the grid points used to approximate the edge weights. An alternative is to assign the fluff using a nearest neighbor rule. The details of fluff assignment do not appear to make much difference in terms of performance.

5 Pruning the graph cluster tree

There is an obvious way of measuring the prominence of a high density cluster in the population case. Consider Figure 2 showing a density with three modes and the corresponding cluster tree. Recall that each node N of the cluster tree represents a subset $D(N)$ of the feature space and is associated with a level $\lambda(N)$. We propose to measure the prominence of a high density cluster by its excess mass

$$E(N) = \mathcal{E}(\lambda, D(N); P) = \int_{D(N)} (p(\mathbf{x}) - \lambda(N))\, d\mathbf{x}\,.$$

In Figure 2(a) the excess mass associated with the left daughter of the root node is represented by the shaded area.

To find a sample analogon to $E(N)$ observe that

$$\begin{aligned} \int_{D(N)} d\mathbf{x} &= \int_{D(N)} \frac{1}{p(\mathbf{x})}\, p(\mathbf{x})\, d\mathbf{x} \\ &\sim \frac{1}{n} \sum_i I(\mathbf{x}_i \in \tilde{D}(N))\, \frac{1}{p(\mathbf{x}_i)}\,, \end{aligned}$$

and therefore

$$E(N) \sim \tilde{E}(N) = \frac{1}{n} \sum_i I(\mathbf{x}_i \in \tilde{D}(N))\, (1 - \frac{\lambda(N)}{p(\mathbf{x}_i)})\,.$$

The estimate $\tilde{E}(N)$ for $E(N)$ may be poor if the number of observations in the corresponding graph high density cluster $\tilde{D}(N)$ is small. However, $\tilde{E}(N)$ is a sensible measure of prominence. If the estimated densities for the observations in $\tilde{D}(N)$ are close to λ (low elevation of the peak above the valley floor) then $\lambda(N)/p(\mathbf{x}_i) \approx 1$ and $\tilde{E}(N)$ is small. If the peak has a high elevation above the valley floor, on the other hand, then $\lambda(N)/p(\mathbf{x}_i) \approx 0$ and $\tilde{E}(N)$ is large.

In order to prune the graph cluster tree we choose an excess mass threshold γ and remove all nodes with excess mass $\tilde{E}(N) < \gamma$ and their incident edges. Note that excess mass is monotone: If node N_2 is a descendant of N_1 then $\tilde{E}(N_2) < \tilde{E}(N_1)$. Monotonicity implies that pruning will not result in any isolated branches or nodes. The resulting graph may no longer be a binary tree, but it can be converted into one by splicing out degree 2 nodes.

The nodes of the graph cluster tree surviving the pruning process are those whose daughters both have excess mass $> \gamma$. Define the *runt excess mass* of an interior node as the smaller of the excess masses of its two daughters. The numbers 46, 2, and 0 next to the interior nodes of the graph cluster tree in Figure 4(a) are the runt excess masses, multiplied by the sample size and rounded for readability. (Informally we use the term "excess mass" for both $\tilde{E}(N)$ and round$(n\, \tilde{E}(N))$. Multiplying by the sample size expresses excess mass in units of observations.) Clearly there is only one split separating two prominent peaks of the estimated density, namely the one represented by the root node; the remaining two split off minor bumps. Pruning the graph cluster tree with excess mass threshold 46 results in a tree with two leaves representing the lump and the banana, respectively.

It would be desirable to have an automatic method for determining an appropriate value for the threshold γ. We do not yet have such a method, so the choice will have to be subjective (see Section 7).

Remark 6: A simpler measure of "significance" of a mode is its size

$$S(N) = \int_{D(N)} p(\mathbf{x})\, d\mathbf{x}$$

which can be estimated by

$$\tilde{S}(N) = \frac{1}{n} \sum_i I(\mathbf{x}_i \in \tilde{D}(N))\,.$$

In Figure 2(a) the size associated with the left daughter of the root node is represented by the hashed area. Hartigan and Mohanty (1992) used size as an indicator for "significance" in their runt test for unimodality. Like excess mass, size is monotone and can be used for pruning the graph cluster tree. The runt sizes for the three interior nodes of the tree in Figure 4(a) are 96, 26, and 5. So in this example, runt size does not provide as clear a guide for pruning as runt excess mass.

6 Connections to single linkage and k-th nearest neighbor clustering

Single linkage and generalized single linkage clustering are connected through the nearest neighbor density estimate

$$\hat{p}^{(1)}(\mathbf{x}) = \frac{1}{d_1(\mathbf{x}, \mathcal{X})}\,,$$

where $d_1(\mathbf{x}, \mathcal{X}) = \min_i d(\mathbf{x}, \mathbf{x}_i)$. In a way, $\hat{p}^{(1)}$ barely deserves the name "density estimate": it has a singularity at every observation and cannot even be normalized. On the other hand, it does provide a sensible measure of density in the non-technical sense of the word: $\hat{p}^{(1)}(\mathbf{x})$ is small if $\mathbf{x}$ is far away from the observations, and large if $\mathbf{x}$ is close.

Let G be the complete graph over the observations with edge weights

$$\hat{p}^{(1)}_{ij} = \min_{t \in [0,1]} \hat{p}^{(1)}((1-t)\,\mathbf{x}_i + t\,\mathbf{x}_j)$$

and vertex weights $\hat{p}^{(1)}_{ii} = \infty$.

Proposition 2: The graph cluster tree of G is isomorphic to the single linkage dendogram.

A proof of Proposition 2 is given in the appendix.

The commonly used method of extracting clusters from a single linkage dendogram is dendogram cutting. Stated in terms of the graph cluster tree, dendogram cutting is equivalent to choosing a density threshold λ^* and removing all nodes with level $\lambda(N) > \lambda^*$ and their incident edges. There are two problems with this pruning strategy. First it tends to result in many singletons or tiny clusters consisting of outliers, and one or a few large clusters. This problem could be remedied by choosing a size threshold and discarding all clusters of size smaller than the threshold. However, there is a more fundamental problem: dendogram cutting forms the clusters based on a single level set of the nearest neighbor density estimate and, as Figure 2 illustrates, there may not be a single level revealing all the groups or modes. An alternative is to apply the pruning method described in Section 5. Note that the nearest neighbor density estimate has a singularity at each data point ($\hat{p}^{(1)}(\mathbf{x}_i) = \infty$) and therefore the measure of prominence

$$\tilde{E}(N) = \frac{1}{n} \sum_i I(\mathbf{x}_i \in \tilde{D}(N)) \, (1 - \frac{\lambda(N)}{p(\mathbf{x}_i)})$$

reduces to the fraction of observations in the graph high density cluster $\tilde{D}(N)$, i.e., to size. Extracting clusters from a single linkage dendogram by pruning branches with small size was proposed by Stuetzle (2003), who also provided experimental results suggesting that pruning is vastly superior to dendogram cutting.

There is also a connection between generalized single linkage clustering and Wong's k-th nearest neighbor clustering (Wong 1979; Wong and Lane 1983). Wong and Lane's method was motivated by the realization that the nearest neighbor density estimate is not consistent and that therefore the level sets of the nearest neighbor density estimate will not be consistent estimates of the level sets of the feature density. Instead they use the k-th nearest neighbor density estimate

$$\hat{p}^{(k)}(\mathbf{x}) \sim \frac{1}{d_k(\mathbf{x}, \mathcal{X})} \, ,$$

where $d_k(\mathbf{x}, \mathcal{X})$ is the k-th smallest distance between $\mathbf{x}$ and one of the $\mathbf{x}_i$.

Wong and Lane's method has three steps: (i) construct the complete graph over the observations with edge weights $w_{ij} = 1/2 \; (1/\hat{p}^{(k)}(\mathbf{x}_i) + 1/\hat{p}^{(k)}(\mathbf{x}_j))$ if x_i is among the (Euclidean) k nearest neighbors of $\mathbf{x}_j$ or $\mathbf{x}_j$ is among the k nearest neighbors of $\mathbf{x}_i$ (i.e., the edge connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ is in the k-nearest neighbor graph) and $w_{ij} = \infty$ otherwise; (ii) calculate the minimal spanning tree of the edge weighted graph; (iii) compute the single linkage dendogram from the minimal spanning tree.

To see the connection to generalized single linkage note that: (i) the minimal spanning tree for edge weights w_{ij} is the maximal spanning tree for edge weights $1/w_{ij}$; (ii) if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close in Euclidean distance (i.e., the connecting edge is in the k-nearest neighbor graph) then $\hat{p}^{(k)}$ is roughly constant along the line segment connecting $\mathbf{x}_i$ and

$\mathbf{x}_j$ and thus $1/w_{ij} \sim \hat{p}_{ij}^{(k)}$; (iii) the maximal spanning tree of G will mostly connect observations that are close in Euclidean distance. Therefore, zero-ing out the weights for edges of G not in the k-nearest neighbor graph will leave the maximal spanning tree and the structure of its threshold graphs basically unchanged.

7 Examples

The goal of this section is to illustrate generalized single linkage clustering on some examples. In the context of the examples we also compare different density estimates and alternative ways of computing the edge weights, and we investigate the correspondence between leaves of the graph cluster tree and modes of the density estimate.

Density estimation: We compare clustering results for two different density estimates: the nearest neighbor density estimate and a kernel density estimate with spherical Gaussian kernel and bandwidth determined by least squares cross-validation (Silverman 1986). The nearest neighbor estimate is computationally attractive because the maximal spanning tree of G is the Euclidean minimal spanning tree of the observations, and computing a Euclidean minimal spanning tree for 10,000 points in ten dimensions only takes about a minute on a standard PC. We chose kernel estimates as the competitor because they are well understood and easy to implement, and least squares cross-validation offers a simple way for automatic bandwidth selection. Unless otherwise noted, we sphere the data before clustering. Sphering is advisable when using automatic bandwidth selection with a spherical kernel; otherwise the bandwidth is essentially determined by the variance of the smallest principal component.

Edge weights: For a kernel density estimate the edge weights

$$\hat{p}_{ij} = \min_{t \in [0,1]} \hat{p}((1-t)\,\mathbf{x}_i + t\,\mathbf{x}_j)$$

are not available in closed form and have to be approximated. A simple approximation method is grid search: approximate $\hat{p}_{ij}$ by the minimum of $\hat{p}$ over a regular grid on the line segment connecting $\mathbf{x}_i$ and $\mathbf{x}_j$. In the examples we used ten grid points. We compare the clustering results for grid search with the results for a global optimization method (Nugent, 2006) that is guaranteed to produce the correct graph cluster tree (see Section 4).

Choosing an excess mass threshold for pruning: We sort the runt excess masses for the interior nodes of the graph cluster tree in decreasing order. Typically there is a small number of large values followed by a long trail of small values, like 98, 32, 22, 4, 3, 3, 3, 2, 2, 1, 1,... in Example 1 below. A large runt excess mass indicates a split separating two prominent modes whereas a small runt excess mass indicates separation of a spurious mode most likely caused by variability of the density estimate. We scan the values from small to large looking for the first clear break, in our example between

4 and 22, and then choose the larger value as the threshold. In our case there are three runt excess masses greater than or equal to the threshold, leading to a pruned tree with three interior nodes and four leaves.

Leaves versus modes: There is a one-to-one correspondence between modes of $\hat{p}$ and leaves of the cluster tree of $\hat{p}$. As pointed out in Section 3, however, the same is not necessarily true for the graph cluster tree. The graph cluster tree may fail to reflect modes of $\hat{p}$ whose domain of attraction does not contain any observations and, more importantly, multiple leaves may correspond to the same mode due to spurious splits of level sets of $\hat{p}$. To see how much of a problem is presented by incorrect splits, we use numerical optimization. We start a numerical optimizer at each of the n observations and then cluster the resulting local optima using Ward's clustering method, a hierarchical version of k-means clustering. Define the loss associated with a partition as the sum of squared distances of the observations from their closest cluster means. Initially, every observation is a cluster. At any stage of the algorithm, Ward's method merges the two clusters leading to the smallest increase in loss. When applying Ward's method to the local optima there typically is a clear jump in the loss. A jump after the i-th merge indicates that the local optima fall into $n - i$ clusters corresponding to $n - i$ modes.

Measuring agreement between partitions: Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two partitions of a set of n objects. The partitions define a contingency table: let n_{ij} be the number of objects that belong to subset i of partition $\mathcal{P}_1$ and to subset j of partition $\mathcal{P}_2$. We measure the agreement between $\mathcal{P}_1$ and $\mathcal{P}_2$ by the adjusted Rand index (Hubert and Arabie 1985) defined as

$$R = \frac{\sum_{ij} \binom{n_{ij}}{2} - \sum_i \binom{n_{i\cdot}}{2} \sum_j \binom{n_{\cdot j}}{2} / \binom{n}{2}}{\frac{1}{2}\left(\sum_i \binom{n_{i\cdot}}{2} + \sum_j \binom{n_{\cdot j}}{2}\right) - \sum_i \binom{n_{i\cdot}}{2} \sum_j \binom{n_{\cdot j}}{2} / \binom{n}{2}} .$$

Here $n_{i\cdot} = \sum_j n_{ij}$, and $n_{\cdot j}$ is defined analogously.

The adjusted Rand index has a maximum value of 1 which is achieved when the two partitions are identical up to re-numbering of the subsets. It has expected value 0 under random assignment of the objects to the subsets of $\mathcal{P}_1$ and $\mathcal{P}_2$ that leave the marginals $n_{i\cdot}$ and $n_{\cdot j}$ fixed.

7.1 Three artificial examples based on the Olive Oil data

The Olive Oil data consist of measurements of eight chemical components on 572 samples of olive oil. The samples come from three different regions of Italy. The regions are further partitioned into nine areas: areas A1 . . . A4 belong to region R1, areas A5 and A6 belong to region R2, and areas A7. . . A9 belong to region R3.

Figure 5 shows a two-dimensional data set "Olive-5-2d" obtained by projecting the five areas A5. . . A9 on the plane spanned by the first two Fisher discriminant coordinates and then sphering the projected data. We chose this data set as an example because the areas are distinct but have different densities, shapes, and degrees of separation.

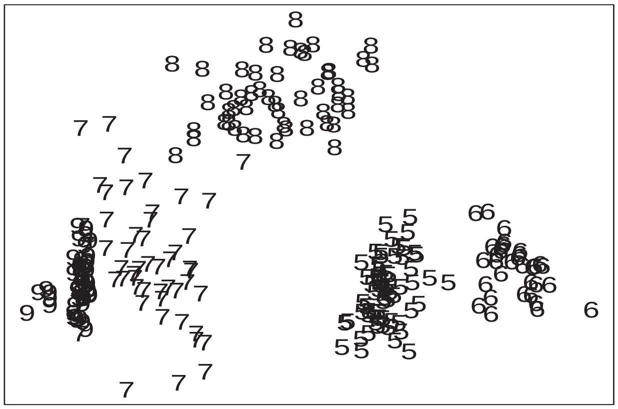


Figure 5: Areas A5 - A9 of Olive Oil data, projected on first two discriminant coordinates.

In Example 1 we cluster Olive-5-2d. For Examples 2 and 3 we add three and eight independent standard Gaussian noise variables, respectively, obtaining data sets "Olive-5-5d" and "Olive-5-10d". By construction, all the group information in Olive-5-5d and Olive-5-10d is contained in the first two variables, which makes it easy to display and compare clustering results.

Example 1: Applying least squares cross-validation to Olive-5-2d gives bandwidth $h = 0.07$. The unpruned graph cluster tree of the corresponding kernel density estimate has 49 leaves, suggesting that the density estimate has 49 modes. To obtain an alternative estimate for the number of modes we start a numerical optimizer at each of the 249 observations and apply Ward's method to the local optima. The loss for the first 202 merges stays below 5×10^{-4} and then abruptly jumps to 1.7×10^{-1}, suggesting that there are at least $249 - 202 = 47$ distinct local optima. We conclude that the kernel estimate indeed has roughly 50 modes.

The runt excess masses of the graph cluster tree (sorted in decreasing order, multiplied by the sample size, and rounded for easier parsing), are 98, 32, 22, 4, 3, 3, 3, 2, 2, 1, 1,..., suggesting a runt excess mass threshold of 22 for pruning. The resulting pruned tree has four leaves. For comparison, the runt sizes are 98, 47, 32, 14, 13, 11, 7, 6, 5, 5,...; runt size pruning with threshold 32 gives the same pruned tree as runt excess mass pruning with threshold 22. The graph high density clusters corresponding to the leaves of the pruned tree are shown in Figure 6(a). Black symbols represent observation in cluster cores, grey symbols represent fluff. The method is unable to separate areas

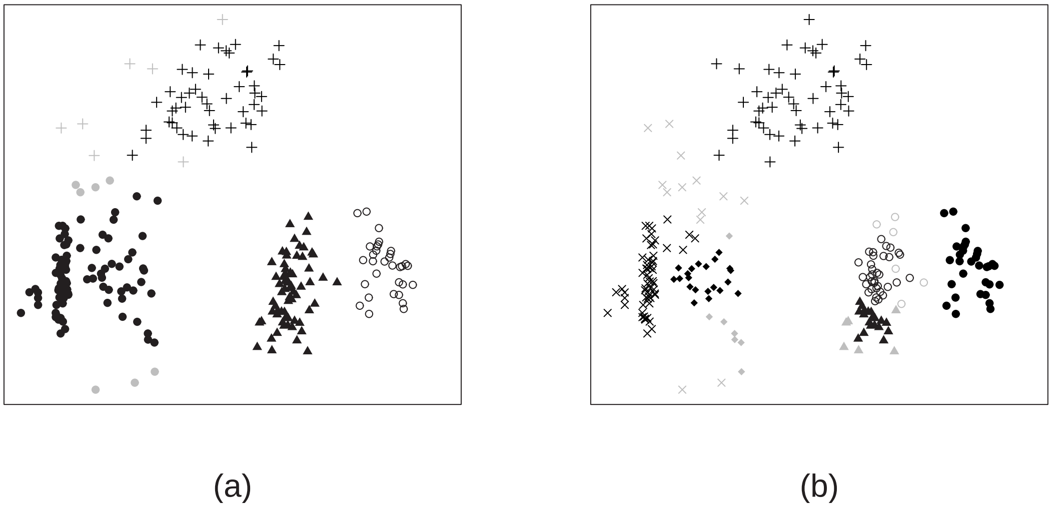


Figure 6: (a) Clustering of Olive-5-2d based on kernel density estimate; (b) clustering based on nearest neighbor estimate.

A7 and A9. The adjusted Rand index is 0.75.

Approximating the edge weights using grid search gives almost the same results as using the exact edge weights. The rand index comparing the respective four-cluster partitions is 0.98.

Figure 6(b) shows the corresponding result for the nearest neighbor density estimate. The runt sizes are 98, 51, 32, 21, 19, 12, 10, 10, 9, 9, 8,..., suggesting a runt size threshold of 19 and a six cluster solution. The core of area A7 is now recognized as distinct from area A9, but area A5 is erroneously split. The rand index is 0.72.

Example 2: We now move on to the five-dimensional data set Olive-5-5d. The bandwidth chosen by least squares cross-validation is 0.45. The unpruned graph cluster tree has 43 leaves. The alternative estimate for the number of modes, using optimization as described above, is 64.

The runt excess masses of the graph cluster tree are 21, 3, 1, 1,..., suggesting runt excess mass threshold 21 for pruning and two clusters. The runt sizes are 72, 21, 3, 2, 2, 2, ..., suggesting a runt size threshold of 21 and three clusters. This is one of the rare cases we have encountered where pruning based on excess mass and pruning based on size result in different numbers of clusters. The adjusted Rand index for two clusters is 0.38, versus 0.58 for three clusters. Figure 7(a) shows the three cluster solution. The adjusted Rand indices for comparing approximate and exact edge weights are 0.98 for both solutions.

Figure 7(b) shows the corresponding result for the nearest neighbor density estimate.

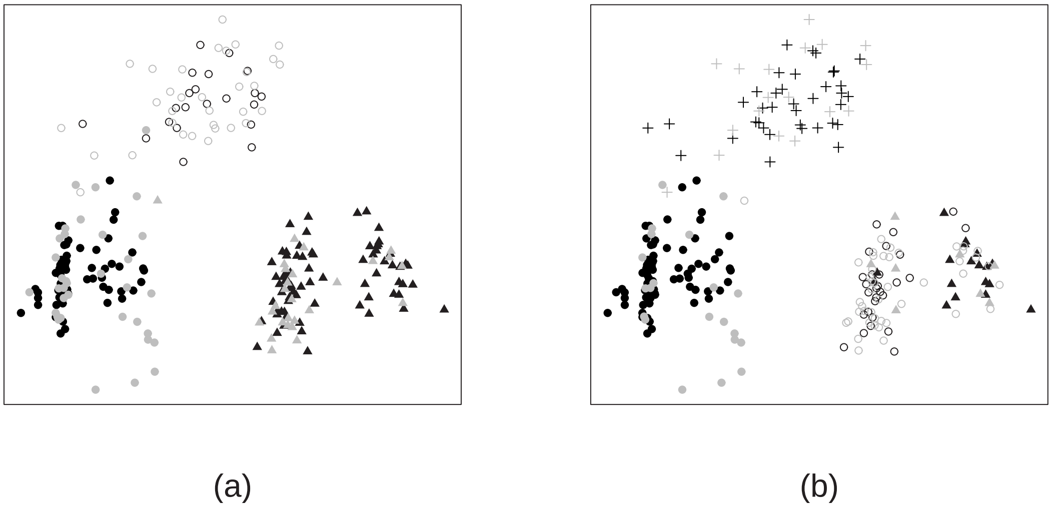


Figure 7: (a) Clustering of Olive-5-5d based on kernel density estimate; (b) clustering based on nearest neighbor estimate.

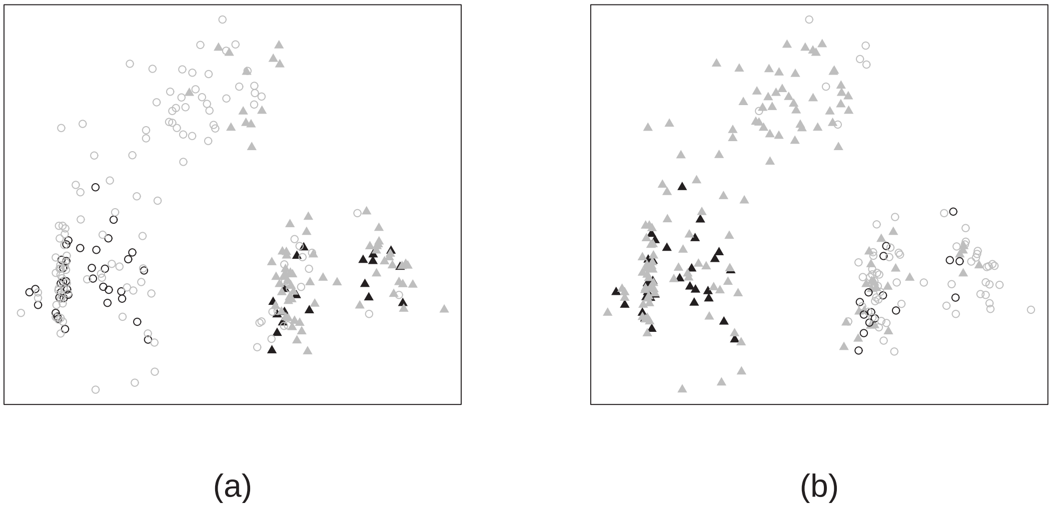
The runt sizes are 87, 37, 18, 11, 11, 8, 8 7, 5,. . . , suggesting a four cluster solution. Areas A7 and A9 are merged, but the remaining areas are correctly separated. The adjusted Rand index is 0.62.

Example 3: Finally we consider the ten-dimensional data set Olive-5-10d. The bandwidth chosen by least squares cross-validation is 0.68. The unpruned graph cluster tree has 209 leaves. The alternative estimate for the number of modes is 219. The runt excess masses are 5, 3, 2, 2, 2,. . . and the runt sizes are 19, 8, 7, 5, 5, 4, 4,. . . , suggesting one or two clusters, respectively. Figure 8(a) shows the solution, which separates region R2 from region R3. The adjusted Rand index is 0.17. The adjusted Rand index for comparing approximate and exact edge weights is 0.97.

Figure 8(b) shows the corresponding result for the nearest neighbor density estimate. The runt sizes are 16, 9, 9, 6, 5, 5, 5,. . . , suggesting a two cluster solution. Again, regions R2 and R3 are separated, but there are only very few observations in the cluster cores. The adjusted Rand index is 0.17.

7.2 The Olive Oil data

We show the results for the kernel estimate. Least squares cross-validation gives a bandwidth of 0.23. The unpruned graph cluster tree has 514 leaves. The alternative estimate for the number of modes is 501. The runt excess masses are 128, 86, 46, 26,



(a)

(b)

Figure 8: (a) Clustering of Olive-5-10d based on kernel density estimate; (b) clustering based on nearest neighbor estimate.

24, 24, 18, 17, 11, 9, 8, 7, 7, 6, 6, . . . , and the runt sizes are 129, 89, 47, 33, 25, 24, 22, 20, 11, 9, 9, 8,. . . , both suggesting a nine cluster solution. Figure 9 shows the graph cluster tree. Table 1 shows a table of area (vertical axis) against leaf code (horizontal axis). Generalized single linkage clustering is unable to isolate area A4; area A3, which has by far the largest number of observations, is split into two clusters (leaf codes 62 and 63); and areas A7 and A8 are not cleanly separated. The adjusted Rand index is 0.62, reflecting the erroneous split of area A3. The adjusted Rand index for comparing approximate and exact edge weights is 0.98.

The runt sizes for the nearest neighbor estimate are 129, 89, 47, 33, 25, 25, 24, 20, 11, 11, 9, 9, . . . , again suggesting nine clusters. The results are virtually indistinguishable from those for the kernel estimate.

7.3 The Acute Lymphoblastic Leukemia data

The purpose of this example is to illustrate that generalized single linkage clustering can be applied to very high-dimensional data sets. The Acute Lymphoblastic Leukemia (ALL) data are oligonucleotide microarray gene expression levels of 12558 genes for each of 360 ALL patients. Yeoh *et al.* (2002) divided the patients into seven diagnostic groups corresponding to six known leukemia subtypes (T-ALL, E2A-PBX1, BCR-ABL, TELAML1, MLL rearrangement, and Hyperploid>50 chromosomes), and one unknown type, labeled OTHER. The data were taken from the Kent Ridge Bio-Medical Data Set Repository, where they have been split into training and test sets. We clustered the

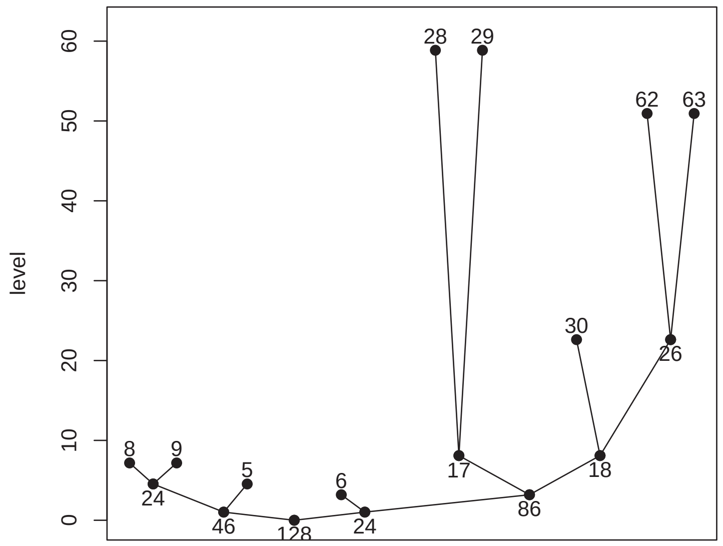


Figure 9: Graph cluster tree of Olive Oil data for kernel density estimate. Numbers above the leaves are labels; numbers below the interior nodes are runt excess masses.

training set comprising 215 patients.

We first selected the 1000 genes with the highest variance and normalized the expression profiles to have zero mean and unit variance; squared Euclidean distance between patients then measures the correlation between the corresponding expression profiles. Sphering does not make sense in this example, as the observations lie in a 213-dimensional subspace of 1000-dimensional space. Next, we computed the graph cluster tree for the nearest neighbor density estimate (the single linkage dendogram). The largest runt sizes are 36, 27, 21, 14, 8, 5, 5,. . . , suggesting five clusters. Figure 10 shows the (pruned) graph cluster tree, and Table 2 shows a table of ALL subtype (vertical axis) against leaf code (horizontal axis). The T-ALL, E2A-PBX1, and TEL-AML1 subtypes correspond to clusters with leaf codes 3, 4, and 23; the remaining subtypes are not isolated. These results are qualitatively similar to the ones obtained by Murua, Stanberry, and Stuetzle (2007) using Potts model clustering. The adjusted Rand index of the generalized single linkage partition is 0.55, compared to 0.53 for Potts model clustering.

	6	62	63	30	28	29	9	8	5
A1	24	1	0	0	0	0	0	0	0
A2	0	1	6	49	0	0	0	0	0
A3	0	95	108	3	0	0	0	0	0
A4	5	0	10	20	0	0	0	1	0
A5	0	0	0	0	64	1	0	0	0
A6	0	0	0	0	5	28	0	0	0
A7	0	0	0	0	0	0	32	16	2
A8	0	0	0	0	0	1	0	49	0
A9	0	0	0	0	0	0	0	0	51

Table 1: Olive oil data: leaf code (horizontal axis) tabulated against area (vertical axis).

	3	4	10	22	23
BCR-ABL	0	0	0	6	3
E2A-PBX1	0	18	0	0	0
Hyperdip>50	0	1	0	41	0
MLL	0	4	10	0	0
OTHERS	0	2	14	24	12
T-ALL	28	0	0	0	0
TEL-AML1	0	0	0	0	52

Table 2: ALL data: leaf code (horizontal axis) tabulated against ALL subtype (vertical axis).

8 Summary and discussion

The goal of clustering is to detect the presence of distinct groups in a data set. Nonparametric clustering is based on the premise that groups correspond to modes of the feature density. The goal then is to detect modes of the density and assign each observation to the domain of attraction of a mode. The modal structure of a density is summarized by its cluster tree; the modes of the density correspond to the leaves of the cluster tree. We have pursued a plug-in approach to cluster tree estimation: estimate the cluster tree of the feature density by the cluster tree of a density estimate. For some density estimates the cluster tree can be computed exactly, for others we have to be content with an approximation. We have developed a graph-based method that can approximate the cluster tree of any density estimate. Due to sampling variability, density estimates tend to have spurious modes that do not reflect modes of the feature density and that will lead to spurious branches in the graph cluster tree. We have proposed excess mass as a measure for the size of branches of the graph cluster tree, reflecting the height of the corresponding peak or peaks of the density above the surrounding valley floor and its spatial extent. Excess mass can be used as a guide for subjective pruning of the graph cluster tree. The graph cluster tree of the nearest neighbor density estimate is (essen-

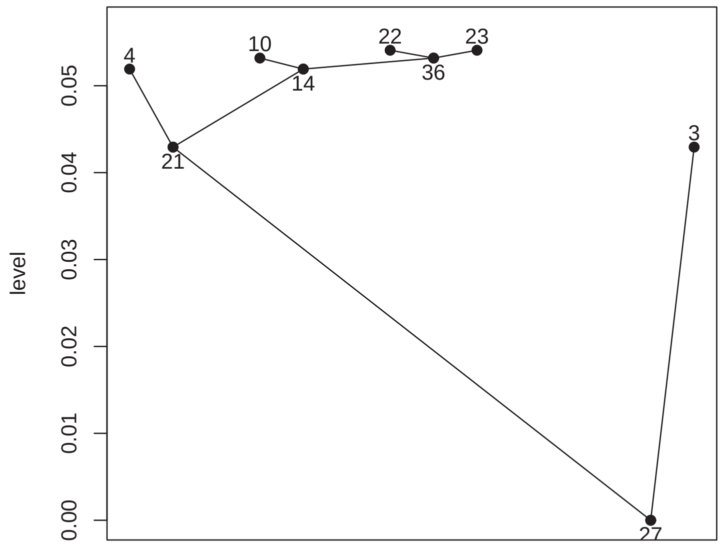


Figure 10: Graph cluster tree of ALL data for nearest neighbor density estimate. Numbers above the leaves are labels; numbers below the interior nodes are runt sizes.

tially) the single linkage dendogram. Excess mass pruning generalizes the runt pruning method for extracting clusters from the single linkage dendogram proposed by Stuetzle (2003).

In the examples presented in Section 7, as well as in about a half dozen others not reported here, we have observed that:

(1) Kernel estimates with span determined by least squares cross-validation tend to have many modes, most of them caused by sampling variability, and pruning the graph cluster tree is crucial.

(2) Approximating the edge weights for kernel estimates by grid search gives clustering results very similar to those obtained using the exact edge weights. The likely reason is that only high density clusters separated from the rest of the data by deep valleys survive pruning, and such valleys are easy to locate even by a crude optimization method like grid search.

(3) Kernel estimates with bandwidth determined by least squares cross-validation and nearest neighbor density estimates give comparable clustering performance.

Observation (3) came as a pleasant surprise, as the calculations for the nearest neighbor estimate are much faster. Note, however, that there may be other methods for bandwidth selection and/or other density estimates resulting in better clustering results than the nearest neighbor estimate.

There are several directions for future work:

Other density estimates. Kernel and near neighbor density estimates are known to be susceptible to the curse of dimensionality. It may be worthwhile to investigate the performance of generalized single linkage clustering with other density estimates potentially less impacted by high dimensionality, like Projection Pursuit density estimates (Friedman, Stuetzle, and Schroeder 1984; Friedman 1987).

Alternative pruning strategies. Excess mass pruning is based solely on the prominence of peaks of the estimated density, i.e. their height and spatial extent; it does not take into account the spatial separation between peaks. Pruning strategies taking into account both prominence and separation may allow for better detection of small but highly isolated groups.

Automatic pruning. Subjective pruning casts doubts on interpretations of clustering results and makes quantitative comparisons of results difficult. A fully automatic pruning method (analogous to model selection methods for regression and classification) would represent a big advance.

9 Appendix

Prop 1: Let G be an edge weighted graph and T its maximal spanning tree. Then two vertices belong to the same connected component of the threshold graph $G(\lambda)$ iff they belong to the same connected component of $T(\lambda)$.

Proof of Prop. 1:

Two vertices in the same connected component of $T(\lambda)$ are in the same connected component of $G(\lambda)$ because the edges of T are a subset of the edges of G.

Now assume that vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ are in different connected components of $T(\lambda)$. This means that the unique path in T connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ contains at least one edge e with weight $\leq \lambda$. Removing e from T breaks T into two connected components T_1 and T_2, one containing $\mathbf{x}_i$ and the other containing $\mathbf{x}_j$. If $\mathbf{x}_i$ and $\mathbf{x}_j$ were in the same connected component of $G(\lambda)$ there would be a path in G connecting $\mathbf{x}_i$ and $\mathbf{x}_j$ for which all edge weights are greater than λ. This path has to contain an edge e^* connecting T_1 and T_2. Replacing e with e^* in T would lead to a tree with larger total edge weight, contradicting the assumption that T was the maximal spanning tree of G.

Prop 2: Let G be the complete graph over the observations with edge weights

$$\hat{p}_{ij}^{(1)} = \min_{t \in [0,1]} \hat{p}^{(1)}((1-t)\,\mathbf{x}_i + t\,\mathbf{x}_j)$$

and vertex weights $\hat{p}_{ii}^{(1)} = \infty$. Then the graph cluster tree of G is isomorphic to the single linkage dendogram.

Lemma 1: A point $\mathbf{x}$ has estimated density $\hat{p}^{(1)}(\mathbf{x}) > \lambda$ iff it is within distance $r = 1/\lambda$ of at least one of the data points:

$$L(\lambda; \hat{p}^{(1)}) = \bigcup_i S(\mathbf{x}_i, r)\,,$$

where $S(\mathbf{x}, r)$ denotes the (open) sphere around $\mathbf{x}$ with radius r.

Lemma 2: $\hat{p}_{ij}^{(1)} \geq 2/d(\mathbf{x}_i, \mathbf{x}_j)$.

Proof of Lemma 2: Suppose there are no other data points in the sphere around the midpoint $(\mathbf{x}_i + \mathbf{x}_j)/2$ with radius $d(\mathbf{x}_i, \mathbf{x}_j)/2$. Then $\hat{p}_{ij}^{(1)} = 2/d(\mathbf{x}_i, \mathbf{x}_j)$.The presence of other data points in the sphere can only increase $\hat{p}_{ij}^{(1)}$.

Lemma 3: Let (k, l) be an edge of G with weight $\hat{p}_{kl}^{(1)} > \lambda$. Then there is a path connecting $\mathbf{x}_k$ and $\mathbf{x}_l$ with maximum edge length $< 2/\lambda$.

Proof of Lemma 3: As $\hat{p}_{kl}^{(1)}$ is the minimum of the nearest neighbor density estimate over the line segment $[\mathbf{x}_k, \mathbf{x}_l]$, the assumption that $\hat{p}_{kl}^{(1)} > \lambda$ implies that the entire line segment $[\mathbf{x}_k, \mathbf{x}_l]$ is covered by spheres around the observations with radius $r = 1/\lambda$ (Lemma 1):

$$[\mathbf{x}_k, \mathbf{x}_l] \subset \bigcup_i S(\mathbf{x}_i, r)\,.$$

Let

$$L_q = [\mathbf{a}_q, \mathbf{b}_q] = [\mathbf{x}_k, \mathbf{x}_l] \cup S(\mathbf{x}_q, r)$$

be the (possibly empty) intersection of the line segment $[\mathbf{x}_k, \mathbf{x}_l]$ with the sphere of radius r around $\mathbf{x}_q$. Without loss of generality assume that $d(\mathbf{x}_k, \mathbf{a}_q) \leq d(\mathbf{x}_k, \mathbf{b}_q)$. Choose $q_1 = k$. Because the L_q collectively cover $[\mathbf{x}_k, \mathbf{x}_l]$ there has to be a q_2 with $b_{q_1} \in S(\mathbf{x}_{q_2}, r)$. Therefore, $d(\mathbf{x}_{q_1}, \mathbf{x}_{q_2}) < 2r$. Repeating this argument shows that there is a path connecting $\mathbf{x}_k$ and $\mathbf{x}_l$ with maximum edge length $< 2r = 2/\lambda$.

Lemma 4: The graph cluster tree of G and the cluster tree of the nearest neighbor density estimate are isomorphic.

Proof of Lemma 4: Let $\mathcal{X}_1, \ldots, \mathcal{X}_k$ be vertex sets of the connected components of $G(\lambda)$. We will show that the connected components of $L(\lambda; \hat{p}^{(1)})$ are the sets

$$L_i = \bigcup_{\mathbf{x}_j \in \mathcal{X}_i} S(\mathbf{x}_j, 1/\lambda)\,.$$

Suppose that L_i is connected. Then for any two vertices $\mathbf{x}_j$ and $\mathbf{x}_l$ in $\mathcal{X}_i$ there exists a polyline connecting them with maximal edge length $< 2/\lambda$ and therefore minimum density $\hat{p}^{(1)} > \lambda$. This implies that $\mathbf{x}_j$ and $\mathbf{x}_l$ are in the same connected component of $G(\lambda)$.

On the other hand, suppose that $\mathbf{x}_j$ and $\mathbf{x}_l$ are in the same connected component of $G(\lambda)$. This implies that they are connected by a path with minimum edge weight $> \lambda$ and therefore maximum edge length $< 2/\lambda$ (Lemma 3), and hence are in the same connected component of $L(\lambda; \hat{p}^{(1)})$.

Proof of Proposition 2: According to Lemma 4, the graph cluster tree and the cluster tree of the nearest neighbor density estimate are isomorphic. On the other hand, Stuetzle (2003, Section 2) has shown that the cluster tree of the nearest neighbor density estimate is isomorphic to the single linkage dendogram.

References

[1] M. Ankerst, M.M. Breuning, H.P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings, ACM SIGMOD International Conference on Management of Data (SIGMOD'99)*, pages 49–60, 1999.

[2] J.W Carmichael, G.A. George, and R.S. Julius. Finding natural clusters. *Systematic Zoology*, 17:144–150, 1968.

[3] A. Cuevas, M. Febrero, and R. Fraiman. Estimating the number of clusters. *Canadian Journal of Statistics*, 28(2):367–382, 2000.

[4] A. Cuevas, M. Febrero, and R. Fraiman. Cluster analysis: a further approach based on density estimation. *Computational Statistics and Data Analysis*, 36(4):441–459, 2001.

[5] M. Ester, H.P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231, 1996.

[6] C. Fraley and A. Raftery. How many clusters? which clustering method? - answers via model-based cluster analysis. *The Computer Journal*, 41:578–588, 1998.

[7] C. Fraley and A. Raftery. Mclust: Software for model-based cluster analysis. *Journal of Classification*, 16:297–306, 1999.

[8] J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, 82:249–266, 1987.

[9] J. H. Friedman, W. Stuetzle, and Anne Schroeder. Projection pursuit density estimation. *Journal of the American Statistical Association*, 79:599–608, 1984.

[10] J.A. Hartigan. *Clustering Algorithms*. Wiley, 1975.

[11] J.A. Hartigan. Consistency of single linkage for high-density clusters. *Journal of the American Statistical Association*, 76:388–394, 1981.

[12] J.A. Hartigan. Estimation of a convex density contour in two dimensions. *Journal of the American Statistical Association*, 82(397):267–270, 1987.

[13] J.A. Hartigan and S. Mohanty. The runt test for multimodality. *Journal of Classification*, 9:63–70, 1992.

[14] L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–217, 1985.

[15] J. Klemelae. Visualization of multivariate density estimates with level set trees. *Journal of Computational & Graphical Statistics*, 13(3):599–620, 2004.

[16] J. Klemelae. Algorithms for manipulation of level sets of nonparametric density estimates. *Computational Statistics*, 20(2):349–368, 2005.

[17] G.J. McLachlan and D.Peel. *Finite Mixture Models.* Wiley, 2000.

[18] D.W. Mueller and G. Sawitzki. Excess mass estimates and tests of multimodality. *Journal of the American Statistical Association*, 86:738–746, 1991.

[19] A. Murua, L. Stanberry, and W. Stuetzle. On potts model clustering, kernel k-means, and density estimation. 2007.

[20] R. Nugent. *Algorithms for estimating the cluster tree of a density.* PhD thesis, University of Washington, 2006.

[21] W. Polonik. Measuring mass concentrations and estimating density contour clusters - an excess mass approach. *The Annals of Statistics*, 23(3):855–881, 1995.

[22] B.W. Silverman. *Density estimation for Statistics and Data Analysis.* Chapman & Hall, 1986.

[23] W. Stuetzle. Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. *Journal of Classification*, 20(5):25–47, 2003.

[24] P.A. Tukey and J.W. Tukey. Data driven view selection, agglomeration, and sharpening. In V. Barnett, editor, *Interpreting Multivariate Data*, pages 215–243. Wiley, 1981.

[25] G. Walther. Granulometric smoothing. *Annals of Statistics*, 25(6):2273–2299, 1997.

[26] D. Wishart. Mode analysis: A generalization of nearest neighbor which reduces chaining effects. In A.J. Cole, editor, *Numerical Taxonomy*, pages 282–311. Academic Press, 1969.

[27] M.A. Wong. *Hybrid Clustering.* PhD thesis, Yale University, 1979.

[28] M.A. Wong and T. Lane. A kth nearest neighbor clustering procedure. *Journal of the Royal Statistical Society, Series B*, 45:362–368, 1983.

[29] E.J. Yeoh, M.E. Ross, S.A. Shurtle, W.K. Williams, D. Patel, R. Mahfouz, F.G. Behm, S.C. Raimondi, M.V. Relling, A. Patel, C. Cheng, D. Campana, D. Wilkins, X. Zhou, J. Li, H. Liu, C.H. Pui, W.E. Evans, C. Naeve, L. Wong, and J.R. Downing. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1:133 – 143, 2002.