A Generalized Single Linkage Method for Estimating the Cluster Tree of a Density

Werner Stuetzle & Rebecca Nugent

A Generalized Single Linkage Method for Estimating the Cluster Tree of a Density

Werner STUETZLE and Rebecca NUGENT

The goal of clustering is to detect the presence of distinct groups in a dataset 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 algorithmic connections to single linkage clustering and illustrate our approach on several examples. Supplemental materials for the article, including an R package implementing generalized single linkage clustering, all datasets used in the examples, and R code producing the figures and numerical results, are available online.

Key Words: Cluster analysis; Excess mass; Level set; Nearest-neighbor density estimation; Runt test; Single linkage clustering.

1. INTRODUCTION

The goal of clustering is to identify distinct groups in a dataset 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 con-

Werner Stuetzle is Professor, Department of Statistics, University of Washington, Seattle, WA 98195-4322 (E-mail: *wxs@u.washington.edu*). Rebecca Nugent is Visiting Assistant Professor, Department of Statistics, Carnegie Mellon University, Pittsburgh, PA 15213 (E-mail: *rnugent@stat.cmu.edu*).
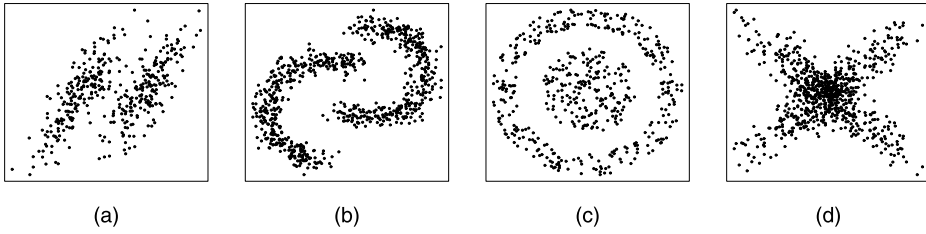
Figure 1.   (a)–(c) Distinct groups in the sense of Carmichael, George, and Julius; (d) groups that would not be considered distinct.

tiguous, relatively empty regions. This definition, although quite general, admittedly is not all-encompassing: Figure 1(a)–(c), would be regarded as showing two groups, whereas Figure 1(d) would not.

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, 2006; 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's rule.

In contrast, the nonparametric approach adopted in this article 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 Wishart's (1969) article 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, sec. 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 $k = 2$
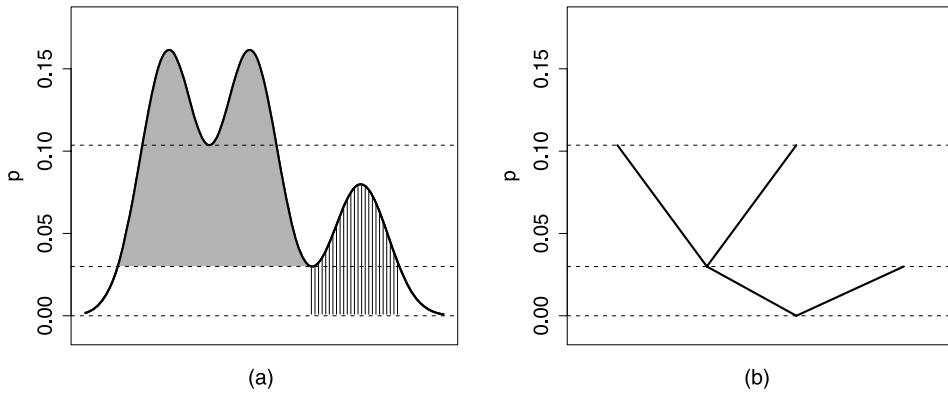
Figure 2.   (a) Density; (b) cluster tree. The shaded area represents the excess mass of the left daughter of the root node. The hashed area represents the size of the right daughter of the root node.

(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 referred to the connected components of all level sets as high density clusters, whereas 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.

We regard estimating the cluster tree as the fundamental goal of nonparametric cluster analysis. In this article 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 article.

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, chap. 4) and set aside all observations with $\hat{p}(\mathbf{x}_i) \leq \lambda$, that is, 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 suggested 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 Tukey and Tukey (1981).

A reincarnation of one-level mode analysis is the DBSCAN algorithm of Ester et al. (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 method 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) estimate $L(\lambda; p)$ by a union of balls centered at those observations with $\hat{p}(\mathbf{x}_i) > \lambda$. Estimating a level set by a union of balls centered at high density observations is also the basic idea of Cuevas, Febrero, and Fraiman (2000, 2001). Their estimate differs from Walther's estimate in the way the radii of the balls are chosen. The clusters produced by their method are the connected components of the estimated level set or, more precisely, the corresponding subsets of the observations. (This short summary does not do justice to the work reported by Cuevas, Febrero, and Fraiman (2000, 2001); their articles contain a number of interesting ideas, e.g., on using the Bootstrap to more accurately estimate level sets and assess the variability of the estimates.)

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 article 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 previously. 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 et al. (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.

Stuetzle (2003) suggested a new method—*runt pruning*—for extracting clusters from the cluster tree of the nearest-neighbor density estimate (which is isomorphic to the single linkage dendogram) by pruning branches believed to correspond to spurious modes. For a generalization of single linkage clustering to the kth nearest-neighbor density estimate see Wong (1979) and Wong and Lane (1983).

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 defined a "level set tree" which is different from the cluster tree in that it has nodes at every one of the (finitely many) levels occurring as values of $\hat{p}$.

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 (Nugent 2006). Such estimates, however, are only viable in low dimensions because binning high-dimensional space is not practical: ten bins per variable in ten dimensions would result in 10^{10} bins. 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; it is not even clear how one would represent level sets on the computer. 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 $\mathbf{x}(t): [0, 1] \rightarrow R^m$ 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 $\hat{p}$. Following Cuevas, Febrero, and Fraiman (2000, 2001), 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 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 $T(\lambda)$.*

Proof: 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. $\square$

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 to generate the examples presented in the article.

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 (e.g., 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 was described by Nugent (2006). In the examples we have tried, grid search, and exact computation produce very similar results.

We now illustrate the cluster tree algorithm on a simple two-dimensional example. Figure 3(a) shows a dataset 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 endpoints, drawn in gray. Therefore, thresholding eliminates this edge and the endpoint, 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 gray. Again, the minimum of the density along the edge is assumed at an endpoint, 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
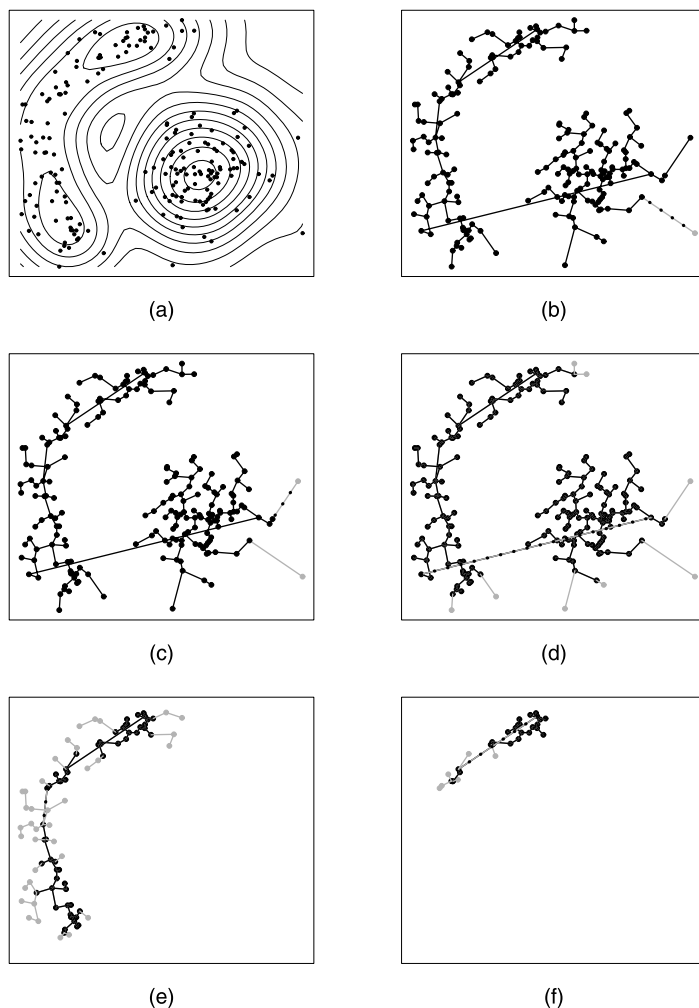
Figure 3.   (a) Dataset and isopleths of kernel density estimate; (b) maximal spanning tree of *G* with "shortest" edge dashed; (c) maximal spanning tree with second shortest edge dashed; (d) first split resulting in two connected components; (e) first split of banana; (f) second split of banana.

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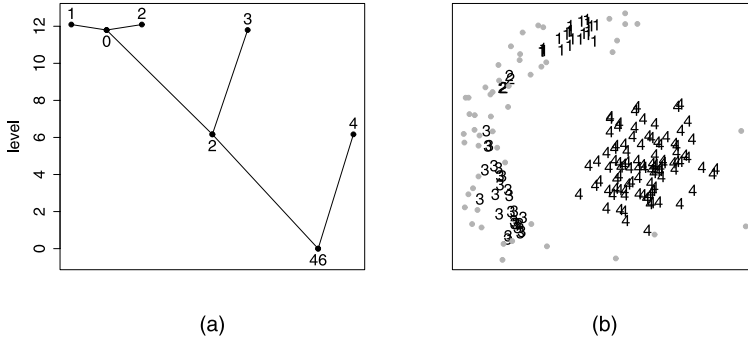
Figure 4. (a) Graph cluster tree; (b) clusters corresponding to leaves of graph cluster tree. Numbers above leaves are labels, numbers below interior nodes are runt excess masses.

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 gray 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 counterintuitive 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, used in the examples, 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

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

In Figure 2(a) the excess mass associated with the left daughter of the root node is represented by the shaded area. The concept of "excess mass" is due to Hartigan (1987); for an approach to clustering based on excess mass see also the works of Mueller and Sawitzki (1991) and Polonik (1995).

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

$$\int_{D(N)} d\mathbf{x} = \int_{D(N)} \frac{1}{p(\mathbf{x})} p(\mathbf{x}) \, d\mathbf{x}$$

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

and therefore

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

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.

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 internal 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 right 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 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 $d_1(\mathbf{x}, \mathcal{X}) = \min_i d(\mathbf{x}, \mathbf{x}_i)$. Admittedly, $\hat{p}^{(1)}$ 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 nontechnical 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

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

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

Proposition 2.   *The graph cluster tree of G is isomorphic to the single linkage dendo-gram.*

A proof of Proposition 2 is given in the Appendix (available online). Proposition 2 shows that generalized single linkage clustering applied to the nearest-neighbor density estimate reduces to single linkage clustering; the name is therefore justified.

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 equiva-lent 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 Sec-tion 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

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

reduces to the fraction of observations in the graph high density cluster $\tilde{D}(N)$, that is, 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.

7. EXAMPLES

The goal of this section is to illustrate generalized single linkage clustering on some examples and to compare it to standard hierarchical clustering methods (single linkage, average linkage, complete linkage, and Ward's method), and to parametric (model-based) clustering with Gaussian mixtures (Fraley and Raftery 1998, 1999, 2006; McLachlan and Peel 2000). (For hierarchical clustering we used the functions "hclust" and "cutree" in R version 2.7.1 (R Development Core Team 2008). For model-based clustering we used the function "Mclust" in the R package mclust version 3.1-5 (Fraley and Raftery 2008).)

Density estimation: Where feasible we present the results of generalized single link-age clustering for two different density estimates: the nearest-neighbor density estimate and a kernel density estimate with bandwidth determined by least squares cross-validation (Silverman 1986). We refer to these two versions as GSL-NN and GSL-K, respectively. 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 Euclid-ean 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 alternative because they are well understood and easy to implement, and least squares cross-validation offers a simple way for automatic bandwidth selection. We use an elliptical Gaussian kernel with the same covariance as the data. For some of the examples with high-dimensional feature space, kernel density estimation is not viable because least squares cross-validation fails or the kernel is singular. For these examples we only give results for the nearest-neighbor density estimate.

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.

In practice it is not necessary to calculate the edge weights for all the edges of the complete graph: edges connecting observations with large Euclidean distance will typically not end up in the maximal spanning tree. In the examples we only calculate the edge weights for a sparse test graph with edges that are either in the Euclidean minimal spanning tree of the observations or in the 20-nearest-neighbor graph. This shortcut results in a large speedup of the edge weight calculation: Evaluating a kernel density estimate at a single point takes work on the order $O(n)$, and therefore computing all the edge weights is $O(n^3)$. The shortcut reduces this to $O(n^2)$ while producing essentially the same clustering results.

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, ... 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. 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 get an alternative estimate for the number of modes 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 ith 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

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

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 renumbering 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   NONELLIPTICAL CLUSTERS—THE BULLSEYE DATA

The data in this example consist of a two-dimensional Gaussian cloud forming the center of the bullseye and a second group in the shape of a ring around the center. Figure 5(a)–(f), shows the 2-partitions generated by single linkage, average linkage, complete linkage, Ward's method, model-based clustering, and generalized single clustering with a kernel density estimate, respectively.
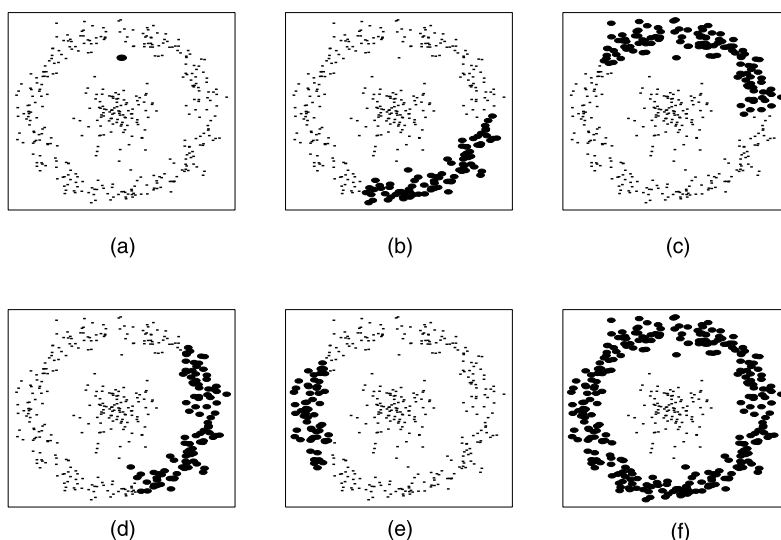

Figure 5.   (a)–(f): 2-partitions found by (a) single linkage, (b) average linkage, (c) complete linkage, (d) Ward's method, (e) model-based clustering, (f) generalized single linkage clustering.

Generalized single linkage clustering is the only method that recovers the obvious group structure. Single linkage clustering partitions the data into a cluster consisting of a single outlier and a second cluster made up of the rest. All the other methods fail in similar ways. This failure is not surprising, as they are all designed to find elliptical or roughly convex clusters.

7.2 GAUSSIAN CLUSTERS—THE SIMPLEX DATA

The data in this example consist of spherical Gaussian clusters with common standard deviation $\sigma = 0.25$, centered at the vertices of the unit simplex in seven dimensions. The cluster sizes are 50, 60, ..., 110. The runt sizes for the nearest-neighbor density estimate are, in descending order, 80, 73, 60, 38, 35, 26, 14, 10, 9, 9, 8, 7, 6, 6, ... The gap between 14 and 26 suggests seven clusters, which is the correct number. The adjusted Rand index is 0.83.

The runt sizes for the kernel density estimate are 118, 61, 41, 40, 38, 24, 5, 4, 3, ..., and the runt excess masses (rounded) are 34, 18, 10, 10, 9, 4, 1, 1, 1, ..., both suggesting seven clusters. The adjusted Rand index is 0.95.

The unpruned graph cluster tree of the corresponding kernel density estimate has 210 leaves, suggesting that the density estimate has 210 modes. To obtain an alternative estimate for the number of modes we start a numerical optimizer at each of the 560 observations and apply Ward's method to the local optima. The loss for the first 324 merges stays below 4×10^{-3} and then abruptly jumps to 3.8×10^{-1}, suggesting that there are at least $560 - 324 = 236$ distinct local optima. We conclude that the kernel estimate indeed has more than 200 modes.

Table 1 summarizes the performance of single linkage, average linkage, complete linkage, Ward's method, model-based clustering, generalized single linkage with nearest-neighbor density estimate, and generalized single linkage with kernel density estimate. The first row of the table contains the values of the adjusted Rand index when the methods are asked to construct a 7-partition. The second row contains the optimal values of the index (optimized over partition size). Numbers in parentheses are standard errors obtained by half-sampling (Shao and Tu 1995, sec. 5.2.2). All methods except single linkage perform well, although generalized single linkage with the nearest-neighbor estimate falls off a bit.

Table 1. Comparison of single linkage, average linkage, complete linkage, Ward's method, model-based clustering, generalized single linkage with nearest-neighbor density estimate, and generalized single linkage with kernel density estimate for the seven-dimensional Simplex data. First row: adjusted Rand index if methods are made to generate seven clusters; second row: adjusted Rand index for optimal partition size; third row: optimal partition size. Numbers in parentheses are standard errors.

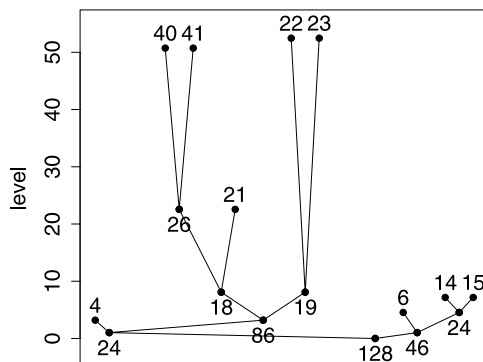| SL	AL	CL	WA	MC	GSL-NN	GSL-K
0.0 (0.02)	0.93 (0.03)	0.89 (0.04)	0.90 (0.03)	0.97 (0.01)	0.83 (0.07)	0.94 (0.09)
0.0 (0.03)	0.93 (0.03)	0.91 (0.03)	0.90 (0.03)	0.97 (0.01)	0.90 (0.06)	0.94 (0.07)
7	8	8	7	7	8	7

Figure 6.   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.

7.3   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.

We show the results for the kernel estimate. The unpruned graph cluster tree has 514 leaves. The alternative estimate for the number of modes (obtained by clustering local optima) is 501.

The runt excess masses are 128, 86, 46, 26, 24, 24, 19, 18, 11, 9, 8, 7, 7, 6, 6, . . . , and the runt sizes are 129, 89, 47, 33, 25, 24, 24, 20, 11, 9, 9, 8, . . . , both suggesting a nine-cluster solution. Figure 6 shows the (pruned) graph cluster tree. Table 2 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 40 and 41); and areas A7 and A8 are not cleanly separated. The adjusted Rand index is 0.61, reflecting the erroneous split of area A3.

Table 2.   Olive Oil data: leaf code (horizontal axis) tabulated against area (vertical axis).

	4	41	40	21	22	23	15	14	6
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 3. Comparison of single linkage, average linkage, complete linkage, Ward's method, model-based cluster-
ing, generalized single linkage with nearest-neighbor density estimate, and generalized single linkage
with kernel density estimate for the Olive Oil data. First row: adjusted Rand index if methods are made
to generate nine clusters; second row: adjusted Rand index for optimal partition size; third row: optimal
partition size. Numbers in parentheses are standard errors.

SL	AL	CL	WA	MC	GSL-NN	GSL-K
0.00 (0.06)	0.61 (0.07)	0.43 (0.08)	0.50 (0.05)	0.56 (0.12)	0.51 (0.05)	0.64 (0.1)
0.0 (0.1)	0.64 (0.06)	0.50 (0.07)	0.62 (0.06)	0.64 (0.07)	0.70 (0.08)	0.64 (0.11)
4	12	7	8	8	5	9

The runt sizes for the nearest-neighbor estimate are 168, 97, 59, 51, 42, 42, 33, 13, 13, 12, 11, 11, 11, 10, ..., suggesting eight clusters. The adjusted Rand index is 0.57.

Table 3 summarizes the performance of single linkage, average linkage, complete link-age, Ward's method, model-based clustering, generalized single linkage with nearest-neighbor density estimate, and generalized single linkage with kernel density estimate. Single linkage fails, and complete linkage is inferior to the best methods. Performance of the other methods is comparable.

7.4 THE HANDWRITTEN DIGIT DATA

The data for this example are 2000 16×16 gray level images of handwritten digits; the data therefore are of dimensionality 256. (The data were previously used to evaluate machine learning algorithms.) Least squares cross-validation failed for this dataset, and we are reporting the results of generalized single linkage clustering only for the nearest-neighbor density estimate. The runt sizes are 288, 283, 90, 84, 74, 47, 37, 35, 22, 21, 21, 19, 19, 18, 13, 12, 12, ... The gap after 35 (vaguely) suggests nine clusters. The corresponding adjusted Rand index is 0.64.

Table 4 summarizes the performance of single linkage, average linkage, complete link-age, Ward's method, model-based clustering, and generalized single linkage with nearest-neighbor density estimate. Ward's method and generalized single linkage perform best; the poor showing of average linkage and model-based clustering is surprising.

Table 4. Comparison of single linkage, average linkage, complete linkage, Ward's method, model-based clus-
tering, generalized single linkage with nearest-neighbor density estimate, for the Handwritten Digit
data. First row: adjusted Rand index if methods are made to generate ten clusters; second row: adjusted
Rand index for optimal partition size; third row: optimal partition size. Numbers in parentheses are
standard errors.

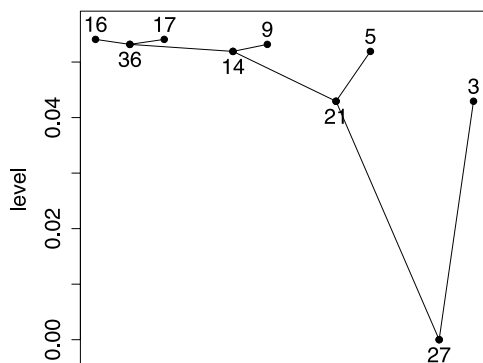| SL	AL	CL	WA	MC	GSL-NN
0.00 (0.00)	0.07 (0.07)	0.28 (0.07)	0.65 (0.05)	0.35 (0.03)	0.54 (0.05)
0.0 (0.00)	0.25 (0.08)	0.35 (0.06)	0.65 (0.03)	0.38 (0.04)	0.64 (0.05)
12	15	13	10	8	7

Figure 7.   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.

7.5 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 datasets. The Acute Lymphoblastic Leukemia (ALL) data are oligonucleotide microarray gene expression levels of 12,558 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, TEL-AML1, 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 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. Kernel density estimation does not make sense in this example, as the observations lie in a 214-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 7 shows the (pruned) graph cluster tree, and Table 5 shows a table of ALL subtype (vertical axis) against leaf code (horizontal axis). The T-ALL, E2A-PBX1, and TEL-AML1 subtypes

Table 5.   ALL data: leaf code (horizontal axis) tabulated against ALL subtype (vertical axis).

	3	5	9	16	17
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 6. Comparison of single linkage, average linkage, complete linkage, Ward's method, model-based clustering, generalized single linkage with nearest-neighbor density estimate, for the Acute Lymphoblastic Leukemia data. First row: adjusted Rand index if methods are made to generate seven clusters; second row: adjusted Rand index for optimal partition size; third row: optimal partition size. Numbers in parentheses are standard errors.

SL	AL	CL	WA	MC	GSL-NN
0.12 (0.03)	0.12 (0.09)	0.30 (0.11)	0.60 (0.08)	0.64 (0.08)	0.54 (0.07)
0.12 (0.03)	0.29 (0.16)	0.48 (0.08)	0.63 (0.07)	0.64 (0.06)	0.55 (0.05)
13	12	15	6	8	5

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 (2008) 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.

Table 6 summarizes the performance of single linkage, average linkage, complete linkage, Ward's method, model-based clustering, and generalized single linkage with nearest-neighbor density estimate. Single, average, and complete linkage are not competitive. Ward's method, model-based clustering, and generalized single linkage show comparable perfomance.

8. SUMMARY AND DISCUSSION

Nonparametric clustering is based on the premise that groups in the data 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 (essentially) the single linkage dendogram. Excess mass pruning is a generalization of the runt pruning method for extracting clusters from a 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. Single linkage clustering—extracting clusters from the single linkage dendogram by dendogram cutting—is not a viable clustering method. Pruning branches with small size is vastly superior.

2. The other standard hierarchical clustering methods (average and complete linkage, and Ward's method) and model-based clustering cannot find clusters that are highly nonconvex (bullseye), whereas generalized single linkage can.

3. In the (admittedly small number of) examples, generalized single linkage clustering was consistently among the top performers. This suggests that the performance penalty paid for the flexibility of generalized single linkage clustering (the ability to cope with nonconvex clusters) is small.

4. Kernel density estimates with bandwidth 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. Runt size and runt excess mass provide good indicators for the number of groups.

5. Kernel and nearest-neighbor density estimates give comparable clustering performance in most examples. This came as a pleasant surprise because the calculations for the nearest-neighbor estimate are very fast: clustering the Handwritten Digit data with 2000 observations and 256 variables takes less than 10 sec on a modern PC.

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, that is, 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 be preferrable. The work of Nugent (2006) contains some preliminary ideas and results.

SUPPLEMENTAL MATERIALS

Data and results: A zip archive containing an R package implementing generalized single linkage clustering as well as the Latex source, R code, and data needed to reproduce

article, Appendix, and all the figures and numerical results presented therein is available online. The file "README.txt" in the archive has a detailed description of the archive. (Stuetzle-Nugent-supp.zip)

ACKNOWLEDGMENTS

REFERENCES

Ankerst, M., Breuning, M. M., Kriegel, H. P., and Sander, J. (1999), "Optics: Ordering Points to Identify the Clustering Structure," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD'99)*, Philadelphia, PA: ACM Press, pp. 49–60. [400]

Carmichael, J. W., George, G. A., and Julius, R. S. (1968), "Finding Natural Clusters," *Systematic Zoology*, 17, 144–150. [397]

Cuevas, A., Febrero, M., and Fraiman, R. (2000), "Estimating the Number of Clusters," *Canadian Journal of Statistics*, 28 (2), 367–382. [400,402]

——— (2001), "Cluster Analysis: A Further Approach Based on Density Estimation," *Computational Statistics and Data Analysis*, 36 (4), 441–459. [400,402]

Ester, M., Kriegel, H. P., Sander, J., and Xu, X. (1996), "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases With Noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, AAAI Press, pp. 226–231. [400]

Fraley, C., and Raftery, A. (1998), "How Many Clusters? Which Clustering Method?—Answers via Model-Based Cluster Analysis," *The Computer Journal*, 41, 578–588. [398,408]

——— (1999), "Mclust: Software for Model-Based Cluster Analysis," *Journal of Classification*, 16, 297–306. [398,408]

——— (2006), "Mclust Version 3 for R: Normal Mixture Modeling and Model-Based Clustering," Technical Report 504, University of Washington, Dept. of Statistics. [398,408]

——— (2008), "mclust: Model-Based Clustering/Normal Mixture Modeling," R package version 3.1-5. [408]

Friedman, J. H. (1987), "Exploratory Projection Pursuit," *Journal of the American Statistical Association*, 82, 249–266. [416]

Friedman, J. H., Stuetzle, W., and Schroeder, A. (1984), "Projection Pursuit Density Estimation," *Journal of the American Statistical Association*, 79, 599–608. [416]

Hartigan, J. A. (1975), *Clustering Algorithms*, New York: Wiley. [398,399]

——— (1981), "Consistency of Single Linkage for High-Density Clusters," *Journal of the American Statistical Association*, 76, 388–394. [398]

——— (1987), "Estimation of a Convex Density Contour in Two Dimensions," *Journal of the American Statistical Association*, 82 (397), 267–270. [406]

Hartigan, J. A., and Mohanty, S. (1992), "The Runt Test for Multimodality," *Journal of Classification*, 9, 63–70. [407]

Hubert, L., and Arabie, P. (1985), "Comparing Partitions," *Journal of Classification*, 2, 193–217. [410]

Klemelae, J. (2004), "Visualization of Multivariate Density Estimates With Level Set Trees," *Journal of Computational and Graphical Statistics*, 13 (3), 599–620. [401]

——— (2005), "Algorithms for Manipulation of Level Sets of Nonparametric Density Estimates," *Computational Statistics*, 20 (2), 349–368. [401]

McLachlan, G. J., and Peel, D. (2000), *Finite Mixture Models*, New York: Wiley. [398,408]

Mueller, D. W., and Sawitzki, G. (1991), "Excess Mass Estimates and Tests of Multimodality," *Journal of the American Statistical Association*, 86, 738–746. [406]

Murua, A., Stanberry, L., and Stuetzle, W. (2008), "On Potts Model Clustering, Kernel *k*-Means and Density Estimation," *Journal of Computational and Graphical Statistics*, 17 (4), 629–658. [415]

Nugent, R. (2006), "Algorithms for Estimating the Cluster Tree of a Density," Ph.D. thesis, University of Washington. [401,403,416]

Polonik, W. (1995), "Measuring Mass Concentrations and Estimating Density Contour Clusters—An Excess Mass Approach," *The Annals of Statistics*, 23 (3), 855–881. [406]

R Development Core Team (2008), *R: A Language and Environment for Statistical Computing*, Vienna, Austria: R Foundation for Statistical Computing. [408]

Shao, J., and Tu, D. (1995), *The Jackknife and Bootstrap*, New York: Springer. [411]

Silverman, B. W. (1986), *Density Estimation for Statistics and Data Analysis*, London–New York: Chapman & Hall. [400,408]

Stuetzle, W. (2003), "Estimating the Cluster Tree of a Density by Analyzing the Minimal Spanning Tree of a Sample," *Journal of Classification*, 20 (5), 25–47. [398,401,408,415]

Tukey, P. A., and Tukey, J. W. (1981), "Data Driven View Selection, Agglomeration, and Sharpening," in *Interpreting Multivariate Data*, ed. V. Barnett, Chichester–New York: Wiley, pp. 215–243. [400]

Walther, G. (1997), "Granulometric Smoothing," *The Annals of Statistics*, 25 (6), 2273–2299. [400]

Wishart, D. (1969), "Mode Analysis: A Generalization of Nearest Neighbor Which Reduces Chaining Effects," in *Numerical Taxonomy*, ed. A. J. Cole, London–New York: Academic Press, pp. 282–311. [398-400]

Wong, M. A. (1979), "Hybrid Clustering," Ph.D. thesis, Yale University. [401]

Wong, M. A., and Lane, T. (1983), "A *k*th Nearest Neighbor Clustering Procedure," *Journal of the Royal Statistical Society, Ser. B*, 45, 362–368. [401]

Yeoh, E. J., Ross, M. E., Shurtle, S. A., Williams, W. K., Patel, D., Mahfouz, R., Behm, F. G., Raimondi, S. C., Relling, M. V., Patel, A., Cheng, C., Campana, D., Wilkins, D., Zhou, X., Li, J., Liu, H., Pui, C. H., Evans, W. E., Naeve, C., Wong, L., and Downing, J. R. (2002), "Classification, Subtype Discovery, and Prediction of Outcome in Pediatric Acute Lymphoblastic Leukemia by Gene Expression Profiling," *Cancer Cell*, 1, 133–143. [414]