# Clustering with Confidence: A Binning Approach

**Rebecca Nugent**
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213

**Werner Stuetzle**
Department of Statistics
University of Washington
Box 354322, Seattle, WA 98195

## Abstract

We present a plug-in method for estimating the cluster tree of a density. The method takes advantage of the ability to exactly compute the level sets of a piecewise constant density estimate. We then introduce *clustering with confidence*, an automatic pruning procedure that assesses significance of splits (and thereby clusters) in the cluster tree; the only user input required is the desired confidence level.

## 1 Introduction

The goal of clustering is to identify distinct groups in a data set and assign a group label to each observation. Ideally, we would be able to find the number of groups as well as where each group lies in the feature space with minimal input from the user. For example, Figure 1 below shows two curvilinear groups, 200 observations each, and two spherical groups, 100 observations each; we would like a four cluster solution, one per group.

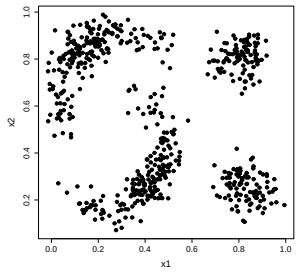



Figure 1: Data set with four well-separated groups

To cast clustering as a statistical problem, we regard the data, $\mathcal{X} = \{\mathbf{x_1}, ..., \mathbf{x_n}\} \in \Re^{\mathbf{P}}$, as a sample from some unknown population density $p(\mathbf{x})$. There are two statistical approaches. Parametric (model-based) clustering assumes the data have been generated by a finite mixture of $g$ underlying parametric probability distributions $p_g(\mathbf{x})$ (often multivariate Gaussian), one component per group (Fraley, Raftery 1998; McLachlan, Basford 1988). This procedure selects a number of mixture components (clusters) and estimates their parameters. However, it is susceptible to violation of distributional assumptions. Skewed or non-Gaussian groups may be modeled incorrectly by a mixture of several Gaussians.

In contrast, the nonparametric approach assumes a correspondence between groups in the data and modes of the density $p(\mathbf{x})$. Wishart (1969) first advocated searching for modes as manifestations of the presence of groups; nonparametric clustering algorithms should be able to "resolve distinct data modes, independently of their shape and variance".

Hartigan expanded this idea and made it more precise (1975, 1981, 1985). Define a level set $L(\lambda; p)$ of a density $p$ at level $\lambda$ as the subset of the feature space for which the density exceeds $\lambda$:

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

Its connected components are the maximally connected subsets of a level set. For any two connected components $A$ and $B$, possibly at different levels, either $A \subset B$, $B \subset A$, or $A \cap B = \emptyset$. This hierarchical structure of the level sets is summarized by the *cluster tree* of $p$.

The cluster tree is easiest to define recursively (Stuetzle 2003). Each node $N$ of the tree represents a subset $D(N)$ of the support $L(0; p)$ of $p$ and is associated with a density level $\lambda(N)$. The root node represents the entire support of $p$ and is associated with density level $\lambda(N) = 0$. To determine the daughters of a node, we find the lowest level $\lambda_d$ for which $L(\lambda_d; p) \bigcap D(N)$ has two or more connected components. If no such $\lambda_d$ exists, then $D(N)$ is a mode of the density, and $N$ is a

leaf of the tree. Otherwise, let $C_1, C_2, ..., C_n$ be the connected components of $L(\lambda_d; p) \bigcap D(N)$. If $n = 2$, we create two daughter nodes at level $\lambda_d$, one for each connected component; we then apply the procedure recursively to each daughter node. If $n > 2$, we create two connected components $C_1$ and $C_2 \bigcup C_3 ... \bigcup C_n$ and their respective daughter nodes and then recurse. We call the regions $D(N)$ the "high density clusters" of $p$.

Note that we have defined a binary tree; other possible definitions with non-binary splits could be used. However, the recursive binary tree easily accommodates level sets with more than two connected components. For example, if the level set $L(\lambda_d; p) \bigcap D(N)$ has three connected components, $C_1, C_2, C_3$, the binary tree first splits, say, $C_1$ from $C_2 \bigcup C_3$ at height $\lambda_d$ and, after finding any subsequent splits and leaves on the corresponding branch for $C_1$, returns to the connected component $C_2 \bigcup C_3$ at height $\lambda_d$, immediately splits $C_2$ and $C_3$, and creates two more daughter nodes at height $\lambda_d$. The final cluster tree structure will contain a node for each of the three connected components, all at height $\lambda_d$. Figure 2 shows a univariate density with four modes and the corresponding cluster tree with initial split at $\lambda = 0.0044$ and subsequent splits at $\lambda = 0.0288, 0.0434$. Estimating the cluster tree is a fundamental goal of nonparametric cluster analysis.

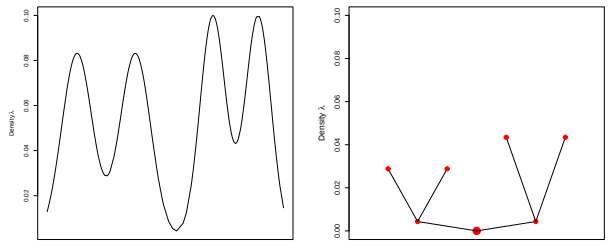


Figure 2: (a) Density with four modes; (b) corresponding cluster tree with three splits (four leaves)

There are several previously suggested clustering methods based on level sets. Wishart's *one level mode analysis* is probably the first such method (1969); its goal is to find the connected components of $L(\lambda; p)$ for a given $\lambda$. After computing a density estimate $\hat{p}$, all observations $\mathbf{x_i}$ where $\hat{p}(\mathbf{x_i}) \leq \lambda$ are set aside. Well-separated groups of the remaining high density observations correspond to connected components of $L(\lambda; p)$. One drawback is that not all groups or modes may be identified by examining a single level set. Wishart's *hierarchical mode analysis* addresses this weakness; this complex algorithm constructs a cluster tree (although not so named) through iterative merging.

Several other level set estimation procedures have been proposed. Walther's level set estimator consists of two steps (1997). First, compute a density estimate $\hat{p}$ of the underlying density $p$. Second, approximate $L(\lambda; \hat{p})$ by $L^*(\lambda; \hat{p})$, a union of balls $B_i$ of radius r centered around $\mathbf{x_i}$ with $\hat{p}(\mathbf{x_i}) > \lambda$ that do not contain $\mathbf{x_i}$ with $\hat{p}(\mathbf{x_i}) \leq \lambda$. While this structure is computationally convenient for finding connected components, its accuracy depends on the choice of $r$. Walther provides a formula for $r$ based on the behavior of $p$ on the boundary $\partial L(\lambda; p)$ of the level set $L(\lambda; p)$. The Cuevas, Febrero, and Fraiman level set estimator also is based on a union of balls of radius $r$ (2000, 2001). However, their estimate $L^*(\lambda; \hat{p})$ is the union of all balls $B_i$ centered around $\mathbf{x_i}$ where $\hat{p}(\mathbf{x_i}) > \lambda$; no balls are excluded. They propose several heuristic methods for choosing $r$ based on interpoint distance.

Stuetzle's runt pruning method (2003) estimates the cluster tree of $p$ by computing the cluster tree of the nearest neighbor density estimate and then prunes branches corresponding to supposed spurious modes. The method is sensitive to the pruning amount parameter chosen by the user. Klemelä developed visualization tools for multivariate density estimates that are piecewise constant over (hyper-)rectangles (2004). He defines a "level set tree" which has nodes at every level corresponding to a unique value of $\hat{p}$. Note that the cluster tree only has nodes at split levels.

## 2 Constructing the cluster tree for a piecewise constant density estimate

We can estimate the cluster tree of a density $p$ by the cluster tree of a density estimate $\hat{p}$. However, for most density estimates, computing the cluster tree is a difficult problem; there is no obvious method for computing and representing the level sets. Exceptions are density estimates that are piecewise constant over (hyper-)rectangles. Let $B_1, B_2, ..., B_N$ be the rectangles, and let $\hat{p}_i$ be the estimated density for $B_i$. Then

$$L(\lambda; \hat{p}) = \bigcup_{\hat{p}_i > \lambda} B_i.$$

If the dimension is low enough, any density estimate can be reasonably binned. To illustrate, we choose the Binned Kernel Density Estimate (BKDE), a binned approximation to an ordinary kernel density estimate, on the finest grid we can computationally afford (Hall, Wand 1996; Wand, Jones 1995). We use ten-fold cross validation for bandwidth estimation. Figure 3a shows a heat map of the BKDE for the four well-separated groups on a 20 x 20 grid with a cross-validation bandwidth of 0.0244. Several high frequency areas are indicated.

Note that during cluster tree construction, $L(\lambda_d; \hat{p}) \bigcap D(N)$ changes structure only when the level $\lambda_d$ is equal to the next higher value of $\hat{p}(B_i)$ for one or more bins $B_i$ in $D(N)$. After identifying and sorting the bins' unique density estimate values, we can compute the cluster tree of $\hat{p}$ by "stepping through" only this subset of all possible levels $\lambda$. Every increase in level then corresponds to the removal of one or more bins from the level set $L(\lambda_d; \hat{p})$.

Finding the connected components of $L(\lambda_d; \hat{p}) \cap D(N)$ can be cast as a graph problem. Let $G$ be an adjacency graph over the bins $B_i \in L(\lambda_d; \hat{p}) \cap D(N)$. The vertices of the graph are the bins $B_i$. Two bins $B_i$ and $B_j$ are connected by an edge if adjacent (i.e. share a lower-dimensional face). The connected components of $G$ correspond to the connected components of the level set $L(\lambda_d; \hat{p}) \cap D(N)$ and finding them is a standard graph problem (Sedgewick 2002).

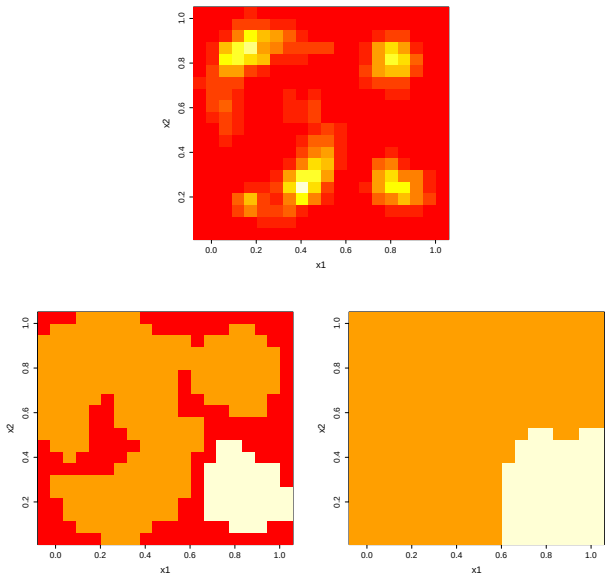



Figure 3: (a) example data BKDE on a 20 x 20 grid; (b) level set at $\lambda = 0.00016$; cluster cores in orange, white; (c) corresponding partitioned feature space

When a node $N$ of the cluster tree has been split into daughters $N_l$, $N_r$, the high density clusters $D(N_l)$, $D(N_r)$, also referred to as the cluster "cores", do not form a partition of $D(N)$. We refer to the bins in $D(N) \backslash (D(N_l) \cup D(N_r))$ (and observations in these bins) as the "fluff". We assign each bin $B$ in the fluff to $N_r$ if the Manhattan distance $d_M(B, D(N_r))$ is smaller that $d_M(B, D(N_l))$. If $d_M(B, D(N_r)) > d_M(B, D(N_l))$, then $B$ is assigned to $N_l$. In case of ties, the algorithm arbitrarily chooses an assignment. The cluster cores and fluff represented by the leaves of the cluster tree form a partition of the support of $\hat{p}$ and a corresponding partition of the observations. The same is true for every subtree of the cluster tree. Figure 3b shows $L(0.00016; \hat{p})$ with removed bins in red. Two daughter nodes are created, one for each connected component/cluster core. Figure 3c illustrates the subsequent partitioning of the feature space after assignment of fluff bins to the closest core.

Figure 4a shows the cluster tree generated from the example BKDE; the corresponding cluster assignments and partitioned feature space (labeled by color) are in Figures 4b,c. The cluster tree indicates the BKDE has nine modes. The first split at $\lambda = 1 \cdot 10^{-16}$ and the mode of $\hat{p}$ around (0.5, 0) (in yellow in Figure 4c) are artifacts of the density estimate; no observations are assigned to the resulting daughter node. The remaining eight leaves correspond to (a subset of) one of the groups.

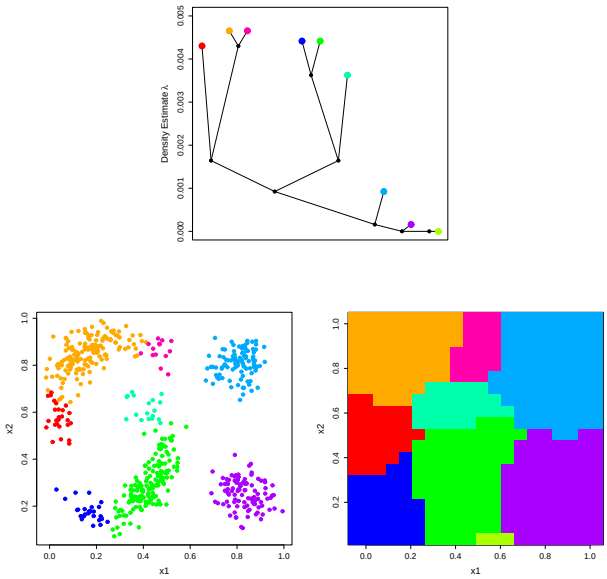



Figure 4: (a) Cluster tree for BKDE; (b) cluster assignments by color; (c) partitioned feature space

We compare these results to common clustering approaches (briefly described here). K-means is a iterative descent approach that minimizes the within-cluster sum of squares; it tends to partition the feature space into equal-sized (hyper-)spheres (Hartigan, Wong 1979). Although we know there are four groups, we range the number of clusters from two to twelve and plot the total within-cluster sum of squares against the number of clusters. The "elbow" in the curve occurs around five or six clusters. Solutions with more clusters have negligible decreases in the criterion. Model-based clustering is described in Section 1. We allow the procedure to search over one to twelve clusters with no constraints on the covariance matrix; a ten-cluster solution is chosen. Finally, we include results

from three hierarchical linkage methods (Mardia, et al 1979). Hierarchical agglomerative linkage methods iteratively join clusters in an order determined by a chosen between-cluster distance; we use the minimum, maximum, or average Euclidean distance between observations in different clusters (single, complete, average respectively). The hierarchical cluster structure is graphically represented by a tree-based dendrogram whose branches merge at heights corresponding to the clusters' dissimilarity. We prune each dendrogram for the known number of groups (four). In all cases, we compare the generated clustering partitions to the true groups using the Adjusted Rand Index (Table 1) (Hubert, Arabie 1985). The ARI is a common measure of agreement between two partitions. Its expected value is zero under random partitioning with a maximum value of one; larger values indicate better agreement.

Table 1: Comparison of Clustering Methods

| **Method** | **ARI** |
|---|---|
| Cluster Tree: 20 x 20 (k = 8) | 0.781 |
| Cluster Tree: 15 x 15 (k = 6) | 0.865 |
| K-means (k = 4) | 0.924 |
| K-means (k = 5) | 0.803 |
| K-means (k = 6) | 0.673 |
| MBC (k = 10) | 0.534 |
| Linkage (single, complete, average) | 1 |

The groups in the simulated data are well-separated; however, the two curvilinear groups lead to an expected increased number of clusters. While the four cluster k-means solution had high agreement with the true partition, the total within-cluster sum of squares criterion was lower for solutions with larger numbers of clusters (5, 6) who had lower ARIs comparable to the cluster tree partition generated for the BKDE on a 20 x 20 grid. Model-based clustering overestimated the number of groups in the population; the ARI correspondingly drops. All three linkage methods gave perfect agreement; this performance is not unexpected given the well-separated groups.

Although the cluster tree method has an ARI of 0.781, the algorithm generates the correct partition for the density estimate. Its performance is dependent on the density estimate. A similarly found density estimate on a 15 x 15 grid with a cross-validation selected bandwidth of 0.015 yielded a cluster tree with an ARI of 0.865. In both cases, the number of groups is overestimated (8 and 6). Figure 4 illustrates this problem in the approach. While the cluster tree is accurate for the given density estimate, a density estimate is inherently noisy, which results in spurious modes not corresponding to groups in the underlying population. In our example, the procedure identified the four original groups (the three splits post-artifact split in the cluster tree) but erroneously continued splitting the clusters. The corresponding branches of the cluster tree need to be pruned.

## 3 Clustering with Confidence

We propose a bootstrap-based automatic pruning procedure. The idea is to find $(1-\alpha)$ simultaneous upper and lower confidence sets for each level set. During cluster tree construction, only splits indicated as significant by the bootstrap confidence sets are taken to signal multi-modality. Spurious modes are discarded during estimation; the only decision from the user is the confidence level.

### 3.1 Bootstrap confidence sets for level sets

We choose upper confidence sets to be of the form $L^u(\lambda;\hat{p}) = L(\lambda - \delta^u_\lambda;\hat{p})$ and lower confidence sets of form $L^l(\lambda;\hat{p}) = L(\lambda + \delta^l_\lambda;\hat{p})$ with $\delta^u_\lambda,\ \delta^l_\lambda > 0$. By construction,

$$Lower = L^l(\lambda;\hat{p}) \subseteq L(\lambda;\hat{p}) \subseteq L^u(\lambda;\hat{p}) = Upper.$$

Let $\hat{p}^*_1, \hat{p}^*_2, ..., \hat{p}^*_m$ be the density estimates for $m$ bootstrap samples of size $n$ drawn with replacement from the original sample. We call a pair $(L(\lambda - \delta^u_\lambda;\hat{p}),\ L(\lambda + \delta^l_\lambda;\hat{p}))$ a non-simultaneous $(1-\alpha)$ confidence band for $L(\lambda;p)$ if for $(1-\alpha)$ of the bootstrap density estimates $\hat{p}^*_i$, the upper confidence set $L(\lambda-\delta^u_\lambda;\hat{p})$ contains $L(\lambda;\hat{p}^*_i)$ and the lower confidence set $L(\lambda+\delta^l_\lambda;\hat{p})$ is contained in $L(\lambda;\hat{p}^*_i)$:

$$P_{boot}\{L(\lambda+\delta^l_\lambda;\hat{p}) \subseteq L(\lambda;\hat{p}^*_i) \subseteq L(\lambda-\delta^u_\lambda;\hat{p})\} \geq 1-\alpha.$$

Here is one method to determine $\delta^u_\lambda, \delta^l_\lambda$ (Buja 2002). For each bootstrap sample $\hat{p}^*_i$ and each of the finitely many levels of $\hat{p}$, find the smallest $\delta^u_\lambda(i)$ such that

$$L(\lambda;\hat{p}^*_i) \subseteq L(\lambda - \delta^u_\lambda(i);\hat{p})$$

and the smallest $\delta^l_\lambda(i)$ such that

$$L(\lambda + \delta^l_\lambda(i);\hat{p}) \subseteq L(\lambda;\hat{p}^*_i).$$

Choose $\delta^u_\lambda = (1 - \frac{\alpha}{2})$ quantile of the $\delta^u_\lambda(i)$ and $\delta^l_\lambda = (1-\frac{\alpha}{2})$ quantile of the $\delta^l_\lambda(i)$. By construction, the pair $(L(\lambda-\delta^u_\lambda;\hat{p}), L(\lambda+\delta^l_\lambda;\hat{p}))$ is a $(1-\alpha)$ non-simultaneous confidence band for $L(\lambda;p)$. To get confidence bands for all $\lambda$ occurring as values of $\hat{p}$ with simultaneous coverage probability $1-\alpha$, we simply increase the coverage level of the individual bands until

$$P_{boot}\{L(\lambda+\delta^l_\lambda;\hat{p}) \subseteq L(\lambda;\hat{p}^*_i) \subseteq L(\lambda-\delta^u_\lambda;\hat{p})\ \forall\lambda\} \geq 1-\alpha.$$

Note that the actual upper and lower confidence sets for $L(\lambda; p)$ are the level sets $(L(\lambda - \delta_\lambda^u; \hat{p}), L(\lambda + \delta_\lambda^l; \hat{p}))$ respectively for $\hat{p}$. The bootstrap is used only to find $\delta_\lambda^u$, $\delta_\lambda^l$.

### 3.2 Constructing the cluster tree

After finding $\delta_\lambda^u$, $\delta_\lambda^l$ for all $\lambda$, we incorporate the bootstrap confidence sets into the cluster tree construction by only allowing splits at heights $\lambda$ for which the corresponding bootstrap confidence set $(L^l(\lambda; \hat{p}), L^u(\lambda; \hat{p}))$ gives strong evidence of a split. We use a similar recursive procedure as described in Section 2. The root node represents the entire support of $\hat{p}$ and is associated with density level $\lambda(N) = 0$. To determine the daughters of a node, we find the lowest level $\lambda_d$ for which a) $L^l(\lambda_d; \hat{p}) \bigcap D(N)$ has two or more connected components that b) are disconnected in $L^u(\lambda_d; \hat{p}) \bigcap D(N)$. Condition (a) indicates that the underlying density $p$ has two peaks above height $\lambda$; condition (b) indicates that the two peaks are separated by a valley dipping below height $\lambda$. Satisfying both conditions indicates a split at height $\lambda$. If no such $\lambda_d$ exists, $N$ is a leaf of the tree. Otherwise, let $C_1^l, C_2^l$ be two connected components of $L^l(\lambda_d; \hat{p}) \bigcap D(N)$ that are disconnected in $L^u(\lambda_d; \hat{p}) \bigcap D(N)$. Let $C_1^u$ and $C_2^u$ be the connected components of $L^u(\lambda_d; \hat{p}) \bigcap D(N)$ that contain $C_1^l$ and $C_2^l$ respectively. If $n = 2$, we create two daughter nodes at level $\lambda_d$ for $C_1^u$ and $C_2^u$ and, to each, apply the procedure recursively. If $n > 2$, we create two connected components $C_1^u$ and $C_2^u \bigcup C_3^u ... \bigcup C_n^u$ and daughter nodes and recurse.

Returning to the example cluster tree (Figure 4a), we examine the split at $\lambda = 0.0036$, the first split that breaks the lower left curvilinear group into two clusters. Figure 5 shows the bootstrap confidence set ($\alpha = 0.05$) for this level set. The original level set $L(0.0036; \hat{p})$ is in Figure 5a, color-coded by subsequent final leaf. The lower confidence set (Figure 5b) is found to be $\delta_\lambda^l = 0.0037$ higher, i.e. $L(0.0073; \hat{p})$. The upper confidence set (Figure 5c) is found to be $\delta_\lambda^u = 0.0026$ lower, i.e. $L(0.001; \hat{p})$. At $\lambda = 0.0036$, the upper confidence set does not have two connected components (no valley). Moreover, even though the lower confidence set does have two connected components, they do not correspond to the two connected components in $L(0.0036; \hat{p})$. We do not have evidence of a significant split and so do not create daughter nodes at this level.

Clustering with Confidence (CWC) using $\alpha = 0.05$ generates the cluster tree and data/feature space partitions seen in Figure 6. The cluster tree's significant splits have identified the four original groups (ARI = 1). No other smaller clusters (or modal artifacts) are found. Note that the split heights are higher than the

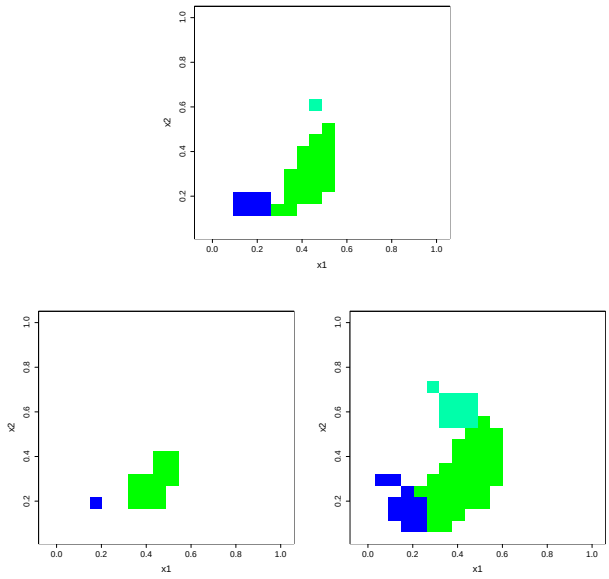



Figure 5: Bootstrap confidence set for $\lambda = 0.0036$: (a)$L(0.0036; \hat{p})$ ; (b) Lower confidence set, $\delta_\lambda^l = 0.0037$; (c) upper confidence set, $\delta_\lambda^u = 0.0026$

corresponding split heights in the cluster tree in Figure 4. The CWC procedure required stronger evidence for a split than was available at the lower levels. It performed similarly to hierarchical linkage methods and more favorably than k-means or model-based clustering (Table 1).

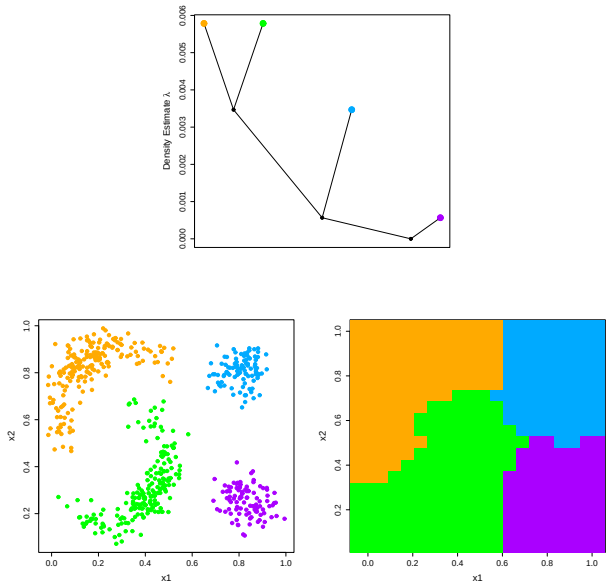



Figure 6: (a) 95% confidence cluster tree (b) cluster assignments; (c) partitioned feature space

# 4 Examples

The algorithms presented could be used for any number of dimensions, but are more tractable for lower dimensions. For easy visualization of the results, we present two examples using real data in two dimensions. We comment on higher dimensionality in the summary and future work section.

## 4.1 Identifying different skill sets in a student population

In educational research, a fundamental goal is identifying whether or not students have mastered the skills taught. The data available usually consists of whether or not a student answered questions correctly and which skills the questions require. Cognitive diagnosis models and item response theory models, often used to estimate students' abilities, can be lengthy in run time; estimation instability grows as the number of students, questions, and skills increases. Clustering capability estimates easily derived from students' responses and skill-labeled questions to partition students into groups with different skill sets has been proposed as a quick alternative (Ayers, Nugent, Dean 2008). A capability vector $B_i = \{B_{i1}, B_{i2}, ..., B_{iK}\} \in [0,1]^K$ for $K$ skills is found for each student. For skill $k$, a $B_{ik}$ near 0 indicates no mastery, near 1 indicates mastery, and near 0.5 indicates partial mastery or uncertainty. The cluster center is viewed as a skill set capability profile for a group of students.

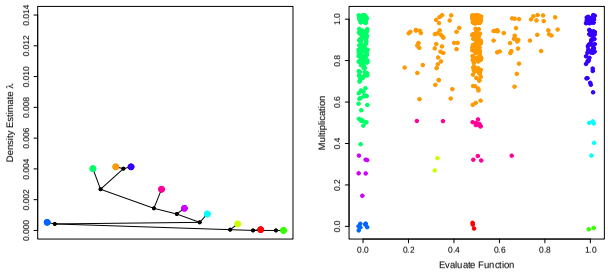



Figure 7: (a) Cluster tree (b) Cluster assignments

Figure 7b shows the (jittered) estimated capabilities for 551 students on two skills, Evaluate Functions and Multiplication, from an online mathematics tutor (Assistment system; Heffernan, Koedinger, Junker 2001). The BKDE cross-validated bandwidth was 0.032. The corresponding cluster tree (Figure 7a) has 10 different skill set profiles (clusters). Figure 7b shows the subsequent partition. Note that the two highest splits in the tree separate the clusters in the top half section of the feature space; these students have partial to complete mastery of Multiplication and are separated by their Evaluate Function mastery level. The more well-separated skill set profiles are identified earlier.

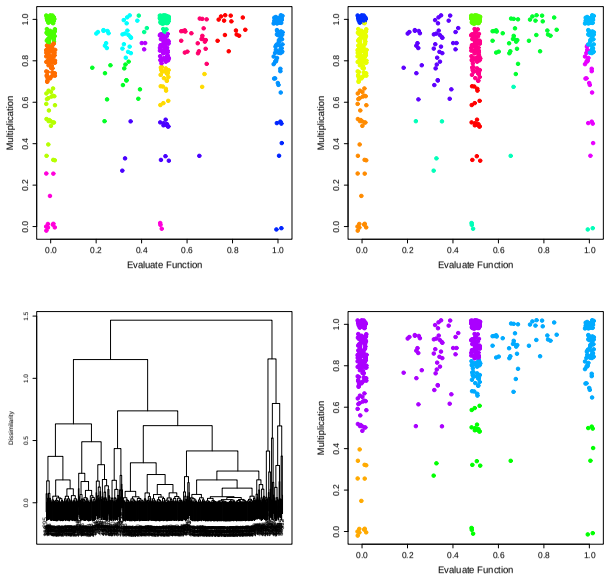



Figure 8: (a) K-means assignment (k=14); (b) Model-based clustering assignment (k = 11);(c) Complete linkage dendrogram; (d) Complete linkage cluster assignment (k = 4)

Figure 8 shows selected results from other clustering procedures. The total within-cluster sum of squares criterion suggested a k-means solution with 14 or 15 clusters. Figure 8a illustrates its propensity to partition into small tight spheres (perhaps overly so). Given a search range of two to 18 clusters with no constraints, model-based clustering chose a eleven cluster solution (Figure 8b). The "strips" are divided into cohesive sections of feature space; the "uncertain mastery" students are clustered together. Note that model-based clustering identified a very loose cluster of students with poor Multiplication mastery and Evaluate Function mastery ranging from 0.2 to 1 that overlaps with a tighter cluster of students with an Evaluate Function value of 0.5. This skill profile is a casualty of the assumption that each mixture component in the density estimate corresponds to a group in the population. However, overall the results seem to give more natural skill profile groupings than k-means. All hierarchical linkage methods yielded dendrograms without obvious cluster solutions; Figures 8c,d show an example of a four cluster complete linkage solution. Students with different Evaluate Function mastery levels are combined; in this respect, the results are unsatisfying.

Developing 10 different instruction methods for a group of students is unreasonable in practice. We use CWC to construct a cluster tree for $\alpha = 0.90$ to

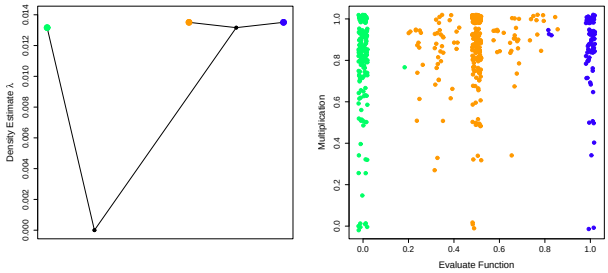



Figure 9: (a) 10% confidence cluster tree (b) cluster assignments

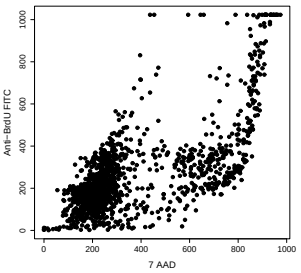
identify skill set profiles in which we are at least 10% confident. Figure 9 shows the subsequent cluster tree and assignments. At 10% confidence, we have only three skill set profiles, each identified only by their mastery level of Evaluate Functions. This result may be a more reasonable partition given the less clear separation along Multiplication mastery. Note that there are instances in which close "fluff" observations are assigned to different "cores", a result of the tie-breaker.

### 4.2 "Automatic Gating" in Flow Cytometry

Flow cytometry is a technique for examining and sorting microscopic particles. Fluorescent tags are attached to mRNA molecules in a population of cells and passed in front of a single wavelength laser; the level of fluorescence in each cell (corresponding, for example, to level of gene expression) is recorded. We might be interested in discovering groups of cells that high fluorescence levels for multiple channels or groups of cells that have different levels across channels. A common identification method is "gating" or subgroup extraction from two-dimensional plots of measurements on two channels. Most commonly, these subgroups are identified subjectively by eyeballing the graphs. Clustering techniques would allow for more statistically motivated subgroup identification (see Lo, Brinkman, Gottardo 2008 for one proposed method).

Figure 10a shows 1545 flow cytometry measurements on two fluorescence markers applied to Rituximab, a therapeutic monoclonal antibody, in a drug-screening project designed to identify agents to enhance its antilymphoma activity (Gottardo, Lo 2008). Cells were stained, following culture, with the agents anti-BrdU and the DNA binding dye 7-AAD.

We construct a cluster tree (cross-validated BKDE bandwidth of 21.834) and plot the cluster assignments indicating whether the observations are "fluff" or part of a cluster "core" (Figures 10b,c). The cluster tree has 13 leaves (8 clusters, 5 modal artifacts). The cores



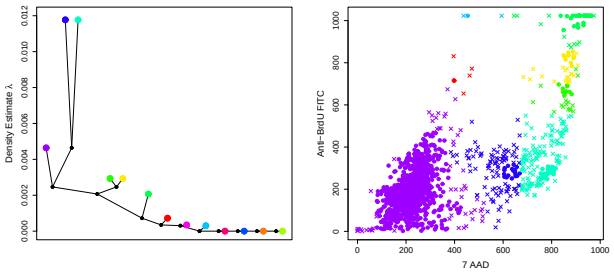




Figure 10: (a) Flow cytometry measurements on the two fluorescent markers anti-BrdU and 7-AAD; (b) Cluster tree with 13 leaves (8 clusters, 5 artifacts); (c) Cluster assignments: "fluff" = x; "core" = •

are located in the high frequency areas; the fluff is appropriately assigned. The sizes of the cores give some evidence as to their eventual significance. For example, the core near (400, 700) consists of one observation; we would not expect the corresponding cluster to remain in the confidence cluster tree for any reasonable $\alpha$.

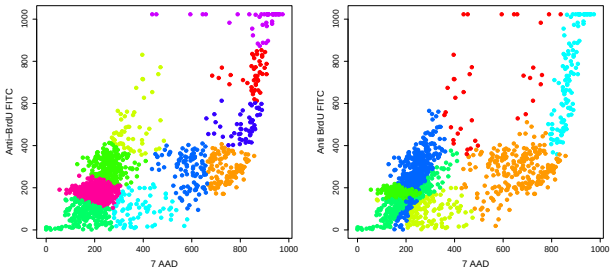



Figure 11: (a) K-means assignment (k=10); (b) Model-based clustering assignment (k = 7)

Figure 11a shows a ten-cluster k-means solution similarly chosen as before. While many of the smaller clusters are found by both k-means and the cluster tree, k-means splits the large cluster in Figure 10c into several smaller clusters. Model-based clustering chooses a seven-cluster solution (Figure 11b). We again see overlapping clusters in the lower left quadrant; this solution would not be viable in this application. Linkage methods again provided unclear solutions.

We use CWC to construct a confidence cluster tree for $\alpha = 0.10$; we are at least 90% confident in the generated clusters (Figure 12). All modal artifacts have been removed, and the smaller clusters have merged into two larger clusters with cores at (200, 200) and (700, 300). Note that the right cluster is a combination of the two high 7-AAD/low anti-BrdU clusters from Figure 10c. CWC did not find enough evidence to warrant splitting this larger cluster into subgroups.

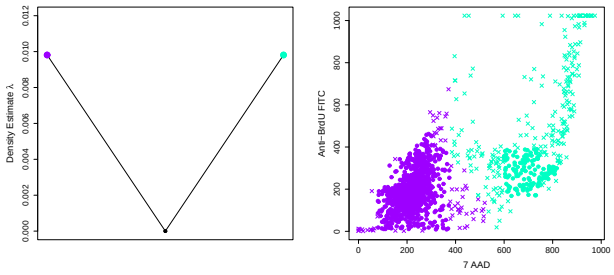



Figure 12: (b) 90% Cluster Tree with two leaves; (c) Cluster assignments: "fluff" = x; "core" = •

## 5 Summary and future work

We have presented a plug-in method for estimating the cluster tree of a density that takes advantage of the ability to exactly compute the level sets of a piecewise constant density estimate. The approach shows flexibility in finding clusters of unequal sizes and shapes. However, the cluster tree is dependent on the (inherently noisy) density estimate. We introduced *clustering with confidence*, an automatic pruning procedure that assesses significance of splits in the cluster tree; the only input needed is the desired confidence level.

These procedures may become computationally intractable as the number of adjacent bins grows with the dimension and are realistically for use in lower dimensions. One high-dimensional approach would be to employ projection or dimension reduction techniques prior to cluster tree estimation. We also have developed a graph-based approach that approximates the cluster tree in high dimensions. Clustering with Confidence then could be applied to the resulting graph to identify significant clusters.

### References


Ayers, E, Nugent R, and Dean, N. (2008) "Skill Set Profile Clustering Based on Student Capability Vectors Computed From Online Tutoring Data". *Proc. of the Int'l Conference on Educational Data Mining* (peer-reviewed). To appear.

Buja, A. (2002) Personal Communication. Also Buja, A. and Rolke, W. "Calibration for Simultaneity: (Re)Sampling Methods for Simultaneous Inference with Applications to Function Estimation and Functional Data". In revision.

Cuevas, A., Febrero M., and Fraiman, R. (2000) "Estimating the Number of Clusters". *The Canadian Journal of Statistics*, 28:367-382.

Cuevas, A., Febrero M., and Fraiman, R. (2001) "Cluster Analysis: a further approach based on density estimation". *Computational Statistics & Data Analysis*, 36:441-459.

Fraley, C and Raftery, A (1998). "How Many Clusters? Which Clustering Method? - Answers Via Model-Based Cluster Analysis". *The Computer Journal*, 41:578-588.

Gottardo, R. and Lo, K.(2008) `flowClust` Bioconductor package

Hall, P. and Wand, M. P. (1996) "On the Accuracy of Binned Kernel Density Estimators". *Journal of Multivariate Analysis*, 56:165-184.

Hartigan, J. A. (1975) *Clustering Algorithms*. Wiley.

Hartigan, J. A. (1981) "Consistency of Single Linkage for High-Density Clusters". *Journal of the American Statistical Association*, 76:388-394.

Hartigan, J. A. (1985) "Statistical Theory in Clustering". *Journal of Classification*, 2:63-76.

Hartigan, J and Wong, M.A. (1979). A k-means clustering algorithm. *Applied Statistics*, 28, 100-108.

Heffernan, N.T., Koedinger K.R., and Junker, B. W. (2001) *Using Web-Based Cognitive Assessment Systems for Predicting Student Performance on State Exams.* Research proposal to the Institute of Educational Statistics, US Department of Education. Department of Computer Science at Worcester Polytechnic Institute, Massachusetts.

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

Klemelä, J. (2004) "Visualization of Multivariate Density Estimates with Level Set Trees". *Journal of Computational and Graphical Statistics*, 13:599-620.

Lo, K., Brinkman R., and Gottardo, R. (2008). "Automated Gating of Flow Cytometry Data via Robust Model-Based Clustering". *Cytometry, Part A*, 73A: 321-332.

Mardia, K., Kent, J.T., and Bibby, J.M. (1979) *Multivariate Analysis*. Academic Press.

McLachlan, G.J. and Basford, K.E. (1988) *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker.

Sedgewick, Robert. (2002) *Algorithms in C, Part 5: Graph Algorithms, 3rd Ed.* Addison-Wesley.

Stuetzle, W. (2003) "Estimating the Cluster Tree of a Density by Analyzing the Minimal Spanning Tree of a Sample". *Journal of Classification*. 20:25-47.

Walther, G. (1997) "Granulometric Smoothing". *The Annals of Statistics*, 25:2273-2299.

Wand, M. P. and Jones, M. C. (1995) *Kernel Smoothing*. Chapman & Hall.

Wishart, D. (1969) "Mode Analysis: A Generalization of Nearest Neighbor which Reduces Chaining Effect. *Numerical Taxonomy*, ed. A. J. Cole, Academic Press, 282-311.