# Estimating the Cluster Tree of a Density

**Werner Stuetzle**
**Rebecca Nugent**

Department of Statistics
University of Washington

# 1. Meanings of "clustering"

The term "clustering" is used to signify both *dissection* and *concept formation*.

**Dissection:** Partition a collection of items into *compact subsets.*

Can for example quantify degree of compactness of a partition $\mathcal{P}_k = P_1, \ldots, P_k$ by sum of squared distances of observations from their group means:

$$\mathbf{rss}\left(\mathcal{P}_k\right) = \sum_{i=1}^{k} \sum_{j \in P_i} \left\| \underline{x}_j - \bar{\underline{x}}_i \right\|^2 .$$

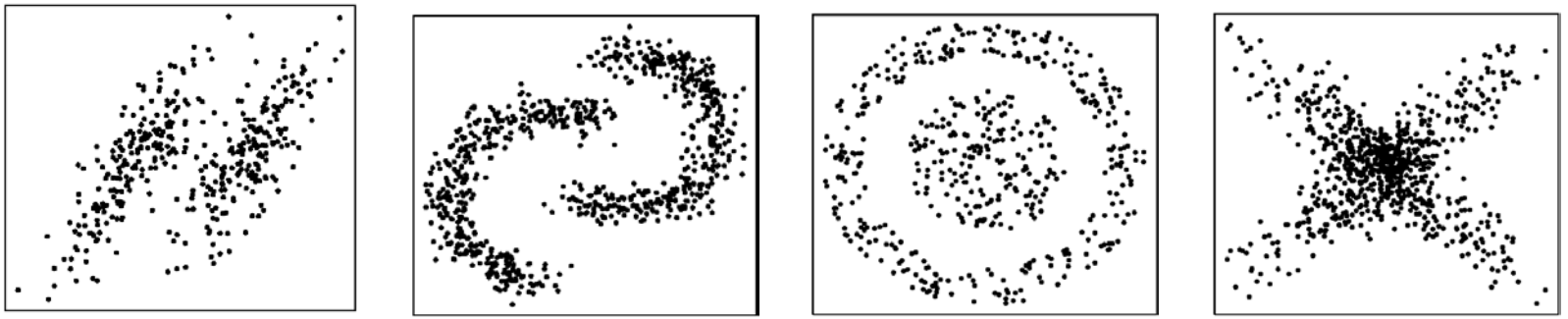
$\Rightarrow$ K-means clustering; (locally) optimal partitions can be found with the Lloyd algorithm.

**Concept formation:** Detect presence of distinct groups.

Definition of *distinct groups* (CG&J):

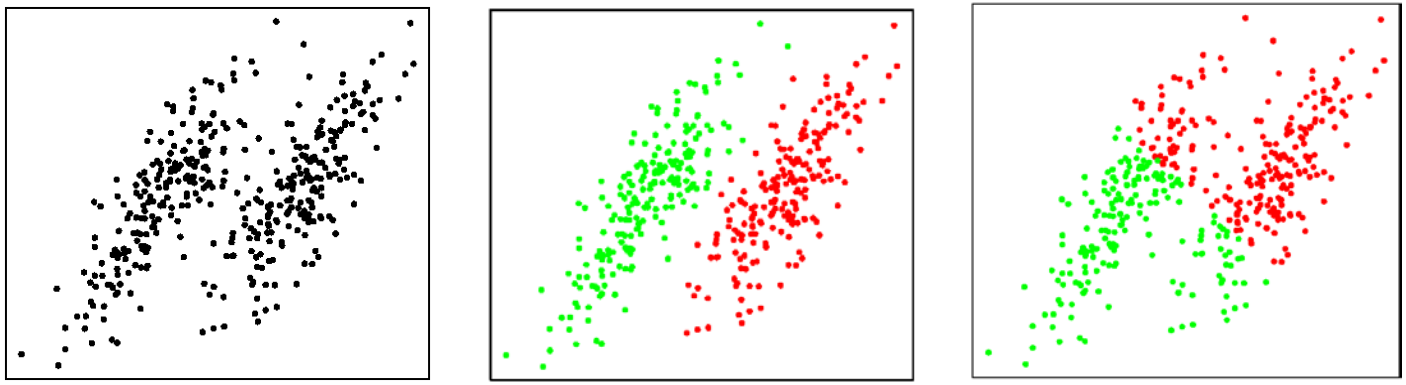
Contiguous, densely populated areas of feature space, separated by contiguous, relatively empty regions.

(a) - (c): Distinct groups in the sense of CG&J;
(d): not covered by definition.



## Note:

- Dissection and concept formation can result in different partitions.



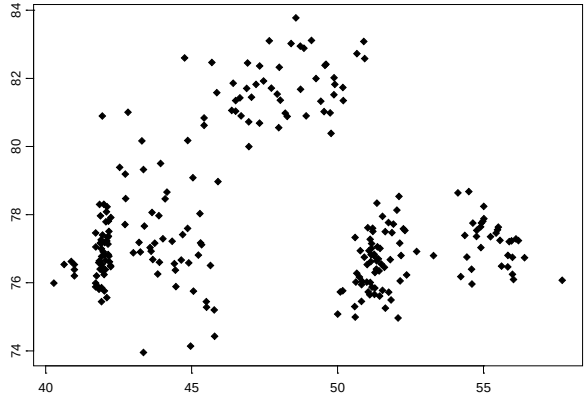

- Several popular clustering methods (complete linkage, k-means) are really dissection methods.
- There are applications, like vector quantization, where disscection makes sense, even if there are no distinct groups in the data.

- Dissection methods can be successful at concept formation if

  (i) We guessed the right number of groups, and
  (ii) Groups are approximately spherical, with the same radius.

**From now on will take "clustering" to mean "concept formation"**

# 2. Statistical approaches to clustering




- Detect that there are 5 or 6 distinct groups.
- Assign group labels to observations.

Need to specify sampling model and population characteristic of interest.

Without sampling model, concept of "cluster validity" does not make sense.

Without well specified population characteristic it is impossible to evaluate and compare clustering methods $\Rightarrow$ no "progress".

Sampling model in this talk:

Feature vectors $\underline{x}_1, \ldots, \underline{x}_n$ are iid sample from some density $p(\underline{x})$.

## 2.1 Parametric approach (model-based clustering)

Based on premise that each group $g$ is represented by density $p_g$ that is a member of some parametric family (e.g., multivariate Gaussian) $\Rightarrow p(\underline{x})$ is a mixture:

$$p(\underline{x}) = \sum_{g=1}^{G} \pi_g \; p_g(\underline{x}) \, .$$

For given number of groups $G$, mle's for group means and covariances can be found using EM.

Alternatively, can used David Scott's squared error criterion.

Number of groups $G$ can be estimated, for example by cross-validation or BIC.

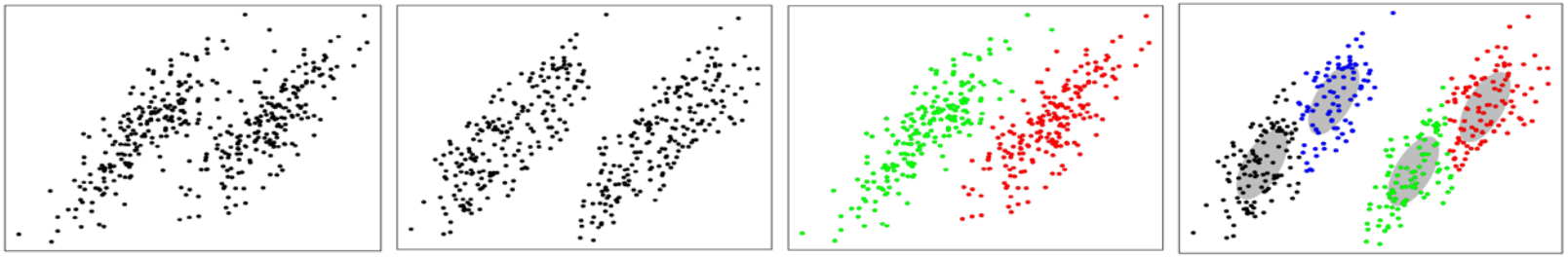
Observations can be labeled using Bayes' rule.

## Strength of model-based clustering:

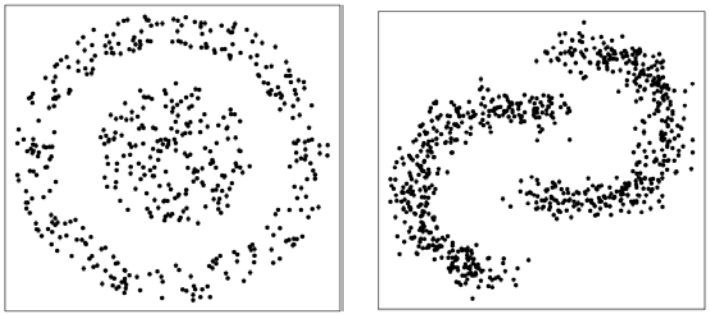
- Intellectually coherent approach.
- Offers a way of estimating the number of groups.
- Can be extended (background noise, multinomial mixtures).

## Weakness of model-based clustering:

- Can lead to unexpected results if Gaussianity assumption is violated.



- Cannot handle non-elliptical groups



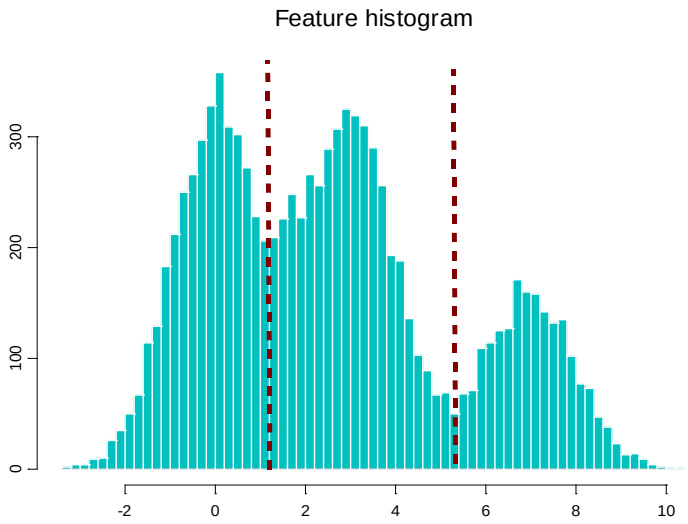
## 2.2 Nonparametric approach

Based on premise that groups correspond to modes of density $p(\underline{x})$.

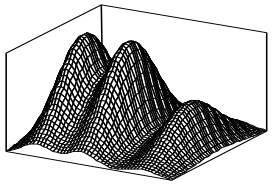
Clustering methods should be able to "detect and resolve distinct data modes, independently of their shape and variance" (Wishart 1969).

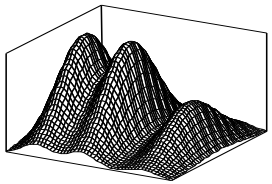
Need to

- Estimate modes;
- Assign each observation to the "domain of attraction" of a mode.

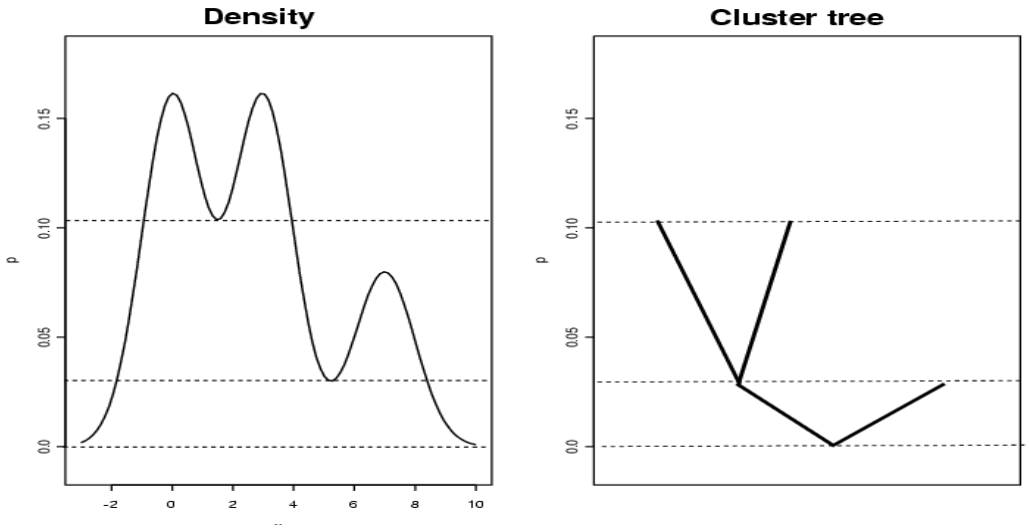



**Will pursue nonparametric approach**





# Structure of level sets is described by cluster tree



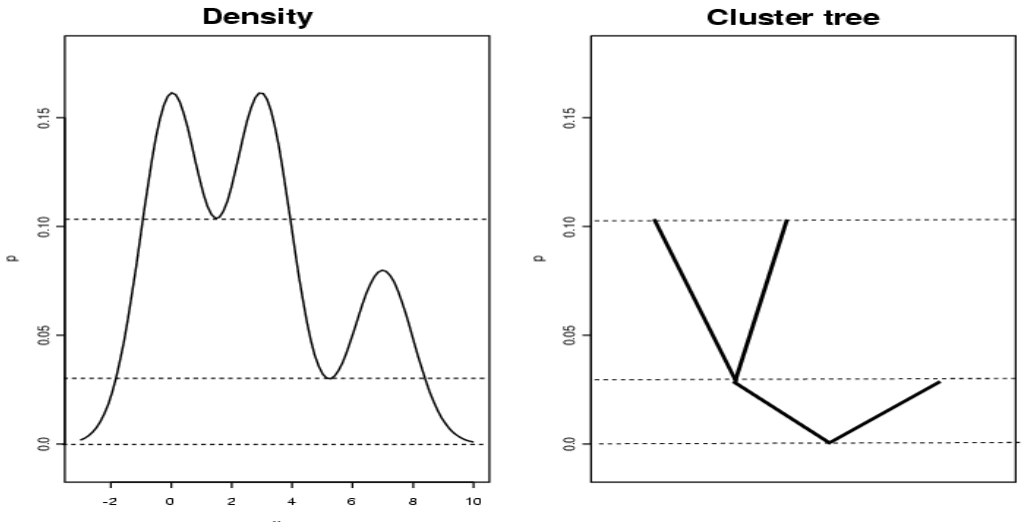

Cluster tree is easiest to define recursively:

Each node N of cluster tree

- represents a subset D(N) of feature space (high density cluster);
- is associated with a density level $\lambda(N)$.

Root node

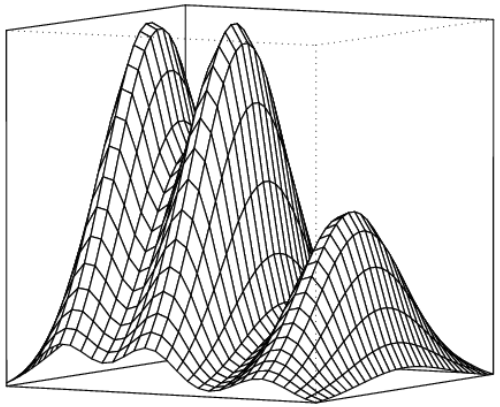
- represents the entire support of the density;
- is associated with density level $\lambda(N) = 0$.

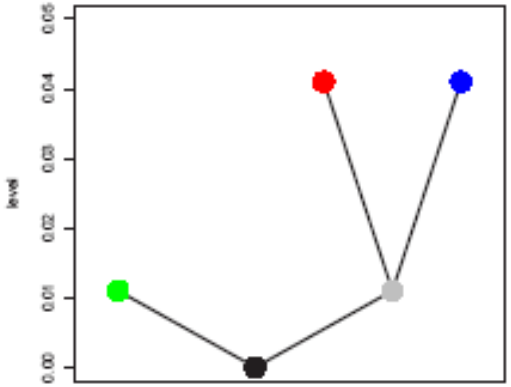


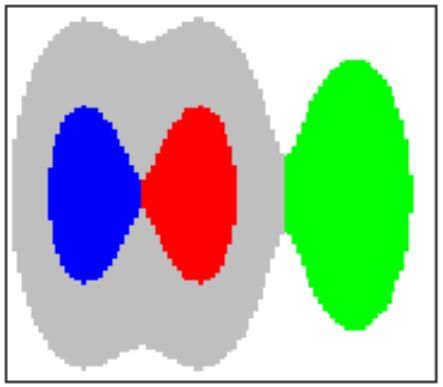

To determine descendents of node N:

- Find lowest level $\lambda_d$ for which $L(\lambda; p) \cap D(N)$ has two connected components.
- If there is no such $\lambda_d$ then $N$ is leaf of the tree.
- Otherwise, create daughter nodes representing the connected components, with associated level $\lambda_d$, and recurse.

# Density, cluster tree, and high density clusters





Leaves of cluster tree correspond to modes of density.

Cluster tree is invariant under non-singular affine transformations of feature space.

**Cluster tree is (a) target population characteristic in non-parametric clustering.**

# 4. Plug-in estimates of the cluster tree

Obvious idea:

- Estimate $p$ by (nonparametric) density estimate $\hat{p}$;
- Estimate cluster tree of $p$ by cluster tree of $\hat{p}$.

However, there are computational as well as statistical problems.

(i) Computational problem:

How can we compute level sets and their connected components?

(ii) Statistical problem:

How do we distinguish spurious components (modes) due to sampling variability from real components reflecting the structure of the true density?

## Computing level sets and their connected components

For density estimates $\hat{p}$ that are piecewise constant over (hyper-) rectangles:

$$\hat{p(\underline{x})} = \sum_{i=1}^{m} c_i \; I(\underline{x} \in R_i) \, ,$$

level sets, their connected components, and the cluster tree can be computed exactly.

Example: Histograms, ASH estimates, piecewise constant approximations of other estimates.

Only viable in low dimensions ($dim \leq 4$ ?)

Otherwise, have to use approximations (current research).

Notable exception: 1-NN density estimate.

# 5. The cluster tree of the 1-NN density estimate

**Given:** Observations $\mathcal{X} = \underline{x}_1, \ldots, \underline{x}_n \in R^m \sim p(\underline{x})$.

1-NN density estimate:

$$\hat{p}_1(\underline{x}) \sim \frac{1}{d(\underline{x}, \mathcal{X})}.$$

$L(\lambda; \hat{p}_1)$ is union of open spheres around the $\underline{x}_i$ with radius $1/\lambda$.

The cluster tree of $\hat{p}_1$ is closely connected to the minimal spanning tree (MST) $T$ of X.

Let $T(d)$ be threshold graph obtained by removing all edges of $T$ with edge length $\geq d$.

**Prop** (Hartigan 1985): The connected components of $L(\lambda; \hat{p}_1)$ are the same as the connected components of $T(2/\lambda)$.

Can compute cluster tree of nearest neighbor density estimate by

- Breaking longest MST edge, thereby splitting MST into two subtrees;
- Recursively applying splitting process to subtrees.

Cluster tree of nearest neighor density estimate is isomorphic to single linkage dendogram.

Problem

1-NN density estimate has singularity at every data point.

Therefore, cluster tree of 1-NN density estimate has leaf for every data point and will be poor estimate for population cluster tree.

It has to be pruned.

# 6. Runt pruning

Consider split of high density cluster of $\hat{p}_1$ "significant" if both daughter nodes contain sufficiently many observations.

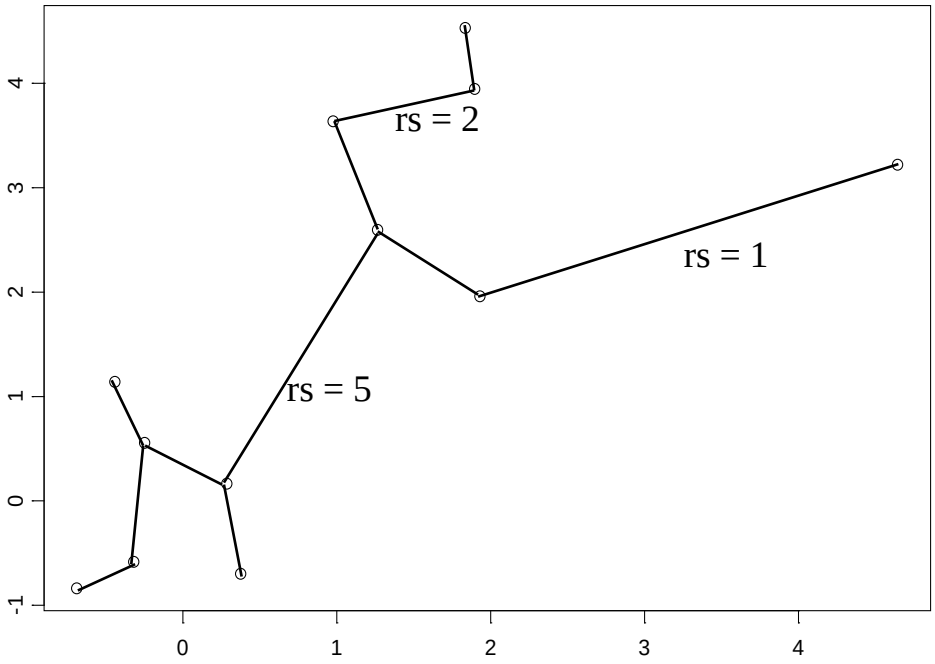
Size of smaller of the daughter nodes is called *runt size* of node.

Runt size threshold controls size of estimated cluster tree.

Maximum of runt sizes was used by Hartigan and Mohanty as test statistic in a test for unimodality.

## Define runt size of MST edge e:

- Break all MST edges with length $\geq \|e\|$;
- $T_1, T_2$ subtrees of $T(\|e\|)$ rooted at endpoints of $e$.
- runtsize$(e) = \min(|T_1|, |T_2|)$.



## Heuristic motivation / justification

Recall multi-fragment algorithm for MST construction:

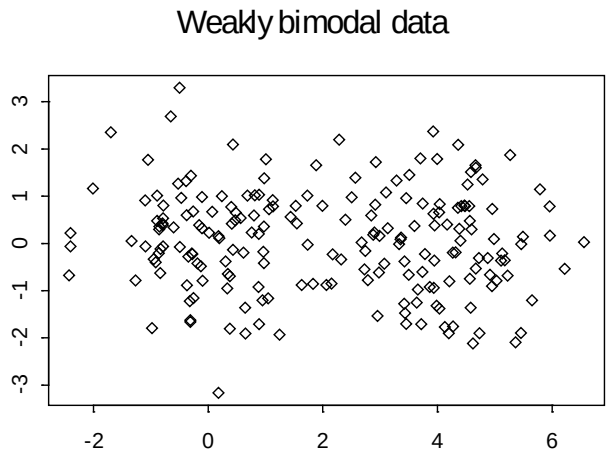
- Define distance between groups as minimum distance between obs.
- Initialize each obs to form its own group.
- Repeat {
    - Find closest groups.
    - Add shortest edge connecting them.
    - Merge them.

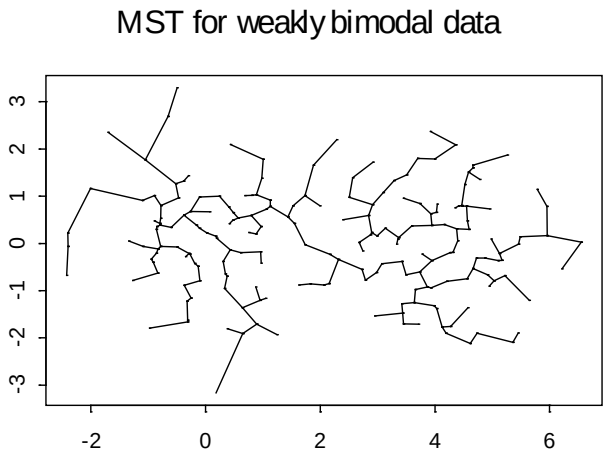
} Until only one group remains.

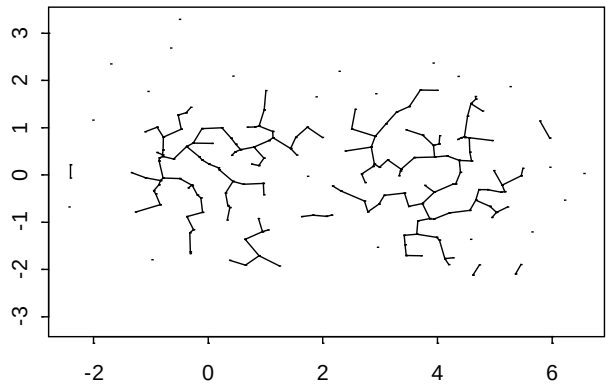
MST fragments will start and grow in high density regions, where distances are small.

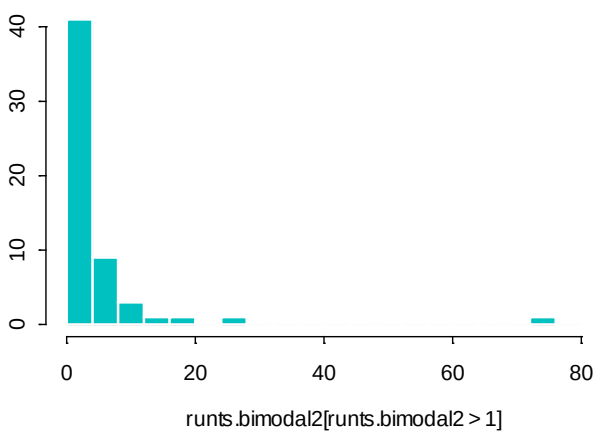
Eventually, these fragments will be joined by edges.

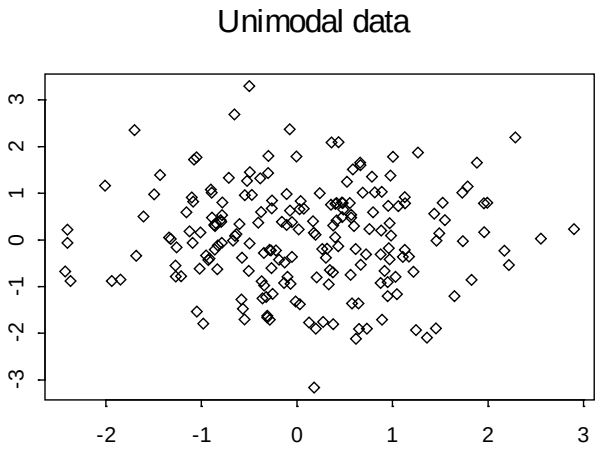
Those edges will have large runt size.

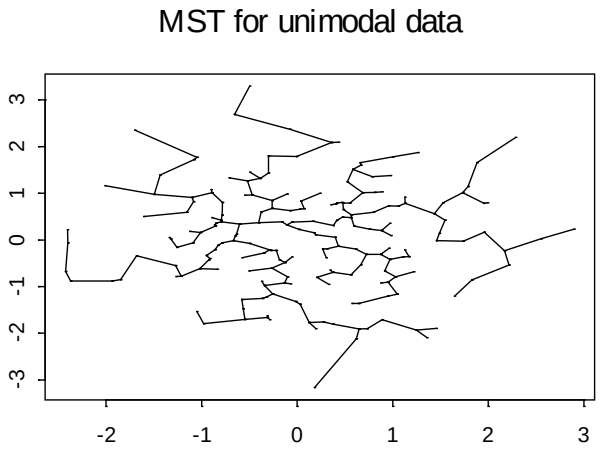

Weakly bimodal data
-3
-2
-1
0
1
2
3
-2
0
2
4
6

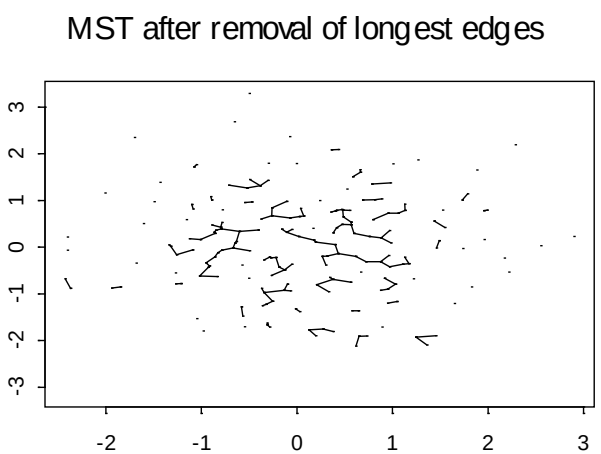

MST for weakly bimodal data
-3
-2
-1
0
1
2
3
-2
0
2
4
6

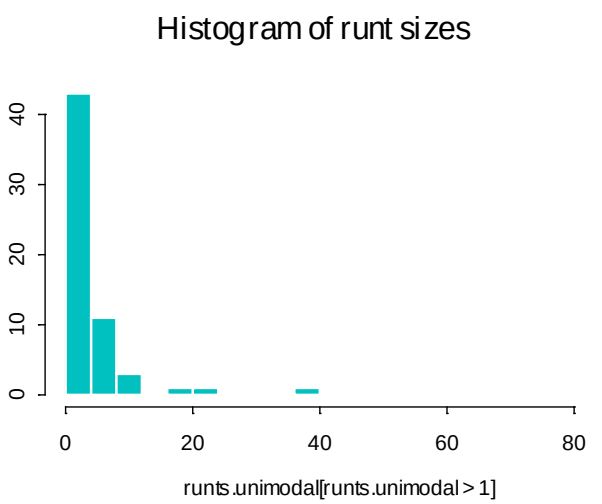

MST after removal of longest edges
-3
-2
-1
0
1
2
3
-2
0
2
4
6


Histogram of runt sizes
0
10
20
30
40
0
20
40
60
80
runts.bimodal2[runts.bimodal2 > 1]


Unimodal data
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3


MST for unimodal data
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3


MST after removal of longest edges
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3


Histogram of runt sizes
40
30
20
10
0
0
20
40
60
80
runts.unimodal[runts.unimodal > 1]

## Relationship between runt pruning and single linkage clustering

Single linkage clustering = standard way of extracting clusters from MST.

To obtain $k$ clusters

- Break $k-1$ longest MST edges or, equivalently
- Cut dendogram at a level resulting in $k$ subtrees.

Bad idea - thanks, Jon, for pointing that out!

## Problems with dendogram cutting:

Breaking longest edges tends to separate stragglers from the bulk of the data and often results in one large and many small clusters.

Choosing a single threshold for edge length means we are finding the connected components of $L(\lambda; \hat{p}_1)$ for a single level $\lambda$.

However, there might not be a single cut level that reveals all the modes.

Therefore, problem with single linkage cannot be fixed by discarding small clusters.

Instead, find all the dendogram nodes with runt size $\geq$ threshold.

# 7. Illustration of Runt Pruning

**Objects:** 572 olive oil samples coming from 9 different areas, grouped into 3 regions (1, 2, 3, 4) (5, 6) (7, 8, 9).

**Features:** Concentration of 8 different chemicals.

**Question:** How well can we recover the grouping into regions and areas?

Note: To empirically evaluate performance of clustering methods, need labeled data.

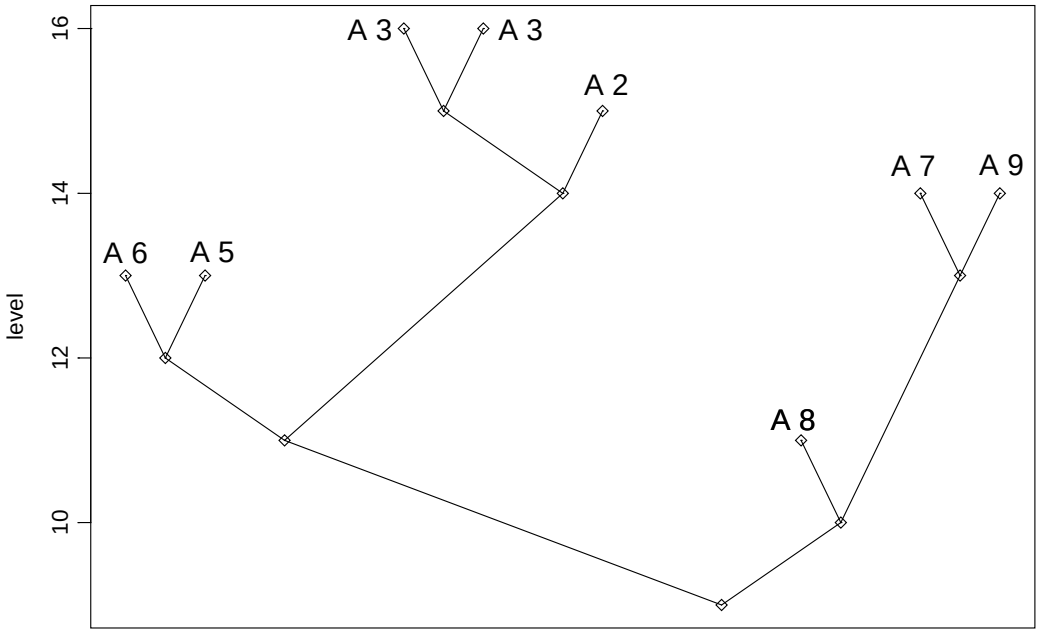
20 largest runt sizes:

168 97 59 51 42 42 33 13 13 12 11 11 11 10 10 8 8 8 8 7

Fairly clear gap: Choose runt size 33 as threshold.

Note: Situation not always that clear cut

# Estimated cluster tree for Olive Oil data




Interpretation:

- Bottom split separates region 3 from regions 1, 2.
- Next split on left separates region 1 from region 2
- Not able to correctly partition region 1 into areas

# Areas vs clusters

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| A1 | 0 | 1 | 0 | 0 | 0 | 17 | 0 | 7 |
| A2 | 0 | 51 | 1 | 0 | 0 | 4 | 0 | 0 |
| A3 | 90 | 11 | 103 | 1 | 0 | 0 | 1 | 0 |
| A4 | 5 | 13 | 4 | 0 | 0 | 14 | 0 | 0 |
| A5 | 0 | 0 | 0 | 64 | 1 | 0 | 0 | 0 |
| A6 | 0 | 0 | 0 | 0 | 33 | 0 | 0 | 0 |
| A7 | 0 | 3 | 0 | 0 | 0 | 43 | 0 | 4 |
| A8 | 0 | 2 | 0 | 0 | 0 | 2 | 45 | 1 |
| A9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 51 |

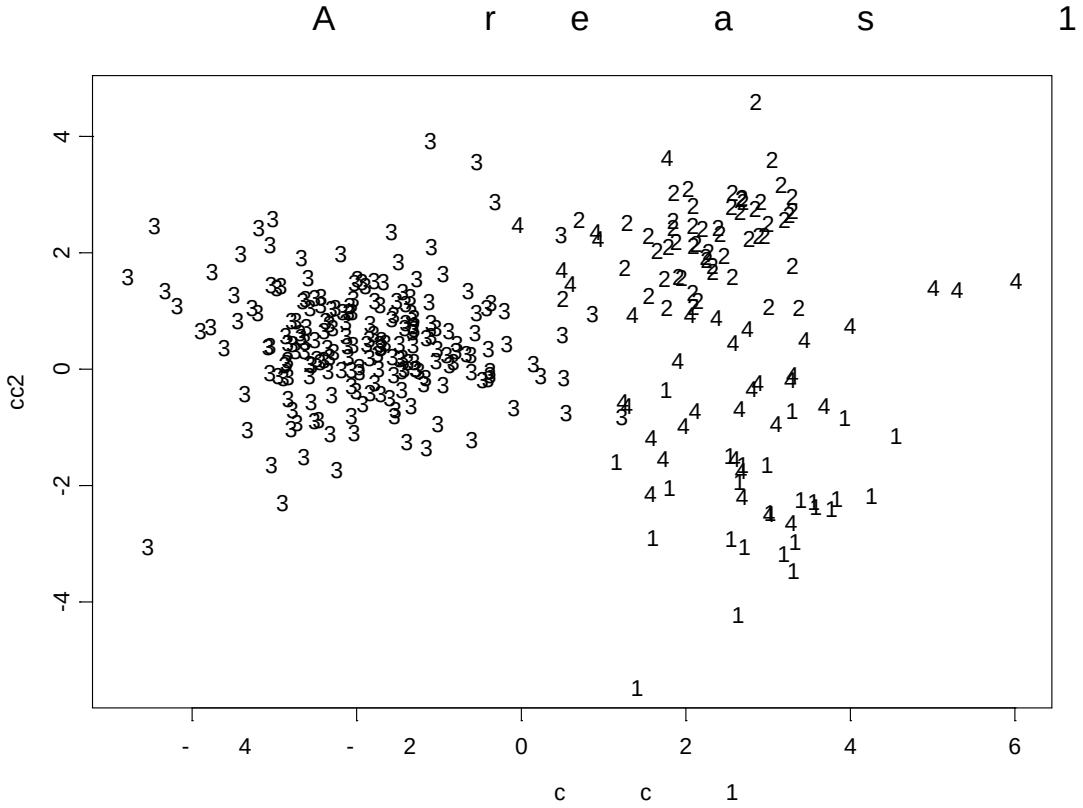
Note able to recognize Areas 1 and 4 in Region 1.

Splits up Area 3.

# How well are Areas 1-4 separated?

Draw projection onto first two discriminant coordinates.

Note: Only possible because we know labels.



# 8. Current research

## 8.1 Computing the cluster tree for other density estimates

**Idea:** Approximate geometric problem of finding connected components of level sets by graph problem.

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

Let $G$ be the complete graph over $\underline{x}_1, \ldots, \underline{x}_n$ with edge weights $\hat{p}_{ij}$ and vertex weights $\hat{p}_i$.

Let $G(\lambda)$ be the threshold graph obtained from $G$ by removing edges with $\hat{p}_{ij} \leq \lambda$ and vertices with $\hat{p}_i \leq \lambda$.

Apply recursive operations of thresholding and finding connected components to the graph $G$ instead of feature space.

## Motivation:

Observations in the same connected component of $G(\lambda)$ are in the same connected component of $L(\lambda; \hat{p})$.

Observations in different connected components of $G(\lambda)$ might be in the same connected component of $L(\lambda; \hat{p})$, but if $\hat{p}$ is smooth this is unlikely.

## 8.2 Clustering with confidence

Problem: Density estimates may have spurious modes due to sampling variability.

Idea:

Compute Bootstrap density estimates.

Connected components present in most Boostrap estimates are probably "real".

Note enough time to give details.

# 9. Summary

The term "clustering" is ambiguous; need to distinguish between *dissection* and *concept formation*.

Goal of concept formation: detect presence of distinct groups.

Premise of nonparametric clustering: groups $\sim$ modes of feature density.

Structure of collection of level sets is described by cluster tree; modes $\sim$ leaves.

Cluster tree is defined recursively — suggests recursive partitioning method for its computation.

For some density estimates, cluster tree can be computed exactly.

For others, cluster tree has to be approximated by solution of a graph problem.

Runt pruning is a simple attempt at eliminating spurious modes.

May be able to use Bootstrap for a more principled approacch.

**Thank you for your interest.**

# 9. Summary

The term "clustering" is ambiguous; need to distinguish between *dissection* and *concept formation*.

Goal of concept formation: detect presence of distinct groups.

Premise of nonparametric clustering: groups $\sim$ modes of feature density.

Structure of collection of level sets is described by cluster tree; modes $\sim$ leaves.

Cluster tree is defined recursively — suggests recursive partitioning method for its computation.

For some density estimates, cluster tree can be computed exactly.

For others, cluster tree has to be approximated by solution of a graph problem.

May be able to use Bootstrap for distinguishing between "real" and "spurious" modes.

Thanks for your interest.

# 5. Computing the cluster tree for piecewise constant density estimates

Suppose that density estimate $\hat{p}$ is piecewise constant over disjoint (hyper-) rectangles forming a partition of feature space:

$$\hat{p}(\underline{x}) = \sum_{i=1}^{m} c_i \; I(\underline{x} \in R_i) \, .$$

Example: Histograms, ASH estimates, Cart estimates, kernel estimates with hyper-rectangular kernels (careful !).

In this case, level sets, their connected components, and the cluster tree can be computed exactly.

Basic idea: Convert geometry problem into graph problem.

Define weighted graph $G$:

- Vertices represent rectangles $R_i$;
- Edges encode adjacency: $(i, j)$ is an edge if $R_i$ is adjacent to $R_j$;
- Weight of vertex $i$ is value of density in rectangle $i$: $w_i = c_i$;
- Weight of edge $(i, j)$ is minimum of density in the two (adjacent) rectangles: $w_{ij} = \min(c_i, c_j)$.

Define *threshold graph* $G(\lambda)$:

- Remove all vertices with $w_i \leq \lambda$;
- Remove all edges with $w_{ij} \leq \lambda$.

Connected components of level set $L(\lambda; \hat{p})$ correspond to connected components of threshold graph $G(\lambda)$.

Finding connected components of $G(\lambda)$ is a standard graph problem:

1. Start graph traversal at an arbitrary vertex and mark all visited vertices;
2. Remove visited vertices and their incident edges;
3. Repeat (1) and (2) until no more vertices remain.

The cluster tree can be computed recursively:

- Each node represents a sub-graph of $G$ and the corresponding subset of feature space;
- The root node represents the entire graph $G$ and the support of $\hat{p}$;
- To find the descendents of a node $N$ representing a graph $H$ we find the smallest value of $\lambda$ for which $H(\lambda)$ has two connected components;
- If there is no such $\lambda$ then $N$ is a leaf of the tree;
- Otherwise we create daughter nodes representing the connected components of $H(\lambda)$ and recurse;

## What about more general density estimates?

Exact computation of level sets seems daunting; how would we even represent them?

In low dimensions ($\leq 4$ ?) can approximate $\hat{p}$ by piecewise constant density $p^*$:

- Put down grid;
- For grid cell $R_i$ with center $\underline{u}_i$ define density $c_i = \hat{p}(\underline{u}_i)$.

Then apply algorithm for piecewise constant estimate.

# 4. Estimating the cluster tree of a density

Obvious plug-in approach:

- Estimate $p$ by (nonparametric) density estimate $\hat{p}$;
- Estimate cluster tree of $p$ by cluster tree of $\hat{p}$.

However, there are computational as well as statistical problems.

(i) Computational problem:

How can we compute the number of connected components of a level set $L(\lambda; \hat{p})$?

(ii) Statistical problem:

How do we distinguish spurious components (modes) due to sampling variability from real components reflecting the structure of the true density?

Will focus on problem (i)

# Digression: nonparametric density estimation

Given: Feature vectors $\underline{x}_1, \ldots, \underline{x}_n \sim p(\underline{x})$.
Goal: Estimate $p(\underline{x})$.

Histogram estimate:

- Partition feature space into (axis parallel rectangular) bins.
- Estimate density in bin by bin count / ($n$ * bin volume)

Average shifted histograms (ASH) estimate:

- Compute histograms for several shifted versions of the grid.
- Average them.

(Will give smoother estimates.)

## Kernel and near-neighbor estimates:

Let $S(\underline{x}, r)$ be sphere in feature space with radius $r$, centered at $\underline{x}$.

Assuming $p(\underline{x})$ is roughly constant over $S(\underline{x}, r)$, expected number $k$of sample points in $S(\underline{x}, r)$ is

$$k \approx n\ p(\underline{x})\ \mathrm{Vol}(S(\underline{x}, r)), \quad \text{or} \quad p(\underline{x}) \approx k/(n\ \mathrm{Vol}(S(\underline{x}, r))\,.$$

Kernel estimate:

Fix radius $r$; $k = \#$ of sample feature vectors in $S(\underline{x}, r)$.

$k$-NN estimate:

Fix count $k$; $r$ =smallest radius for which $S(\underline{x}, r)$ contains $k$ sample feature vectors.

Many refinements have been suggested.

**Many other estimates:** Projection Pursuit, Cart, . . . .

## Estimating the cluster tree of a density

Easy to compute cluster tree of histogram or ASH estimate, but Histogram and ASH estimates only viable in low dimensions.

Cluster tree of 1-NN density estimate can be computed exactly but needs to be pruned.

For other kernel and near-neighbor estimates, projection pursuit estimates, etc, exactly computing cluster tree seems hard geometry problem.

Instead, solve closely related graph problem.

# 5. Runt pruning

**Given:** Observations $\mathcal{X} = \underline{x}_1, \ldots, \underline{x}_n \in R^m \sim p(\underline{x})$.

1-NN density estimate:

$$\hat{p}_1(\underline{x}) \sim \frac{1}{n\, d^m(\underline{x}, \mathcal{X})}\,.$$

Define

$$r(\lambda) = (\frac{1}{n\,\lambda})^{\frac{1}{m}}\,.$$

Level set $L(\lambda; \hat{p}_1)$ is union of open spheres around the $\underline{x}_i$ with radius $r(\lambda)$:

$$L(\lambda; \hat{p}_1) = \bigcup S(\underline{x}_i; r(\lambda))\,.$$

## Computing the cluster tree of the nearest neighbor density estimate

Let $T$ denote the Euclidean Minimal Spanning Tree of $\mathcal{X}$.

Let $T(d)$ be threshold graph obtained by removing all edges of $T$ with edge length $\geq d$.

**Prop** (Hartigan 1985): The (sample) connected components of $L(\lambda; \hat{p}_1)$ are the same as the connected components of $T(2r(\lambda))$.

Can compute cluster tree of nearest neighbor density estimate by

- Breaking longest MST edge, thereby splitting MST into two subtrees;
- Recursively applying splitting process to subtrees.

Cluster tree of nearest neighor density estimate is isomorphic to single linkage dendogram.

## Problem

1-NN density estimate has singularity at every data point.

Therefore, cluster tree of 1-NN density estimate has leaf for every data point and will be poor estimate for population cluster tree.

It has to be pruned.

## Idea:

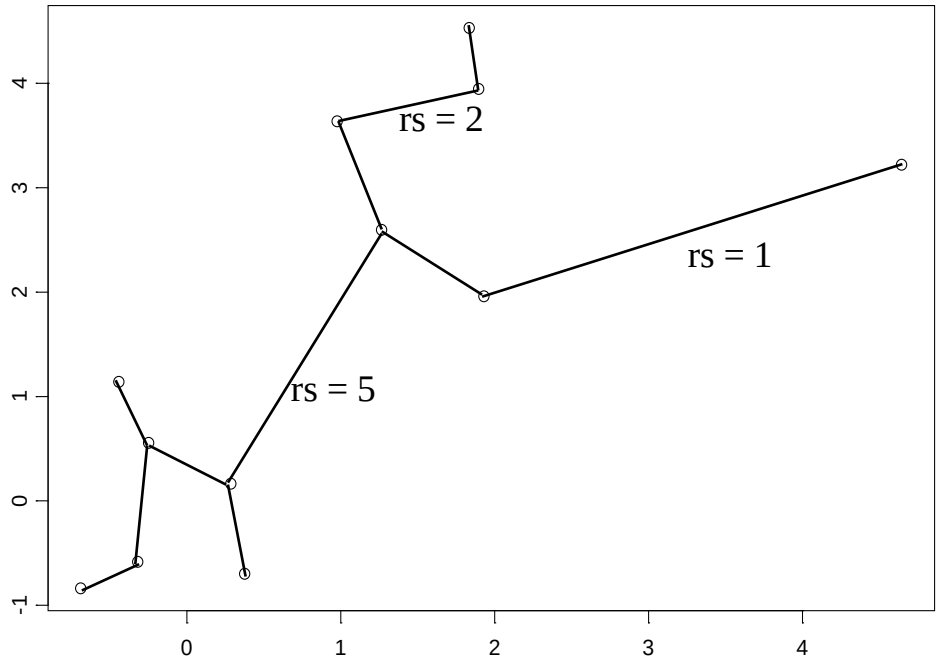
Consider split of high density cluster of $\hat{p}_1$ "significant" if both daughter nodes contain sufficiently many observations.

Size of smaller of the daughter nodes is called *runt size* of node.

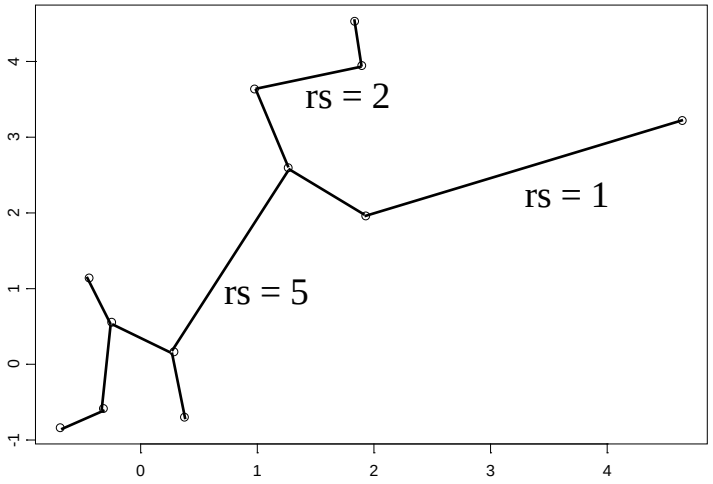
Runt size threshold controls size of estimated cluster tree.

## Define runt size of MST edge e:

- Break all MST edges with length $\geq \|e\|$;
- $T_1, T_2$ subtrees of $T(\|e\|)$ rooted at endpoints of $e$.
- runtsize$(e) = \min(|T_1|, |T_2|)$.



```
generate_cluster_tree_node (mst, runt_size_threshold) {
    node = new_cluster_tree_node
    node.leftson = node.rightson = NULL
    node.obs = leaves (mst)
    cut_edge = longest_edge_with_large_runt_size (mst, runt_size_threshold)
    if (cut_edge) {
        node.leftson = generate_cluster_tree_node (left_subtree (cut_edge), runt_size_threshold)
        node.rightson = generate_cluster_tree_node (right_subtree (cut_edge), runt_size_threshold)
    }
    return (node)
}
```



## Heuristic motivation

Recall multi-fragment algorithm for MST construction:

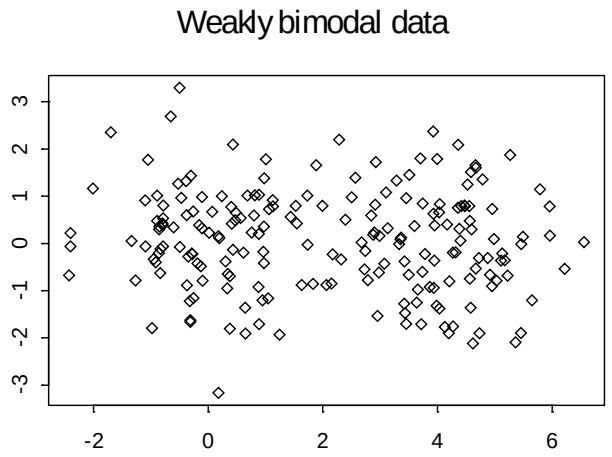
- Define distance between groups as minimum distance between obs.
- Initialize each obs to form its own group.
- Repeat {
    - Find closest groups.
    - Add shortest edge connecting them.
    - Merge closest groups.

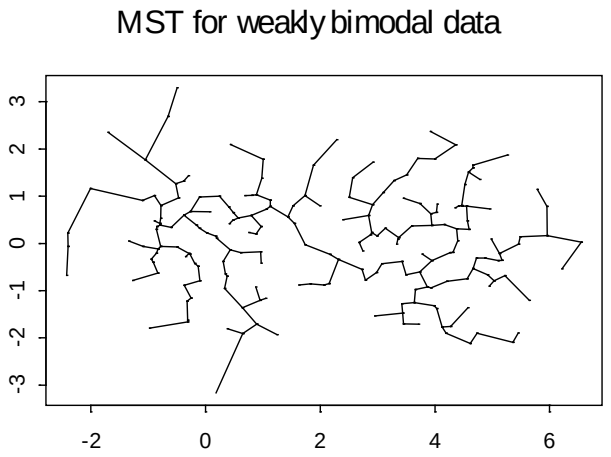
} Until only one group remains.

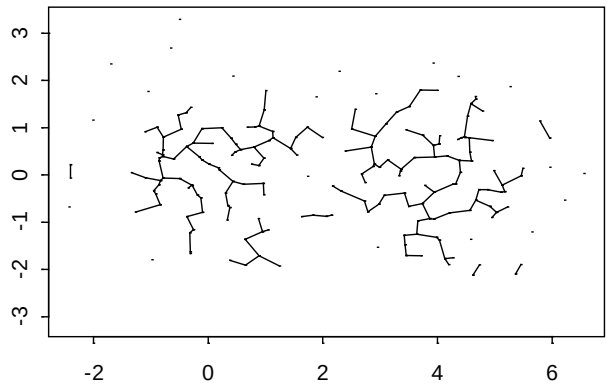
MST fragments will start and grow in high density regions, where distances are small.

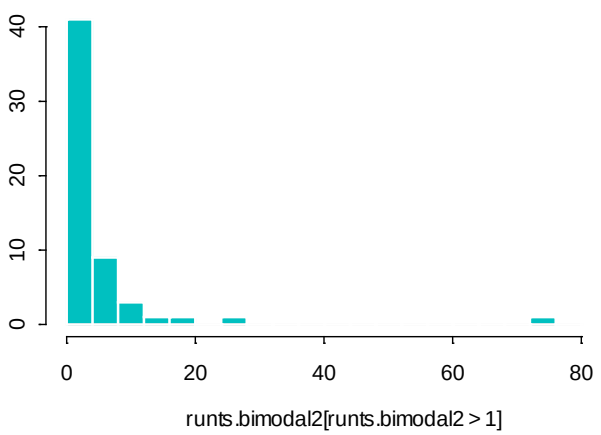
Eventually, these fragments will be joined by edges.

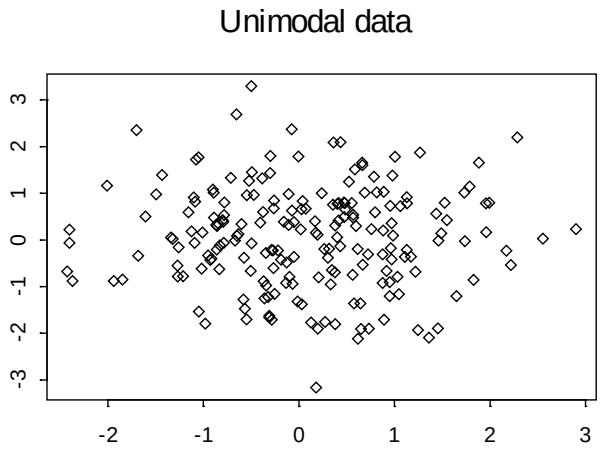
Those edges will have large runt size.

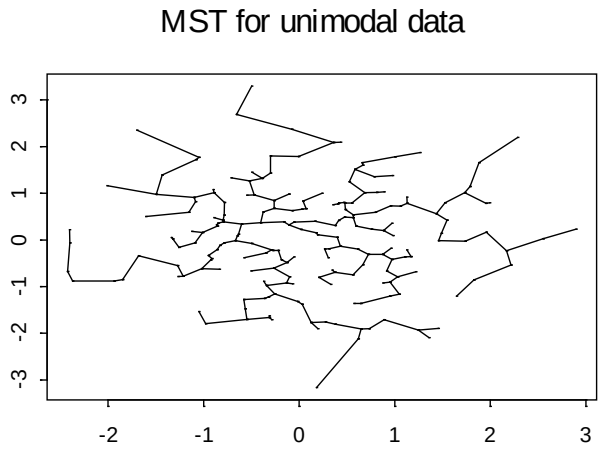

Weakly bimodal data
-3
-2
-1
0
1
2
3
-2
0
2
4
6

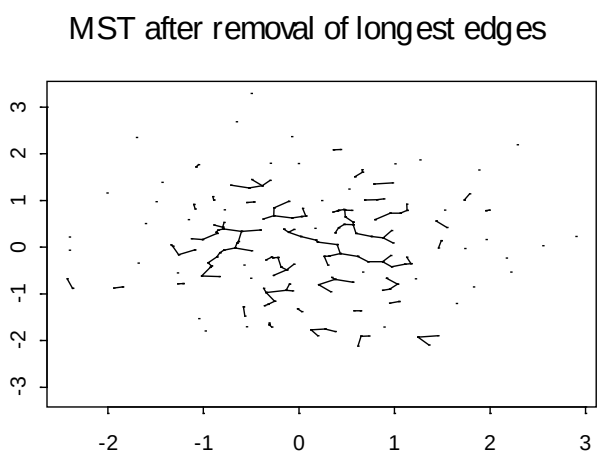

MST for weakly bimodal data
-3
-2
-1
0
1
2
3
-2
0
2
4
6

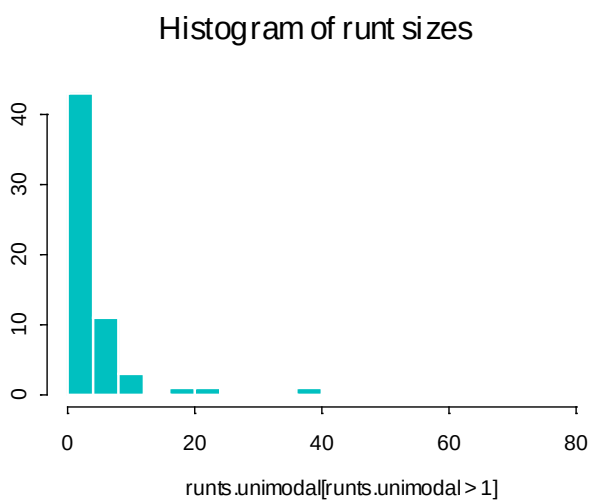

MST after removal of longest edges
-3
-2
-1
0
1
2
3
-2
0
2
4
6


Histogram of runt sizes
0
10
20
30
40
0
20
40
60
80
runts.bimodal2[runts.bimodal2 > 1]


Unimodal data
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3


MST for unimodal data
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3


MST after removal of longest edges
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3


Histogram of runt sizes
40
30
20
10
0
0
20
40
60
80
runts.unimodal[runts.unimodal > 1]

## Relationship to single linkage clustering

Single linkage clustering = standard way of extracting clusters from MST.

To obtain $k$ clusters, break $k-1$ longest MST edges.

Problems:

Breaking longest edges tends to separate stragglers from the bulk of the data and often results in one large and many small clusters (chaining)

Choosing a single threshold for edge length means choosing a single cut level for 1-NN density estimate.

However, there might not be a single cut level that reveals all the modes $\Rightarrow$

Problem with single linkage cannot be fixed by discarding small clusters.

# 6. Illustration of Runt Pruning

**Objects:** 572 olive oil samples coming from 9 different areas, grouped into 3 regions (1, 2, 3, 4) (5, 6) (7, 8, 9).

**Features:** Concentration of 8 different chemicals.

**Question:** How well can we recover the grouping into regions and areas?

Note: To empirically evaluate performance of clustering methods, need labeled data.

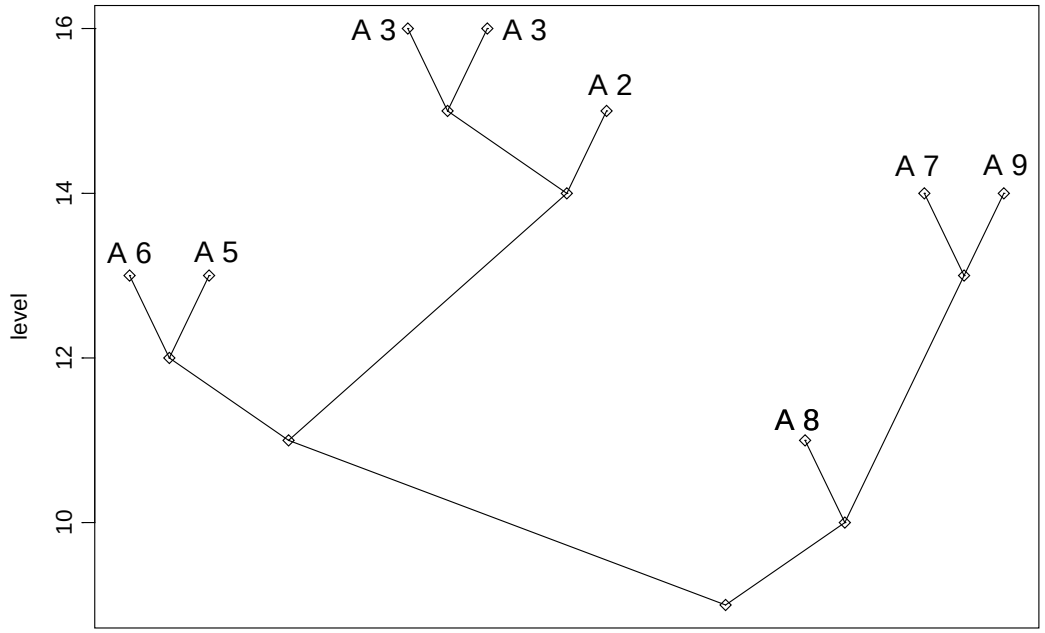
20 largest runt sizes:

168 97 59 51 42 42 33 13 13 12 11 11 11 10 10 8 8 8 8 7

Fairly clear gap: Choose runt size 33 as threshold.

Note: Situation not always that clear cut

# Estimated cluster tree for Olive Oil data




Interpretation:

- Bottom split separates region 3 from regions 1, 2.
- Next split on left separates region 1 from region 2
- Not able to correctly partition region 1 into areas

# Areas vs clusters

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| A1 | 0 | 1 | 0 | 0 | 0 | 17 | 0 | 7 |
| A2 | 0 | 51 | 1 | 0 | 0 | 4 | 0 | 0 |
| A3 | 90 | 11 | 103 | 1 | 0 | 0 | 1 | 0 |
| A4 | 5 | 13 | 4 | 0 | 0 | 14 | 0 | 0 |
| A5 | 0 | 0 | 0 | 64 | 1 | 0 | 0 | 0 |
| A6 | 0 | 0 | 0 | 0 | 33 | 0 | 0 | 0 |
| A7 | 0 | 3 | 0 | 0 | 0 | 43 | 0 | 4 |
| A8 | 0 | 2 | 0 | 0 | 0 | 2 | 45 | 1 |
| A9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 51 |

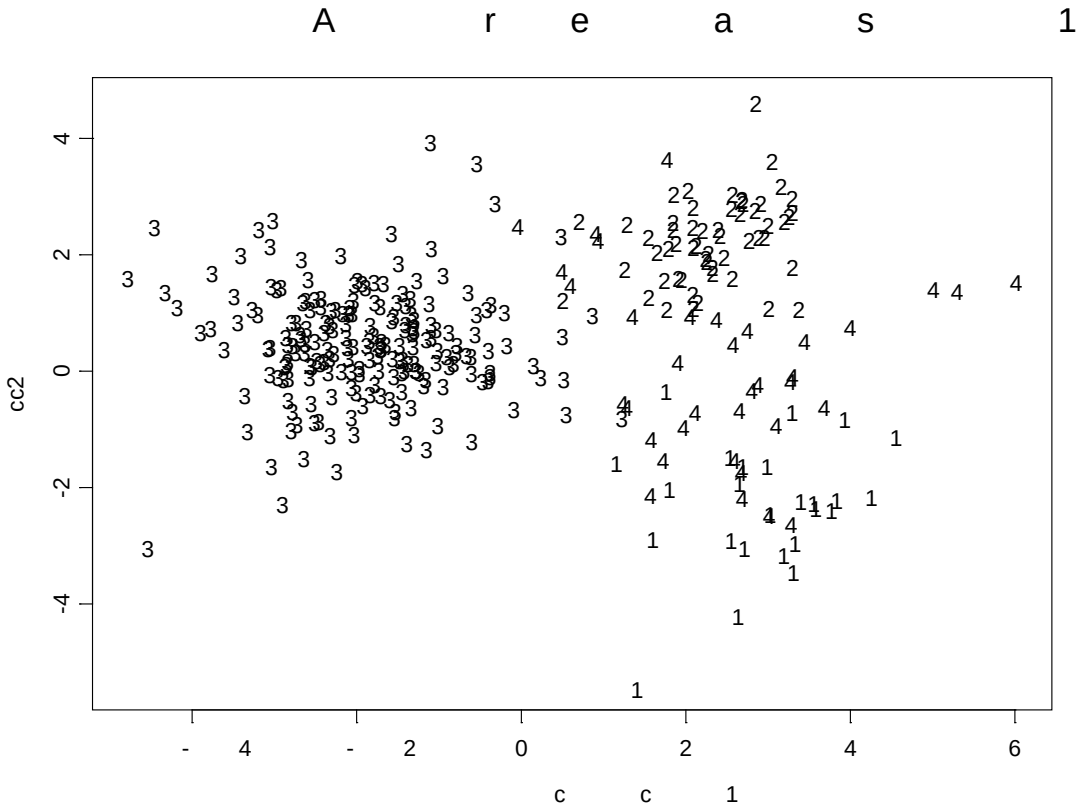
Note able to recognize Areas 1 and 4 in Region 1.

Splits up Area 3.

# How well are Areas 1-4 separated?

Draw projection onto first two discriminant coordinates.

Note: Only possible because we know labels.



# 7. Current research

## 7.1 Computing the cluster tree for other density estimates

**Idea:** Approximate geometric problem of finding connected components of level sets by graph problem.

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

Let $G$ be the complete graph over $\underline{x}_1, \ldots, \underline{x}_n$ with edge weights $\hat{p}_{ij}$ and vertex weights $\hat{p}_i$.

Let $G(\lambda)$ be the threshold graph obtained from $G$ by removing edges with $\hat{p}_{ij} \leq \lambda$ and vertices with $\hat{p}_i \leq \lambda$.

Apply recursive operations of thresholding and finding connected components to the graph $G$ instead of feature space.

## Motivation:

Observations in the same connected component of $G(\lambda)$ are in the same connected component of $L(\lambda; \hat{p})$.

Observations in different connected components of $G(\lambda)$ might be in the same connected component of $L(\lambda; \hat{p})$, but if $\hat{p}$ is smooth this is unlikely.

## 7.2 Clustering with confidence

Problem: Density estimates may have spurious modes due to sampling variability.

Idea:

Compute Bootstrap density estimates.

Connected components present in most Boostrap estimates are probably "real".

Note enough time to give details.

# 8. Summary

The term "clustering" is ambiguous; need to distinguish between *dissection* and *concept formation*.

Goal of concept formation: detect presence of distinct groups.

Premise of nonparametric clustering: groups $\sim$ modes of feature density.

Structure of collection of level sets is described by cluster tree; modes $\sim$ leaves.

Cluster tree is defined recursively — suggests recursive partitioning method for its computation.

For some density estimates, cluster tree can be computed exactly.

For others, cluster tree has to be approximated by solution of a graph problem.

May be able to use Bootstrap for distinguishing between "real" and "spurious" modes.

Thanks for your interest.

Cluster tree or density estimate may have spurious modes and will have to be pruned.

Boostrapping may offer a way of deciding which branches are "real". Power?

# Thank you for your patience

## General goals of research

Development of nonparametric methods for concept formation:

- Basics — what are we trying to estimate?
- Estimation methods and algorithms
- Diagnostics
- Theory
- Extension to other domains (discrete data, graphs)

Domain-specific adaptation — clustering microarray data, topic detection and tracking.

## Primary goal at the moment

Develop method for estimating the number of groups.

# 2. Why are we interested in concept formation

**General answer:**

Concept formation ("unsupervised learning") is an important component of cognition.

Most of human learning is unsupervised (semi-supervised?)

**Specific example:**

Generating a taxonomy of diseases from gene expression data.

## Generating a taxonomy of diseases from gene expression data

DNA microarrays allow simultaneous measurement of expression levels for 1000's of genes.

Patterns of expression might allow discovery and prediction of new disease classes

- Independent of previous biological knowledge, and
- In the absence of clearly distinct clinical symptom patterns.

Successful differentiation would allow class specific treatment that might improve treatment success.

Example: Data from Golub et al (Science, 1999)

3051 genes

Tumor RNA samples from 38 leukemia patients:

- 27 acute lymphoblastic leukemia (ALL) cases
- 11 acute myeloid leukemia (AML) cases

**Questions:**

- Can we find the two groups without using the labels?
- Are there other groups?

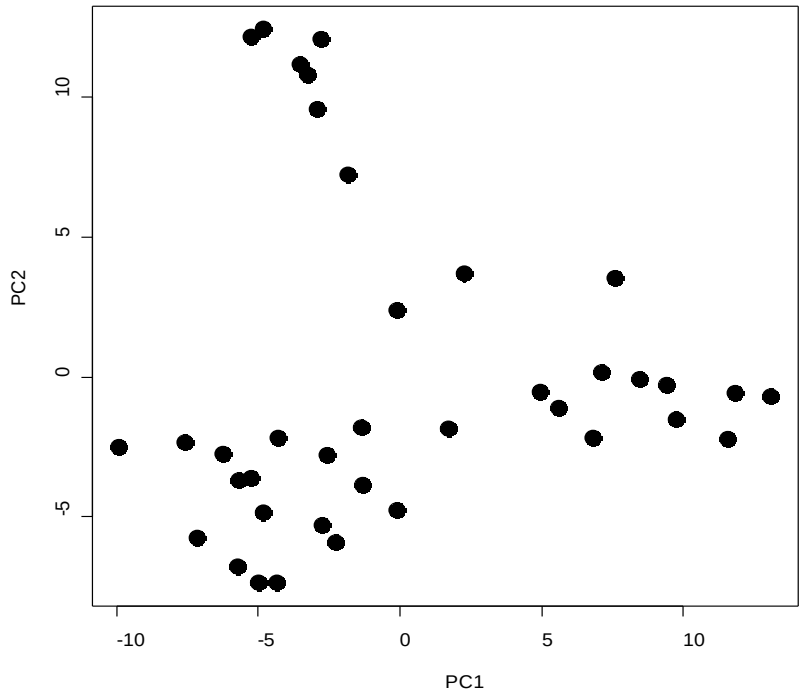
Seems like a challenging problem — 38 points in 3051-dimensional space.

Not impossible, though!

## Sample analysis

Choose 500 genes with highest values of (mean * standard deviation).

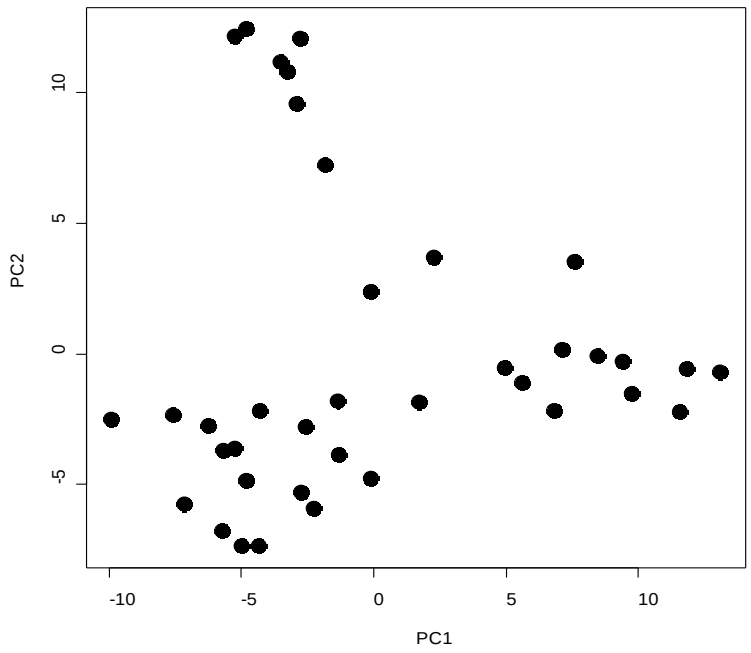
Project onto two largest principal components.

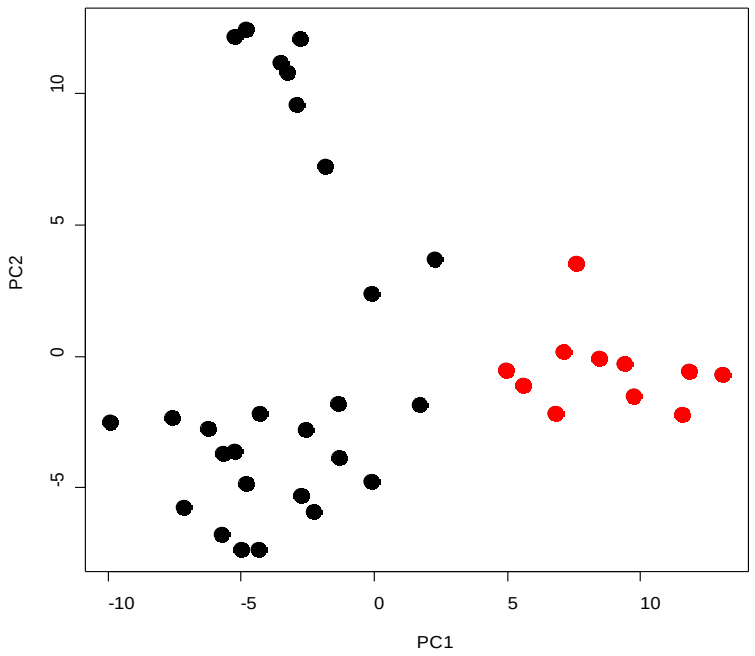


## Sample analysis

Choose 500 genes with highest values of (mean * standard deviation).

Project onto two largest principal components.




11 AML samples highlighted

## Dissection:

**Given:** Collection of $n$ objects characterized by feature vectors $\underline{x}_1, \ldots, \underline{x}_n$.

**Goal:** Partition collection into compact subsets.

The prototypical dissection method: *K-means clustering*.

Let $\mathcal{P}_k = P_1, \ldots, P_k$ be a partition of the objects into $k$ groups.

Measure badness of partition by sum of squared distances of observations from their group means:

$$\mathbf{rss}\left(\mathcal{P}_k\right) = \sum_{i=1}^{k} \sum_{j \in P_i} \left\| \underline{x}_j - \bar{\underline{x}}_i \right\|^2 .$$

Find optimal partition (for example with the Lloyd algorithm).

```
generate_cluster_tree_node (mst, runt_size_threshold) {
    node = new_cluster_tree_node
    node.leftson = node.rightson = NULL
    node.obs = leaves (mst)
    cut_edge = longest_edge_with_large_runt_size (mst, runt_size_threshold)
    if (cut_edge) {
        node.leftson = generate_cluster_tree_node (left_subtree (cut_edge), runt_size_threshold)
        node.rightson = generate_cluster_tree_node (right_subtree (cut_edge), runt_size_threshold)
    }
    return (node)
}
```