Hierarchical clustering

Rebecca C. Steorts, Duke University

STA 325, Chapter 10 ISL

globally set figure width and height and cache

```
knitr::opts_chunk$set(fig.width=5, fig.height=4,
                      cache=TRUE)
```

Agenda

- K-means versus Hierarchical clustering
- Agglomerative vs divisive clustering
- Dendogram (tree)
- Hierarchical clustering algorithm
- Single, Complete, and Average linkage

From K-means to Hierarchical clustering

Recall two properties of K-means clustering:

1. It fits exactly K clusters (as specified)
2. Final clustering assignment depends on the chosen initial cluster centers

- Assume pairwise dissimilarites d_{ij} between data points.
- Hierarchical clustering produces a consistent result, without the need to choose initial starting positions (number of clusters).

Catch: choose a way to measure the dissimilarity between groups, called the linkage

- Given the linkage, hierarchical clustering produces a sequence of clustering assignments.
- At one end, all points are in their own cluster, at the other end, all points are in one cluster

Agglomerative vs divisive clustering

Agglomerative (i.e., bottom-up):

- Start with all points in their own group
- Until there is only one cluster, repeatedly: merge the two groups that have the smallest dissimilarity

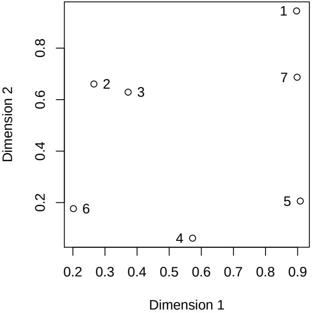
Divisive (i.e., top-down):

- Start with all points in one cluster
- Until all points are in their own cluster, repeatedly: split the group into two resulting in the biggest dissimilarity

Agglomerative strategies are simpler, we'll focus on them. Divisive methods are still important, but you can read about these on your own if you want to learn more.

Simple example

Given these data points, an agglomerative algorithm might decide on a clustering sequence as follows:



Step 1: $\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\}$;

Step 2: $\{1\},\{2,3\},\{4\},\{5\},\{6\},\{7\}$;

Step 3: $\{1,7\},\{2,3\},\{4\},\{5\},\{6\}$;

Step 4: $\{1,7\},\{2,3\},\{4,5\},\{6\}$;

Step 5: $\{1,7\},\{2,3,6\},\{4,5\}$;

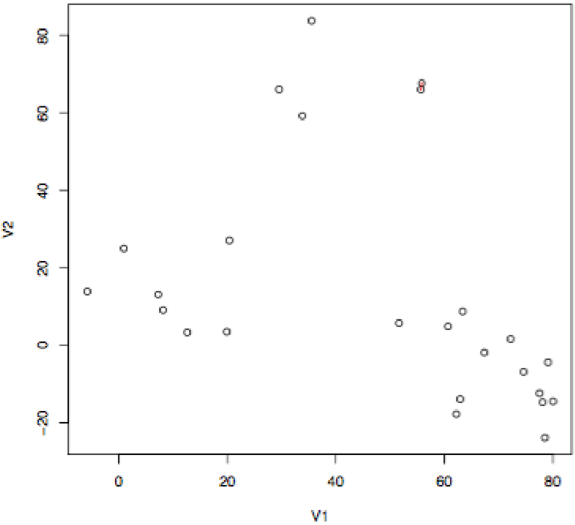
Step 6: $\{1,7\},\{2,3,4,5,6\}$;

Step 7: $\{1,2,3,4,5,6,7\}$.

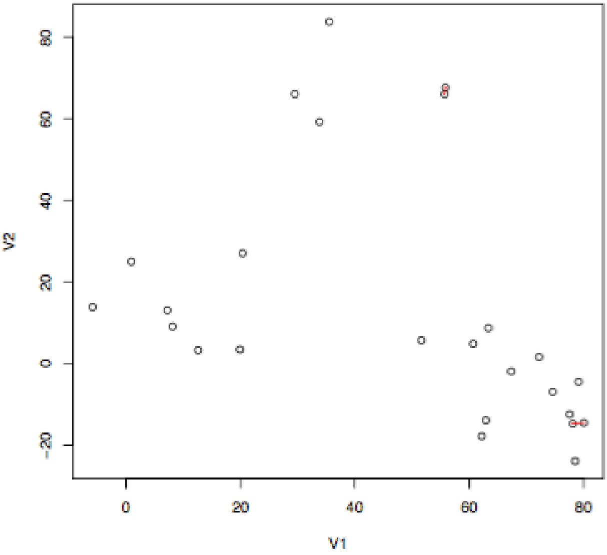
Algorithm

1. Place each data point into its own singleton group.
2. Repeat: iteratively merge the two closest groups
3. Until: all the data are merged into a single cluster

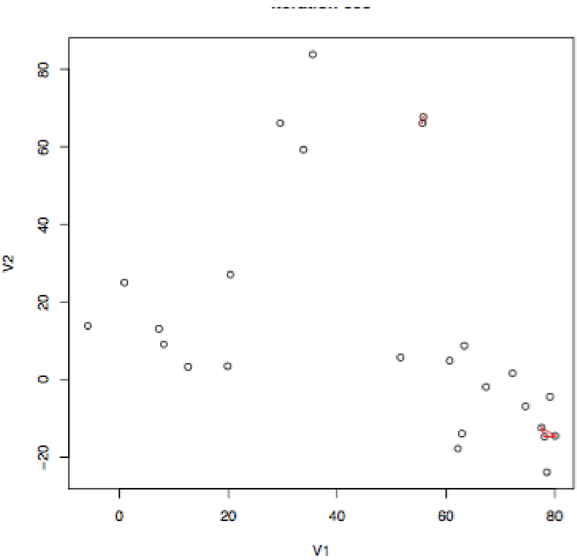
Example



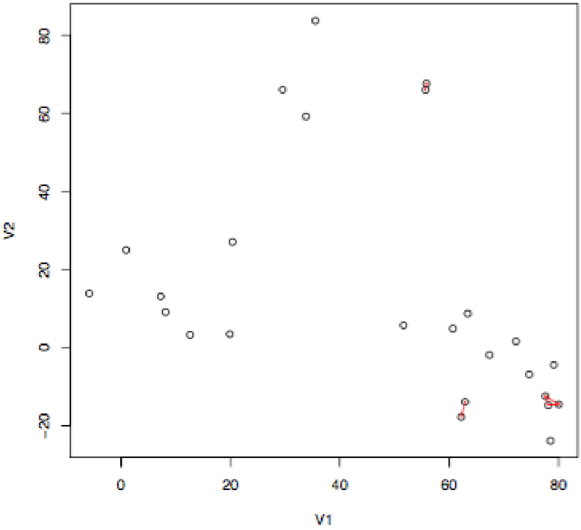
Iteration 2



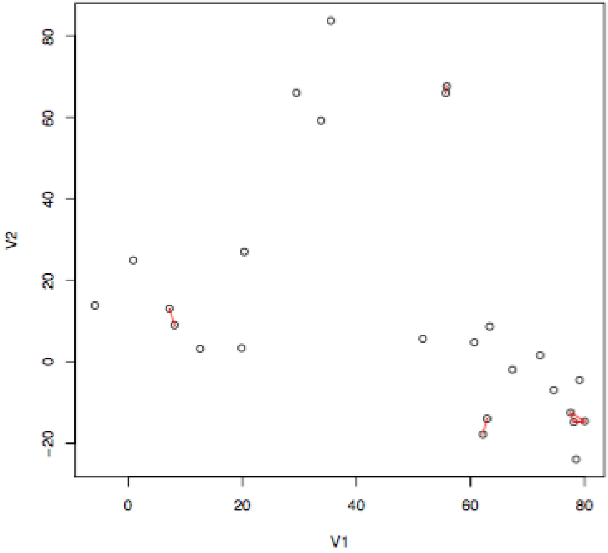
Iteration 3



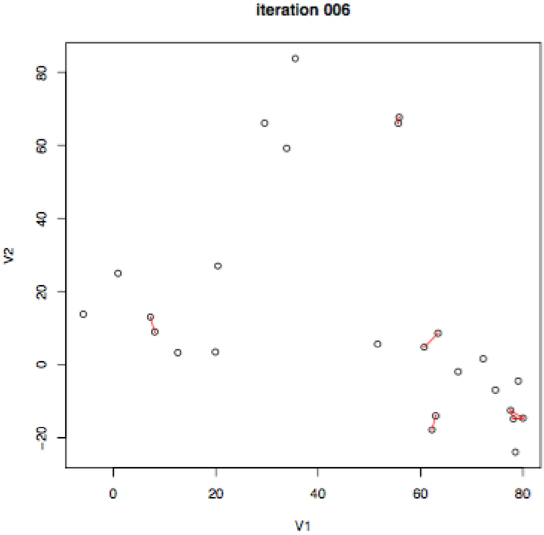
Iteration 4



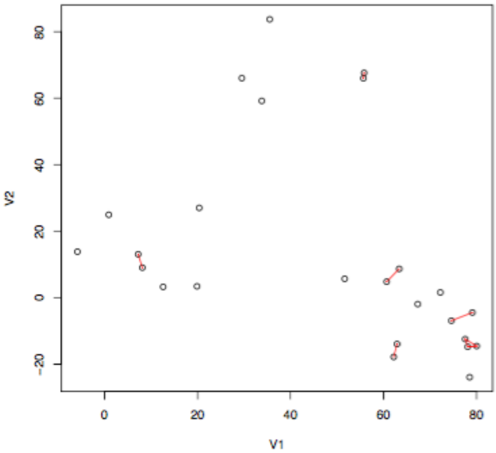
Iteration 5



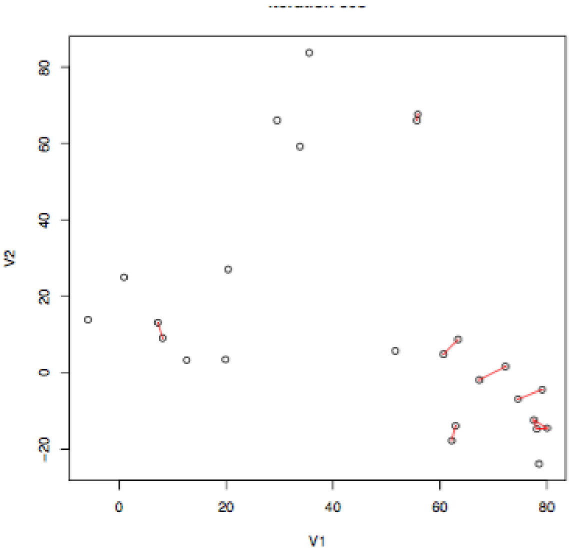
Iteration 6



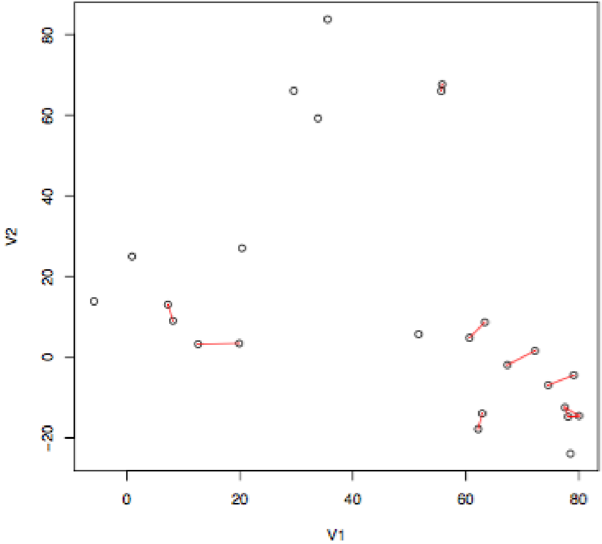
Iteration 7



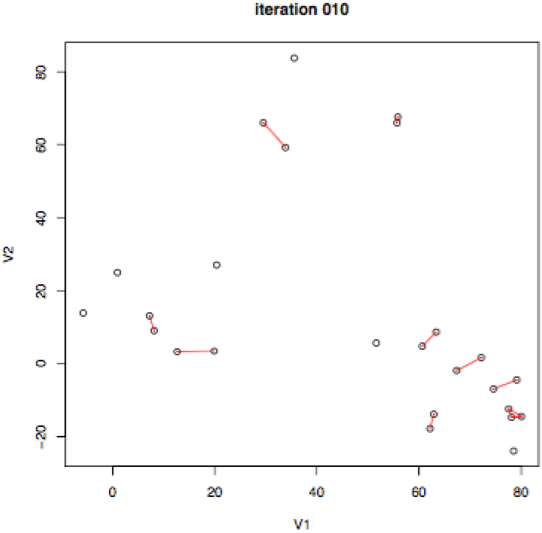
Iteration 8



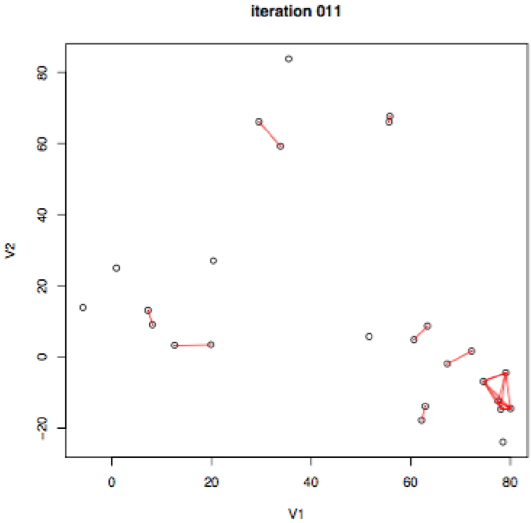
Iteration 9



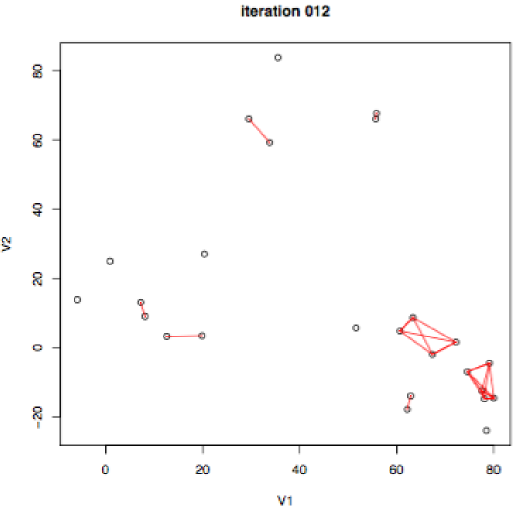
Iteration 10



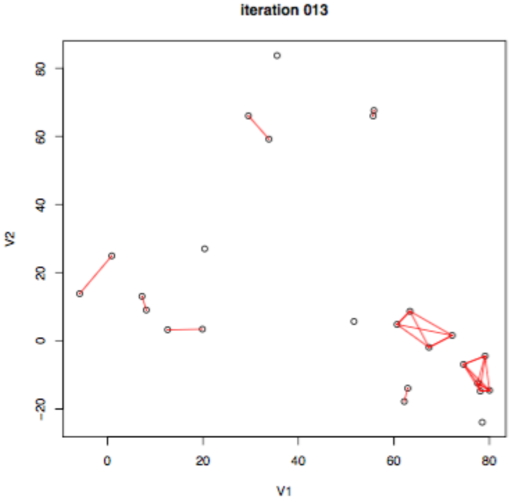
Iteration 11



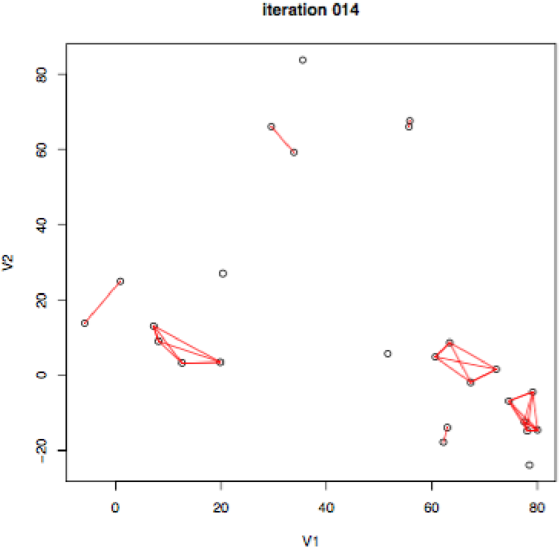
Iteration 12



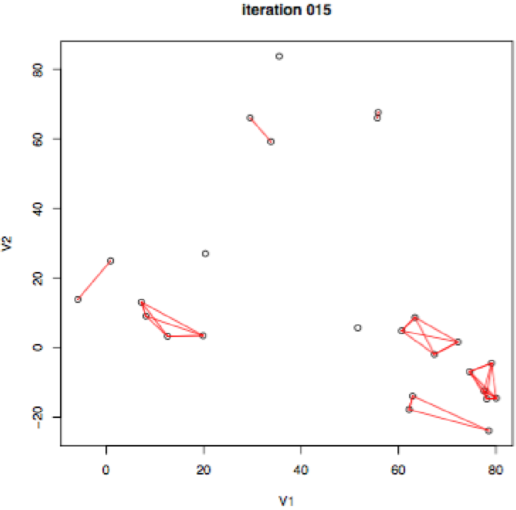
Iteration 13



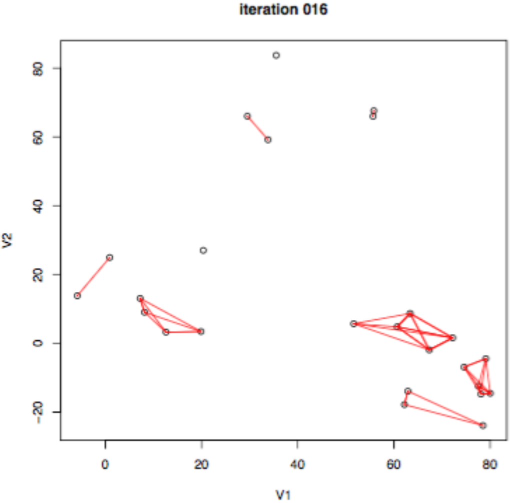
Iteration 14



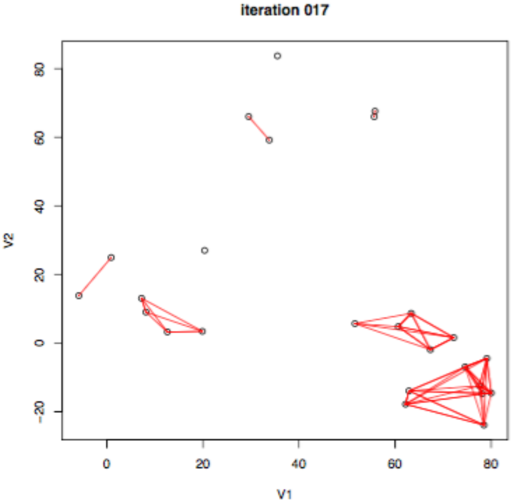
Iteration 15



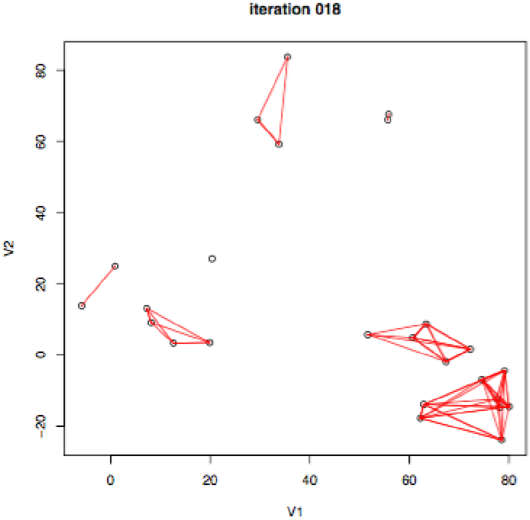
Iteration 16



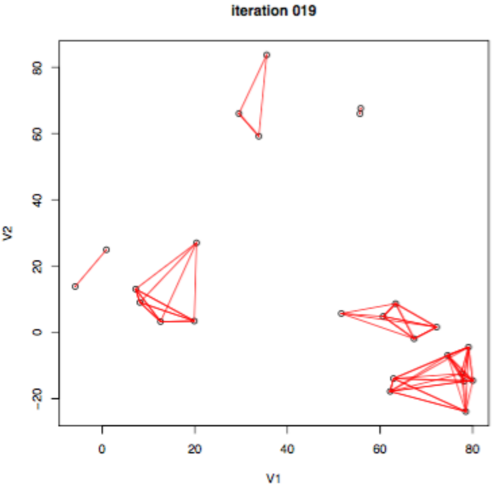
Iteration 17



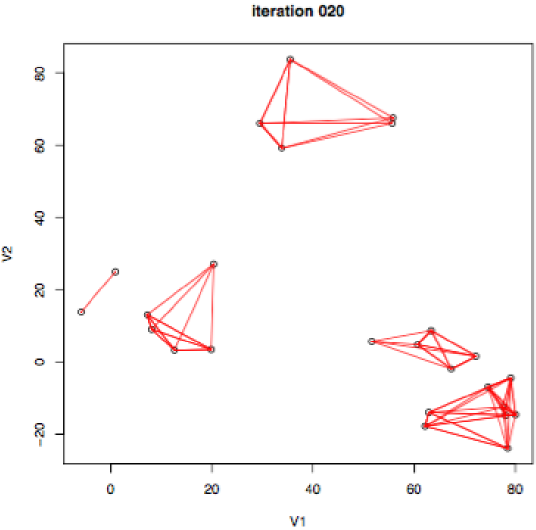
Iteration 18



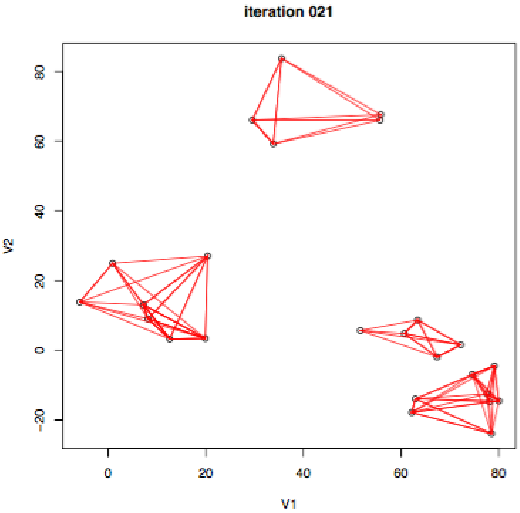
Iteration 19



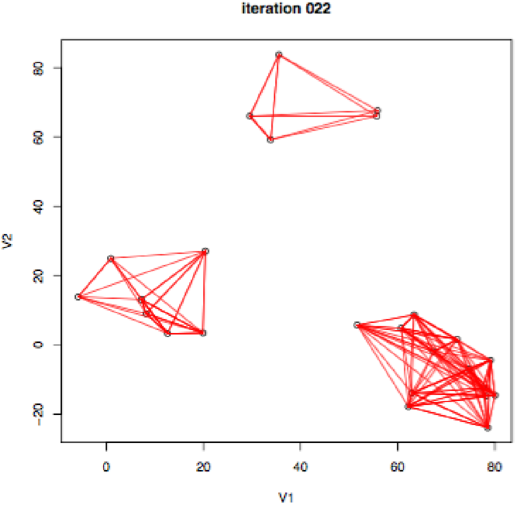
Iteration 20



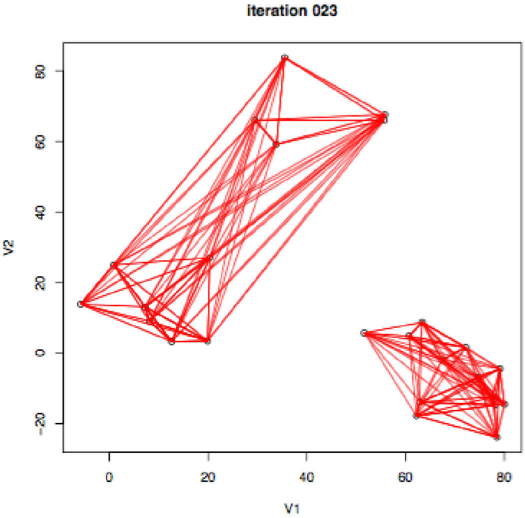
Iteration 21



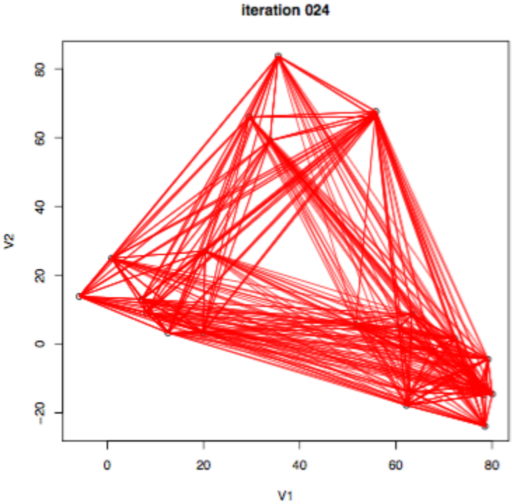
Iteration 22



Iteration 23



Iteration 24



Clustering

Suppose you are using the above algorithm to cluster the data points in groups.

- How do you know when to stop?
- How should we compare the data points?

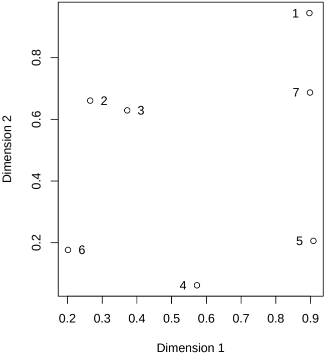
Let's investigate this further!

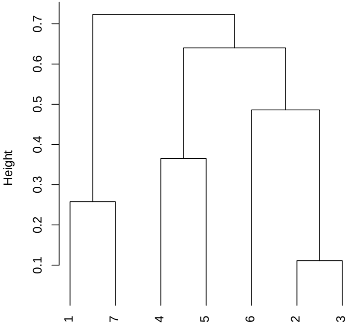
Agglomerative clustering

- Each level of the resulting tree is a segmentation of the data
- The algorithm results in a sequence of groupings
- It is up to the user to choose a “natural” clustering from this sequence

Dendogram

We can also represent the sequence of clustering assignments as a dendrogram:





Note that cutting the dendrogram horizontally partitions the data points into clusters

Dendogram

- Agglomerative clustering is monotonic
- The similarity between merged clusters is monotone decreasing with the level of the merge.[1]
- Dendrogram: Plot each merge at the (negative) similarity between the two merged groups
- Provides an interpretable visualization of the algorithm and data
- Useful summarization tool, part of why hierarchical clustering is popular

[1] A function that is increasing or decreasing at some point is called monotone at that point.

Group similarity

Given a distance similarity measure (say, Eucliclean) between points, the user has many choices on how to define intergroup similarity.

1. Single linkage: the similiarity of the closest pair

$$d_{SL}(G,H) = \min_{i \in G, j \in H} d_{i,j}$$

2. Complete linkage: the similarity of the furthest pair

$$d_{CL}(G,H) = \max_{i \in G, j \in H} d_{i,j}$$

3. Group-average: the average similarity between groups

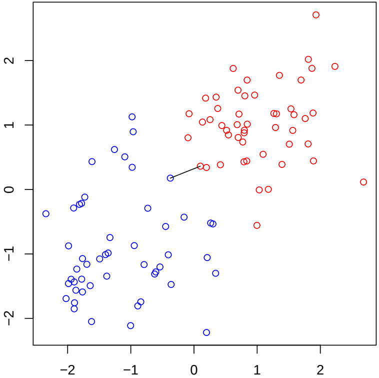
$$d_{GA} = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{j \in H} d_{i,j}$$

Single Linkage

In single linkage (i.e., nearest-neighbor linkage), the dissimilarity between G, H is the smallest dissimilarity between two points in opposite groups:

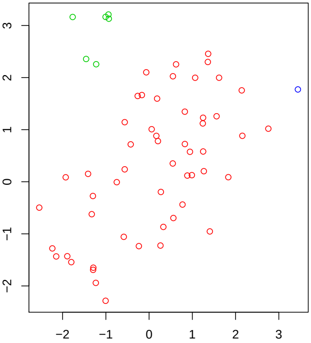
$$d_{\text{single}}(G, H) = \min_{i \in G, j \in H} d_{ij}$$

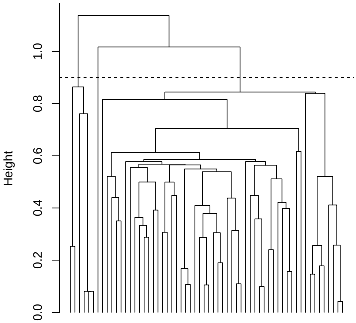
Example (dissimilarities d_{ij} are distances, groups are marked by colors): single linkage score $d_{\text{single}}(G, H)$ is the distance of the closest pair



Single Linkage Example

Here $n = 60$, $X_i \in \mathbb{R}^2$, $d_{ij} = \|X_i - X_j\|_2$. Cutting the tree at $h = 0.9$ gives the clustering assignments marked by colors





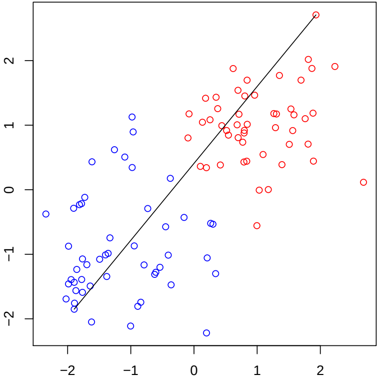
Cut interpretation: for each point X_i, there is another point X_j in its cluster with $d_{ij} \leq 0.9$

Complete Linkage

In complete linkage (i.e., furthest-neighbor linkage), dissimilarity between G, H is the largest dissimilarity between two points in opposite groups:

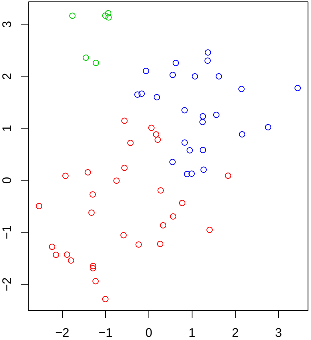
$$d_{\text{complete}}(G, H) = \max_{i \in G,\, j \in H} d_{ij}$$

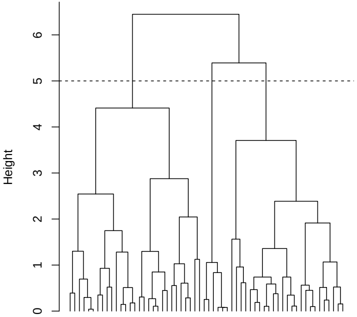
Example (dissimilarities d_{ij} are distances, groups are marked by colors): complete linkage score $d_{\text{complete}}(G, H)$ is the distance of the furthest pair



Complete Linkage Example

Same data as before. Cutting the tree at $h = 5$ gives the clustering assignments marked by colors





Cut interpretation: for each point X_i, every other point X_j in its cluster satisfies $d_{ij} \leq 5$

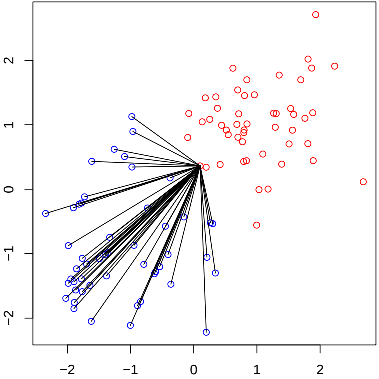
Average Linkage

In average linkage, the dissimilarity between G, H is the average dissimilarity over all points in opposite groups:

$$d_{\text{average}}(G,H) = \frac{1}{n_G \cdot n_H} \sum_{i \in G,\, j \in H} d_{ij}$$

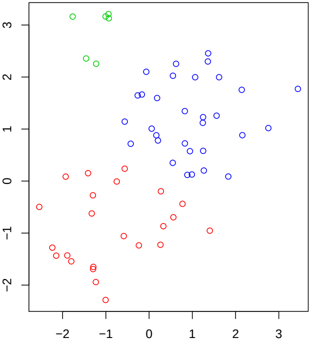
Example (dissimilarities d_{ij} are distances, groups are marked by colors): average linkage score $d_{\text{average}}(G,H)$ is the average distance across all pairs

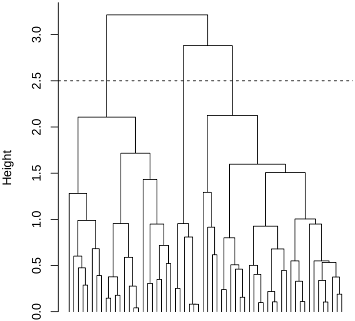
(Plot here only shows distances between the blue points and one red point)



Average linkage example

Same data as before. Cutting the tree at $h = 2.5$ gives clustering assignments marked by the colors





Cut interpretation: there really isn't a good one!

Properties of intergroup similarity

- Single linkage can produce “chaining,” where a sequence of close observations in different groups cause early merges of those groups
- Complete linkage has the opposite problem. It might not merge close groups because of outlier members that are far apart.
- Group average represents a natural compromise, but depends on the scale of the similarities. Applying a monotone transformation to the similarities can change the results.

Things to consider

- Hierarchical clustering should be treated with caution.
- Different decisions about group similarities can lead to vastly different dendrograms.
- The algorithm imposes a hierarchical structure on the data, even data for which such structure is not appropriate.