

# Data Mining: Hierarchical Clustering

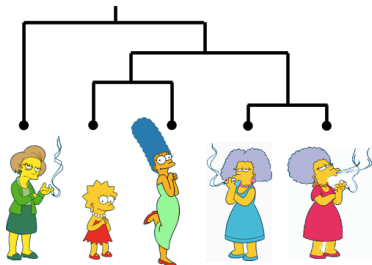
cs4821-cs5831

Some slides adapted from P. Smyth; A. Moore, D. Klein Han,  
Kamber, Pei; Tan, Steinbach, Kumar; L. Kaebling; R. Tibshirani;  
T. Taylor; and L. Hannah

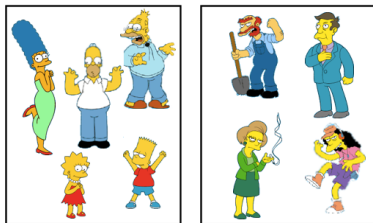
# Two Main Types of Clustering

- Partitional methods: construct partitions of the data using some criteria; may be hard or fuzzy partitions
- Hierarchical methods: create hierarchical decomposition of data using some criteria

## Hierarchical

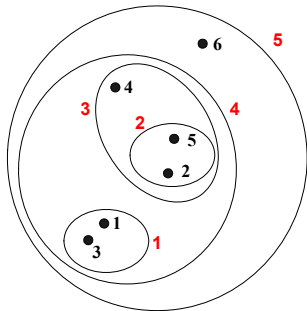
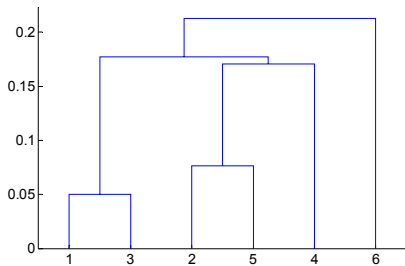


## Partitional



# Hierarchical Clustering

- Produce a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree-like diagram that records the sequences of merges or splits



# Dendrograms

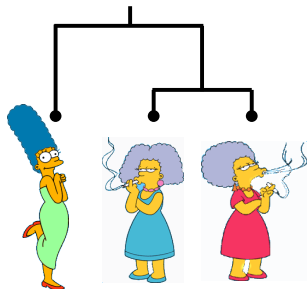
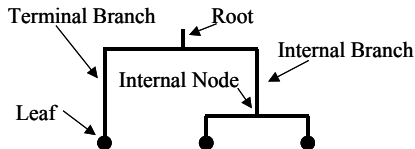
- The dendrogram provides a nested partitioning (tree of clusters)
- A single clustering of the objects can be obtained by cutting the dendrogram at a desired level, then each connected component forms a cluster

A dendrogram is a tree:

- Each node represents a group
- Each leaf node is a singleton (group of a single data point)
- Root node is the group containing the whole data set
- Each internal node has two child nodes, representing the groups that were merged to form it

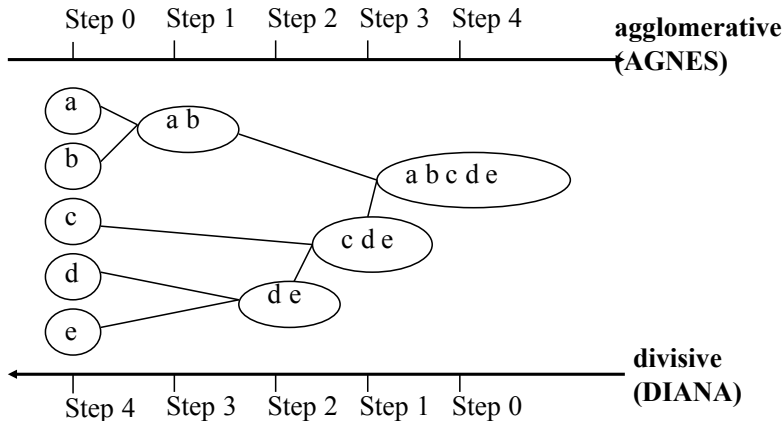
# Dendrograms

- A useful tool for summarizing similarity measurements
- The similarity between two objects in a dendrogram is represented as the height of the lowest internal node they share
  - If the leaf nodes are fixed at height zero, the internal nodes can be drawn at a height proportional to the dissimilarity between the two child nodes



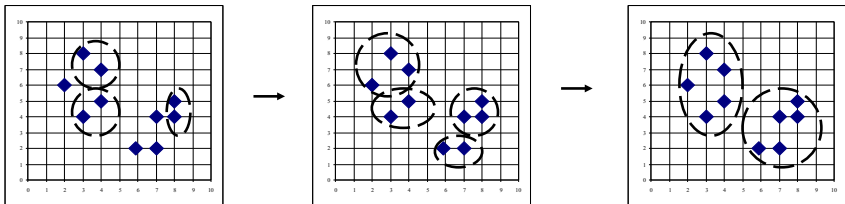
# Hierarchical Clustering

- Use distance matrix as clustering criteria. Method does not require the number of clusters,  $k$ , as input; but does need stopping criteria and selection of other parameters



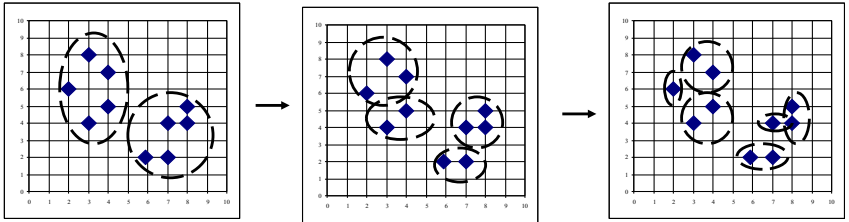
# AGNES (Agglomerative Nesting)

- Introduced in Kaufman and Rousseeuw, '90
- Used single-link method and the dissimilarity matrix
- Merge nodes that have least dissimilarity
- Repeat in a non-descending fashion
- Eventually all nodes belong to the same cluster



# DIANA (Divisive Analysis)

- Introduced in Kaufman and Rousseeuw, '90
- Inverse order of AGNES
- Eventually each node forms a cluster on its own





# A Challenge of Hierarchical Clustering

What dendrogram (or clustering) is best?

The number of dendrograms with  $n$  leafs is

$$\frac{(2n - 3)!}{2^{n-1}(n - 2)!}$$

NumLeafs	NumDendrograms
2	1
3	3
4	15
5	105
10	34459425

Conclusion: We can not consider all possible trees, use heuristic search to find local optima

# Agglomerative vs. Divisive

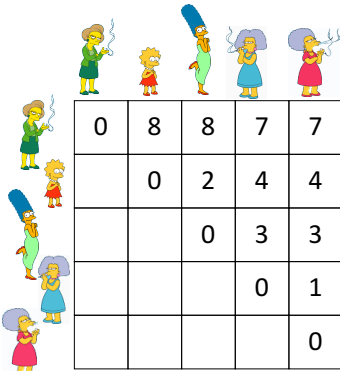
- Agglomerative (bottom-up)
  - Start with all points in their own group
  - Until there is only one cluster, repeatedly: merge the two groups that have the smallest dissimilarity (best pair to merge)
- Divisive (top-down)
  - Start with all points in one cluster
  - Until all points are in their own cluster, repeatedly: split the group into two resulting in the biggest dissimilarity (best group to divide)
- Comments:
  - Agglomerative are simpler methods






# Example of Hierarchical Clustering

Hierarchical clustering methods typically runs on a dissimilarity / distance matrix  $D$ , where  $d_{ij}$  is the dissimilarity between data points  $x_i$  and  $x_j$ .


$$D(\text{Marge}, \text{Bart}) = 8$$


$$D(\text{Lisa}, \text{Maggie}) = 1$$

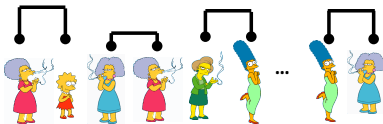


				
0	8	8	7	7
	0	2	4	4
		0	3	3
			0	1
				0

# Example of Hierarchical Clustering

**Bottom-Up (agglomerative)** Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

Consider all  
possible  
merges...



Choose  
the best

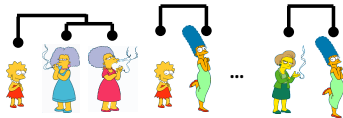


Slide from Eamonn Keogh (eamonn@cs.ucr.edu)

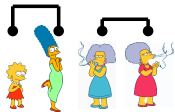
# Example of Hierarchical Clustering

**Bottom-Up (agglomerative)** Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

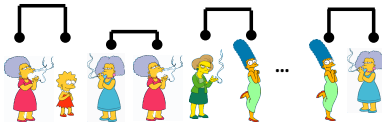
Consider all possible merges...



Choose the best



Consider all possible merges...



Choose the best



Slide from Eamonn Keogh (eamonn@cs.ucr.edu)

# Linkage Techniques

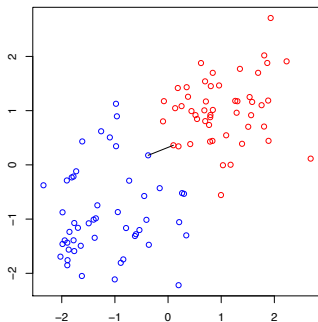
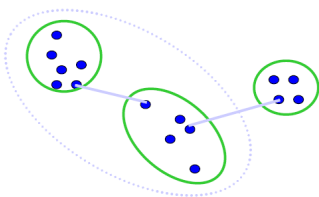
- We know how to measure the distance between two objects
  - dissimilarity measure
- How do we define the distance between an object and a cluster?
  - between two clusters?

Methods to solve this task are called **linkages**

# Single Linkage

**Single linkage** (nearest neighbor linkage) - the dissimilarity between two cluster  $G$ ,  $H$  is the smallest dissimilarity between two data objects in different clusters (the dissimilarity of the two “closest” objects in the different clusters)

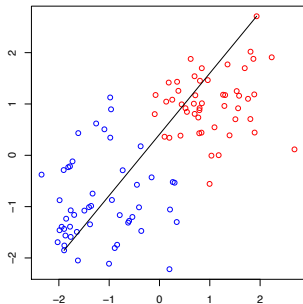
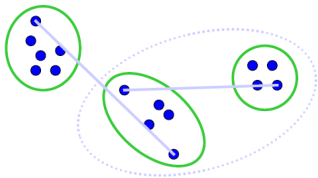
$$d_{single}(G, H) = \min_{i \in G, j \in H} d_{ij}$$



# Complete Linkage

**Complete linkage** (furthest neighbor linkage) - the dissimilarity between two cluster  $G$ ,  $H$  is the largest dissimilarity between two data objects in different clusters (the dissimilarity of the two “most separate” objects in the different clusters)

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

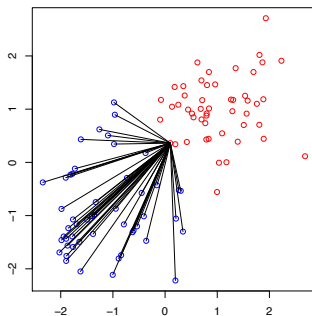
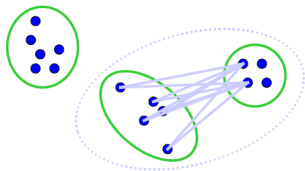




# Average Linkage

**Average linkage** - the dissimilarity between two cluster  $G$ ,  $H$  is calculated as the average dissimilarity between all pairs of objects in the two clusters

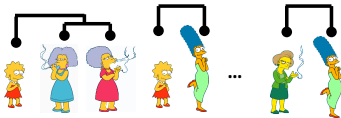
$$d_{average}(G, H) = \frac{1}{n_G \cdot n_H} \sum_{i \in G, j \in H} d_{ij}$$



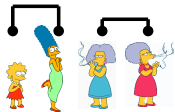
# Example of Hierarchical Clustering

**Bottom-Up (agglomerative)** Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

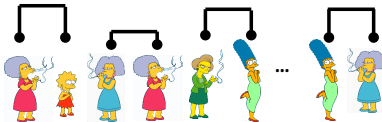
Consider all possible merges...



Choose the best



Consider all possible merges...



Choose the best

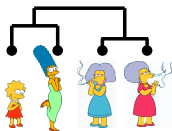


Slide from Eamonn Keogh (eamonn@cs.ucr.edu)

# Example of Hierarchical Clustering

**Bottom-Up (agglomerative)** Starting with each item in its own cluster, find the best pair to merge into a new cluster. Repeat until all clusters are fused together.

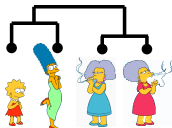
Consider all possible merges...



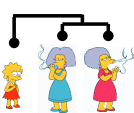
...



Choose the best



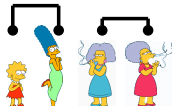
Consider all possible merges...



...



Choose the best



Consider all possible merges...



...



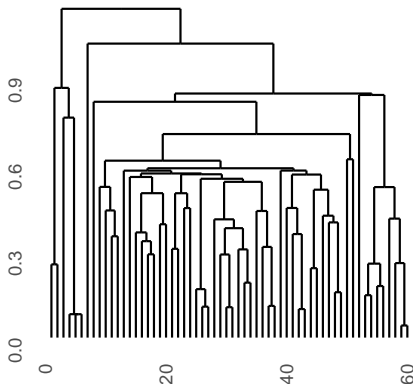
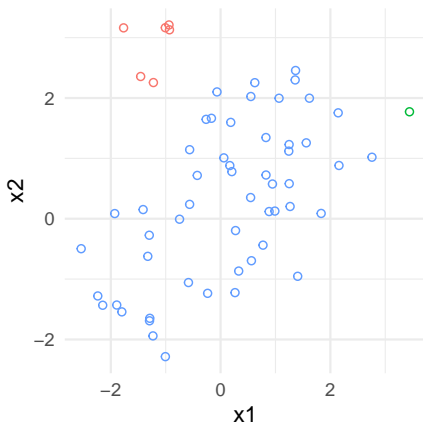
Choose the best



Slide from Eamonn Keogh (eamonn@cs.ucr.edu)

## Example 2: Single Linkage

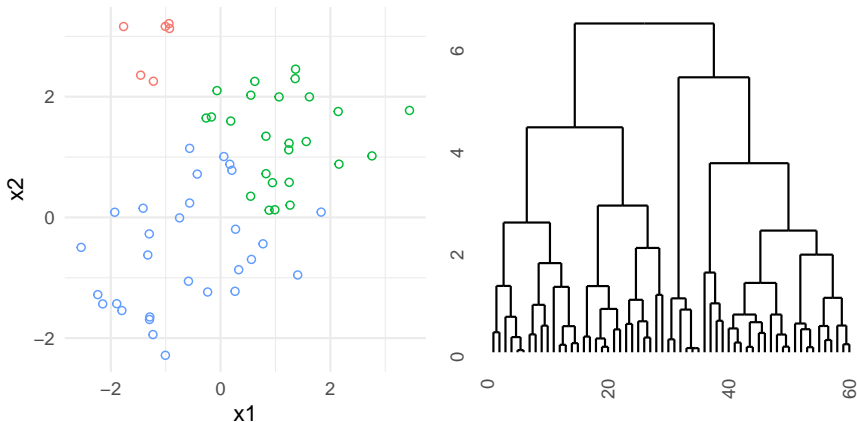
Example:  $n=60$ ,  $X_i \in \mathbb{R}^2$ ,  $d_{ij} = \|X_i - X_j\|_2$ . If the tree is cut at  $y=0.9$ , three clusters formed (shown by colors)



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

## Example 2: Complete Linkage

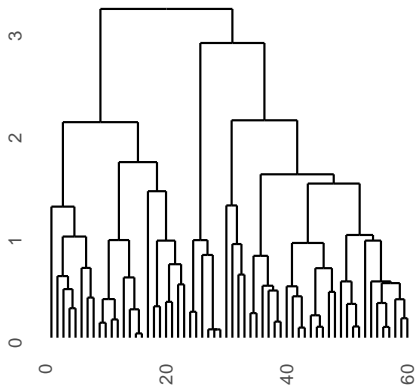
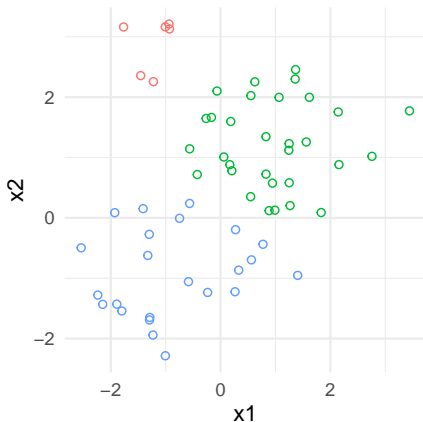
Same data as before; cut tree at  $y = 5$  for three clusters.



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

## Example 2: Average Linkage

Same data as before; cut tree at  $y = 2.5$  for three clusters.



Cut Interpretation: does not exist

# Common Linkage Properties

Single, complete, and average linkage share the following properties:

- The linkage methods work on dissimilarities,  $d_{ij}$ , rather than the original data; data points do not need to be in Euclidean space
- Running the agglomerative clustering with the linkage results in a dendrogram with **no inversions**
  - The dissimilarity scores between merged clusters only increases; in the dendrogram, the height of a parent is always higher than the height of its children

# Weaknesses of Single, Complete Linkage

Single and complete linkage have the following problems:

- Single linkage suffers from **chaining**  
To merge two groups, only need one pair of points to be close (ignoring all others).
- Complete linkage suffers from **crowding**  
Worst-case dissimilarity is used, therefore a point can be closer to points in other clusters than to points in its own cluster.

Average linkage tries to balance between these two, so that clusters can be compact and spread out.



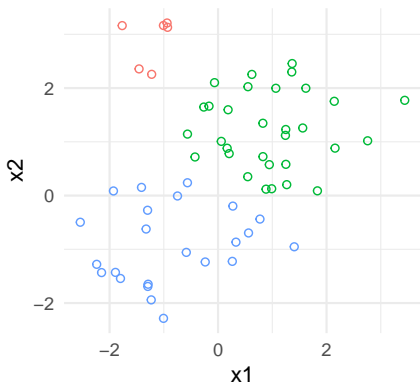
# Issues with Average Linkage

Average linkage also has problems:

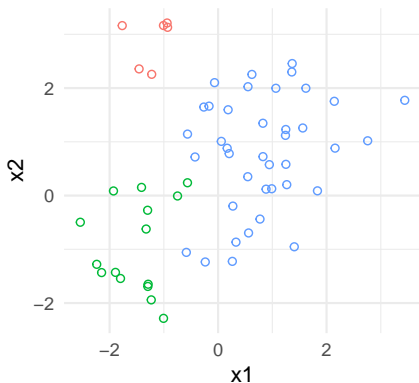
- Both single and complete linkages have easily interpretations of relations among points when the tree is cut at height  $h$ ; average linkage has no such interpretation.
- Results of average linkage can change with a monotone increasing transformation of dissimilarities.
  - Results of single and complete linkage clustering are unchanged under monotone transformations.

# Example of Monotone Increasing Transformation

Ave Linkage: distance



Ave Linkage: distance<sup>2</sup>



# Linkage Review

We have learned about hierarchical agglomerative clustering  
Basic idea, repeatedly merge two most similar groups - measured by linkage

Three linkages: **single**, **complete**, **average** linkage

- **single** and **complete** linkage can have problems with chaining and crowding respectively, **average** linkage does not
- cutting an **average** dendrogram has no interpretation, but interpretations are available for **single** and **complete**
- **average** linkage is sensitive to a monotone transformation of the dissimilarities, but **single** and **complete** are not
- all three produce dendrograms with out inversions

# Linkage Review

Given data points  $X_1, \dots, X_n$  and pairwise dissimilarities  $d_{ij}$

**Single Linkage:** measures the closest pair of points

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

**Complete Linkage:** measures the farthest pair of points

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

**Average Linkage:** measures the average dissimilarity over all pairs

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

# Hierarchical clustering in R, Python

R:

```
1 d = dist(x)
2 tree.avg = hclust(d, method="average")
3 #plot(tree.avg)
```

Python:

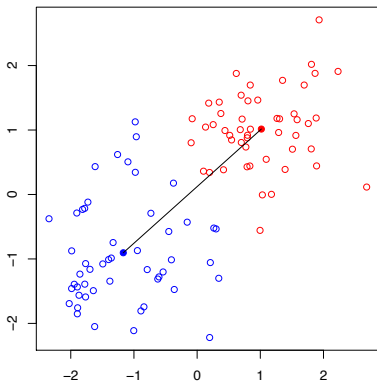
```
1 # using sklearn library
2 model = cluster.AgglomerativeClustering(distance_threshold=0,
3                                         n_clusters=None,
4                                         linkage="single")
5 model.fit(x)
6 # using scipy library
7 Z = hierarchy.linkage(dis1, 'single')
8 dn = hierarchy.dendrogram(Z)
```

# Centroid Linkage

**Centroid linkage.** Let  $\bar{x}_G, \bar{x}_H$  denote the group averages for clusters  $G, H$ .

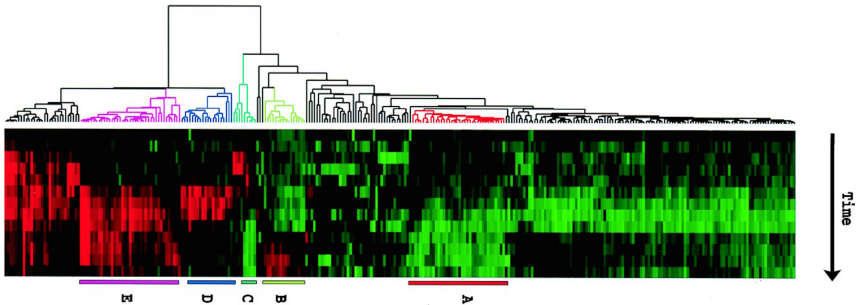
$$d_{centroid}(G, H) = \|\bar{x}_G - \bar{x}_H\|_2$$

Group  $G$  are the blue points, Group  $H$  the red.  
centroid linkage score  
 $d_{centroid}(G, H)$  is the  
distance between the group  
centroids (i.e., group  
averages)



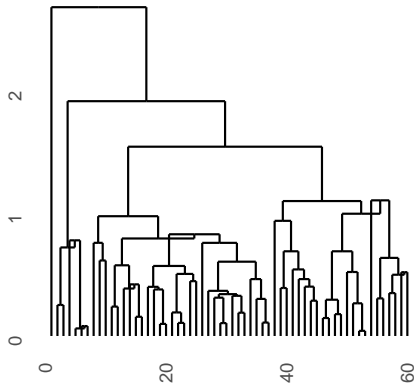
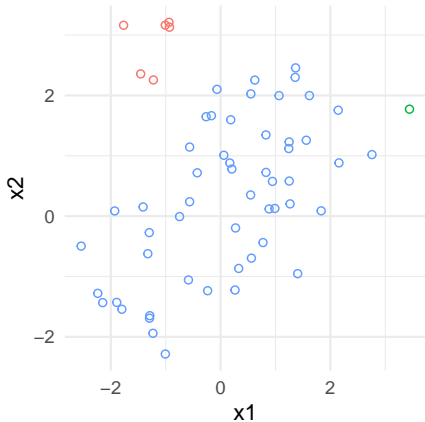
# Examples of Centroid Linkage

**Centroid** linkage is simple, easy to understand, and easy to implement. Used widely in biology. (Eisen et al., PNAS 1998 )



## Example 2: Centroid Linkage

Example:  $n=60$ ,  $x_i \in \mathbb{R}^2$ ,  $d_{ij} = \|x_i - x_j\|_2$ . If the tree is cut at  $y=1.7$ , three clusters formed (shown by colors). Note, the dendrogram can not be cut at all heights due to inversions



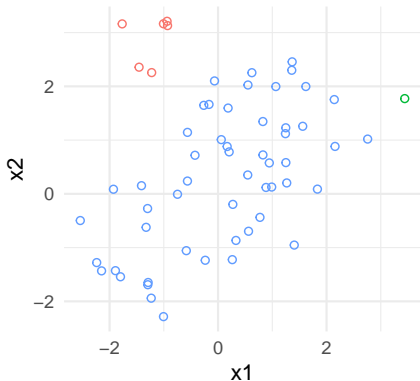
Cut Interpretation: none



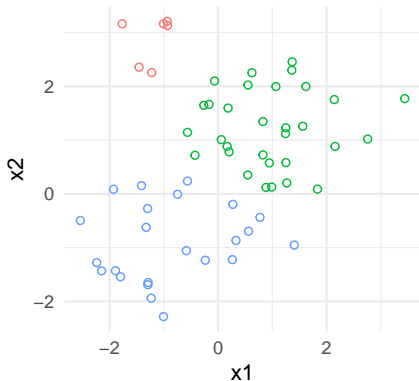
# Issues with centroid linkage

- Dendrograms may have inversions
- No interpretation for the clusters resulting from cutting the tree in partitions
- Monotone transformations can change the clustering results

Centroid Linkage: distance



Centroid Linkage: distance<sup>2</sup>

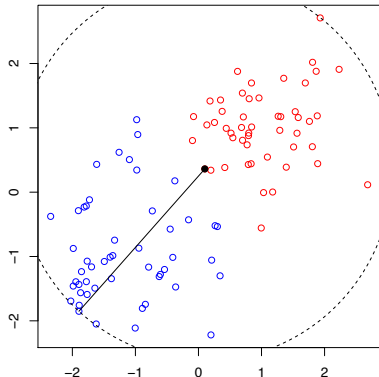


# Minimax Linkage

**Minimax Linkage:** first define radius of a group of points  $G$  around  $x_i$  as  $r(x_i, G) = \max_{j \in G} d_{ij}$ . Then:

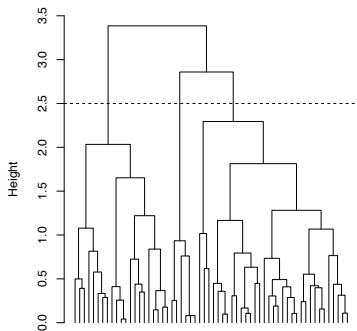
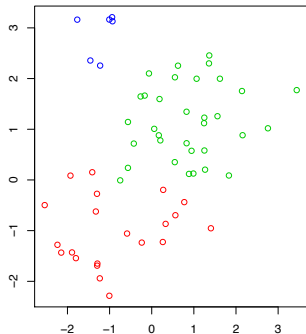
$$d_{\text{minimax}}(G, H) = \min_{i \in G \cup H} r(x_i, G \cup H)$$

minimax linkage score  $d_{\text{minimax}}(G, H)$  is the **smallest radius** encompassing all points in  $G$  and  $H$ . The center  $x_c$  is the black point.



## Example 2: Minimax Linkage

Same data as before; cut tree at  $y = 2.5$  for three clusters.



**Cut Interpretation:** for each  $x_i$ , belongs to a cluster whose center  $X_c$  satisfies  $d_{ic} \leq 2.5$

# Properties of Minimax Linkage

- Cutting a minimax tree at height  $h$  has an interpretation: each point  $\leq h$  in dissimilarity to the center of its cluster.
- Creates dendrograms with no inversions
- Unaffected by monotone transformation of dissimilarity
- Centers of clusters are from the data, this is analogous to K-medoids

# Linkage Summary

Linkage	No Inversions?	Unchanged w/ monotone transforms?	Cut Interpretations?	Notes
Single	✓	✓	✓	chaining
Complete	✓	✓	✓	crowding
Average	✓	✗	✗	
Centroid	✗	✗	✗	simple
Minimax	✓	✓	✓	centers are data points

Overall selection of linkage methods is very much data / situation dependent

# Hierarchical Clustering

- Limitations
  - must choose a linkage method
  - does not scale well
    - time complexity  $O(n^2)$  to  $O(n^3)$
    - space complexity  $O(n^2)$
- Benefits
  - provide dendrogram visualization and knowledge of relationship among data points