Machine Learning, STOR 565

Clustering: Overview and Basic Methods

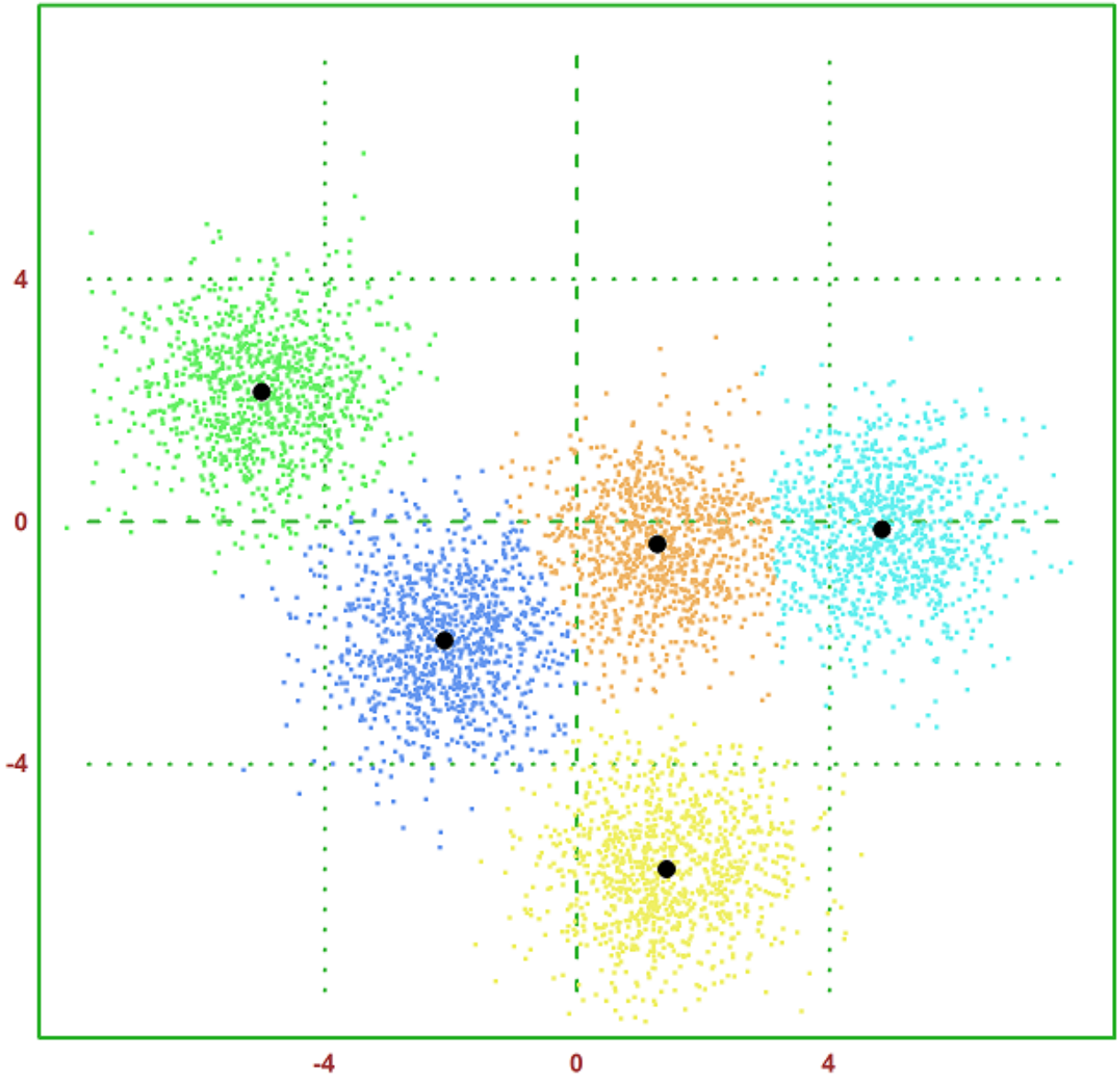
Andrew Nobel

February, 2021

Overview

Task: Divide a set of objects (e.g. data points) into a small number of disjoint groups such that objects in the same group are close together, and objects in different groups are far apart.

Example (http://rosettacode.org)



Example (https://apandre.files.wordpress.com)

How many clusters?

Six Clusters

Two Clusters

Four Clusters

General Setting

Given: Objects $x_1, \ldots, x_n$ in feature space $\mathcal{X}$

- Dissimilarity or distance $d(x_i, x_j)$ between pairs of objects

Goal: Divide $x_1, \ldots, x_n$ into disjoint groups $C_1, \ldots, C_k$, called *clusters*, s.t.

- Objects in same cluster are close together
- Objects in different clusters are far apart
- Number of clusters k is small

Distinction: Clustering is *complete* if it partitions $\mathcal{X}$ and *incomplete* if it partitions only $x_1, \ldots, x_n$.

Clustering: Areas of Application

Genomics, Biology

Data Compression

Psychology

Computer Science

Social and Political Science

Feature Vectors

Objects $\mathbf{x} \in \mathcal{X}$ typically represented by a *feature vector*

$$\mathbf{x} = (x_1, \ldots, x_p)^t$$

where x_i is a numerical/categorical measurement of interest:

- $x_i \in \mathbb{R}$ numerical feature
- $x_i \in \{a, b, \ldots\}$ categorical feature

Examples

Medicine

- Object $=$ patient
- Feature x_i $=$ outcome of a diagnostic test on patient

Microarrays (Genomics)

- Object $=$ tissue sample
- Feature x_i $=$ measured expression level of gene i in that sample

Data Mining

- Object $=$ consumer
- Features x_i $=$ type, location, or amount of recent purchases

Dissimilarities/Distances Between Feature Vectors

Euclidean $d(\mathbf{u}, \mathbf{v}) = \sqrt{\sum_i (u_i - v_i)^2}$

Manhattan $d(\mathbf{u}, \mathbf{v}) = \sum_i |u_i - v_i|$

Correlation $d(\mathbf{u}, \mathbf{v}) = 1 - \text{corr}(u, v)$

Hamming $d(\mathbf{u}, \mathbf{v}) = \sum_i I\{u_i \neq v_i\}$

Mixtures of these

Basic Steps in Clustering

Objects $\mathbf{x}_1, \ldots, \mathbf{x}_n$

$\Downarrow$

Selection and Extraction of Features

$\Downarrow$

Dissimilarity matrix $D = \{d(\mathbf{x}_i, \mathbf{x}_j) : 1 \leq i, j \leq n\}$

$\Downarrow$

Clustering Algorithm

$\Downarrow$

Partition $\pi = \{C_1, \ldots, C_k\}$ of $\mathbf{x}_1, \ldots, \mathbf{x}_n$.

Some Clustering Methods

Hierarchical: Candidate divisions of data described by a binary tree

- Agglomerative (bottom-up)
- Divisive (top-down)

Iterative: Search for local minimum of simple cost function

- k-means and variants
- partitioning around medioids, self organizing maps

Model-based: Fit feature vectors with a finite mixture model

Spectral: Threshold eigenvectors of Laplacian of Dissimilarity Matrix

Features of Clusters

Features of clusters can affect the performance of different procedures, e.g., whether clusters are

- Spherical or elliptical in shape
- Similar in overall variance/spread
- Similar in size (number of points)

The k-Means Algorithm

The k-Means Algorithm

Setting: Objects $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ are vectors. Seek k clusters

Approach: Focus on cluster centers

- Find good cluster centers $\mathbf{c}_1, \ldots, \mathbf{c}_k \in \mathbb{R}^p$
- Let cluster $C_j =$ vectors $\mathbf{x}_i$ closer to $\mathbf{c}_j$ than other centers $\mathbf{c}_l$

Optimization: Select centers to minimize sum of squares (SoS) cost function

$$\text{Cost}(\mathbf{c}_1, \ldots, \mathbf{c}_k) \;=\; \sum_{i=1}^{n} \min_{1 \le j \le k} ||\mathbf{x}_i - \mathbf{c}_j||^2$$

Problem: Exact solution of optimization problem not feasible. Resort to iterative methods that find local minimum

Ingredient 1: Centroids

Definition: The centroid of vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m \in \mathbb{R}^p$ is their (vector) average

$$\mathbf{c} \;=\; \frac{1}{m}\sum_{i=1}^{m} \mathbf{v}_i$$

- Centroid $\mathbf{c}$ is the center of mass of the point configuration $\mathbf{v}_1, \ldots, \mathbf{v}_m$
- Centroid $\mathbf{c}$ is an *optimal representative* for $\mathbf{v}_1, \ldots, \mathbf{v}_m$, in the sense that

$$\sum_{i=1}^{m} ||\mathbf{v}_i - \mathbf{c}||^2 \;\leq\; \sum_{i=1}^{m} ||\mathbf{v}_i - \mathbf{v}||^2$$

  for every vector $\mathbf{v} \in \mathbb{R}^p$

Ingredient 2: Nearest Neighbor Partitions

Idea: Given centers $\mathbf{c}_1, \ldots, \mathbf{c}_k \in \mathbb{R}^p$ one can partition $\mathbb{R}^p$ into corresponding cells $A_1, \ldots, A_k$ where

$$A_j = \{\mathbf{x} : ||\mathbf{x} - \mathbf{c}_j|| \leq ||\mathbf{x} - \mathbf{c}_s|| \text{ all } l \neq j\}$$

contains vectors that are closer to center $\mathbf{c}_j$ than any other center $\mathbf{c}_s$ (where we break ties by index)

Definition: Cells $\{A_1, \ldots, A_k\}$ called the *nearest neighbor* or *Voronoi* partition of $\mathbb{R}^p$ generated by centers $\mathbf{c}_1, \ldots, \mathbf{c}_k$

Note: $A_j = \bigcap_{s \neq j} \{\mathbf{x} : ||\mathbf{x} - \mathbf{c}_j|| \leq ||\mathbf{x} - \mathbf{c}_l||\}$ is an intersection of half-spaces

The k-Means Algorithm

Initialize: Centers $\mathcal{C}_0 = \{\mathbf{a}_1, \ldots, \mathbf{a}_k\}$

Iterate: For $m = 1, 2, \ldots$ do:

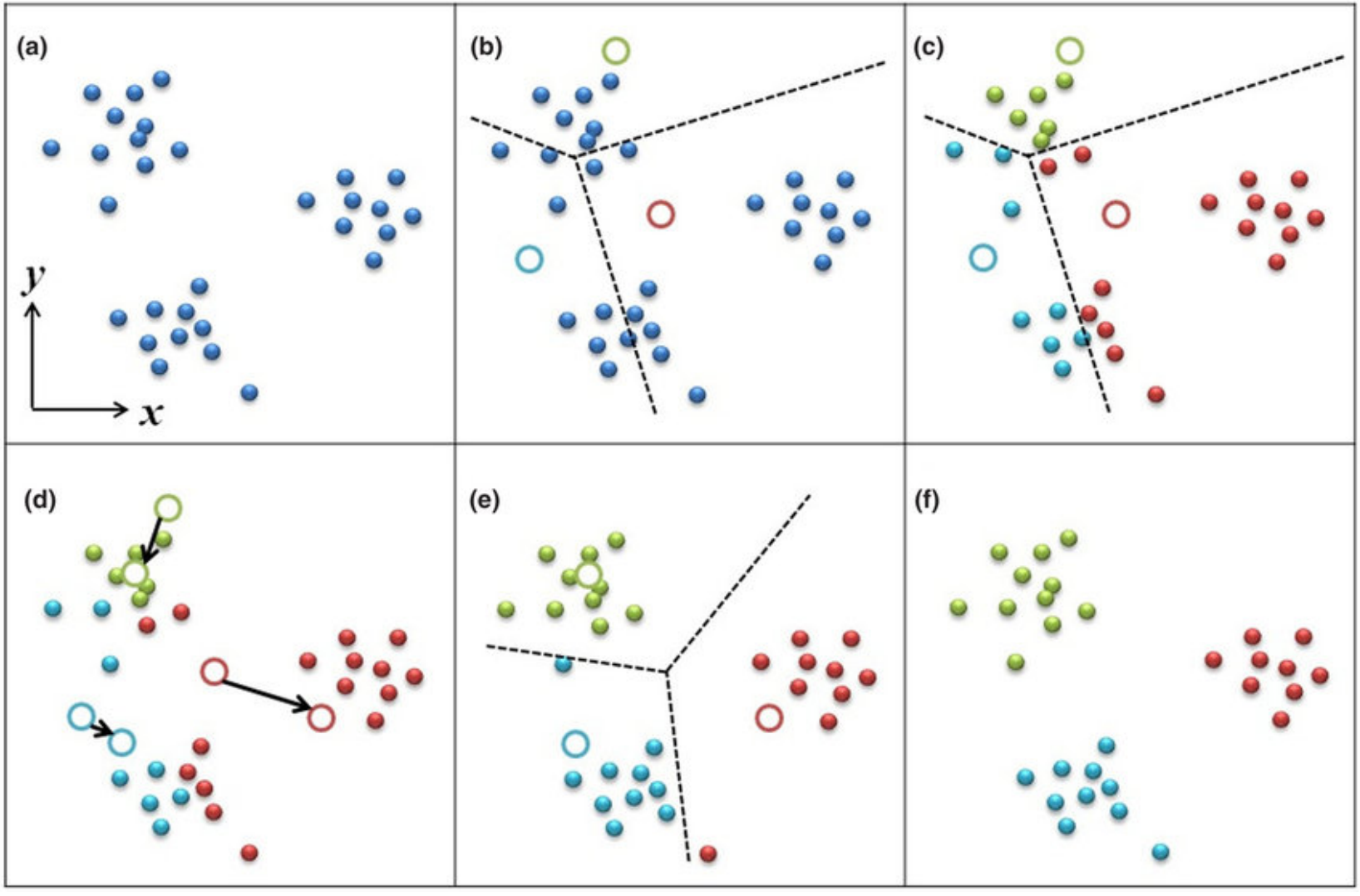
- Let π_m be the nearest neighbor partition of the centers $\mathcal{C}_{m-1}$.
- Let $\mathcal{C}_m$ be the centroids of the vectors in each cell of π_m

Stop: When Cost$(\mathcal{C}_m)$ is close to Cost$(\mathcal{C}_{m+1})$

Key Fact: Cost function decreases at each iteration of algorithm. Recall

$$\text{Cost}(\mathbf{c}_1, \ldots, \mathbf{c}_k) \;=\; \sum_{i=1}^{n} \min_{1 \le j \le k} ||\mathbf{x}_i - \mathbf{c}_j||^2$$

k-means (Yu-Zhong Chen, ResearchGate)



The k-Means Algorithm

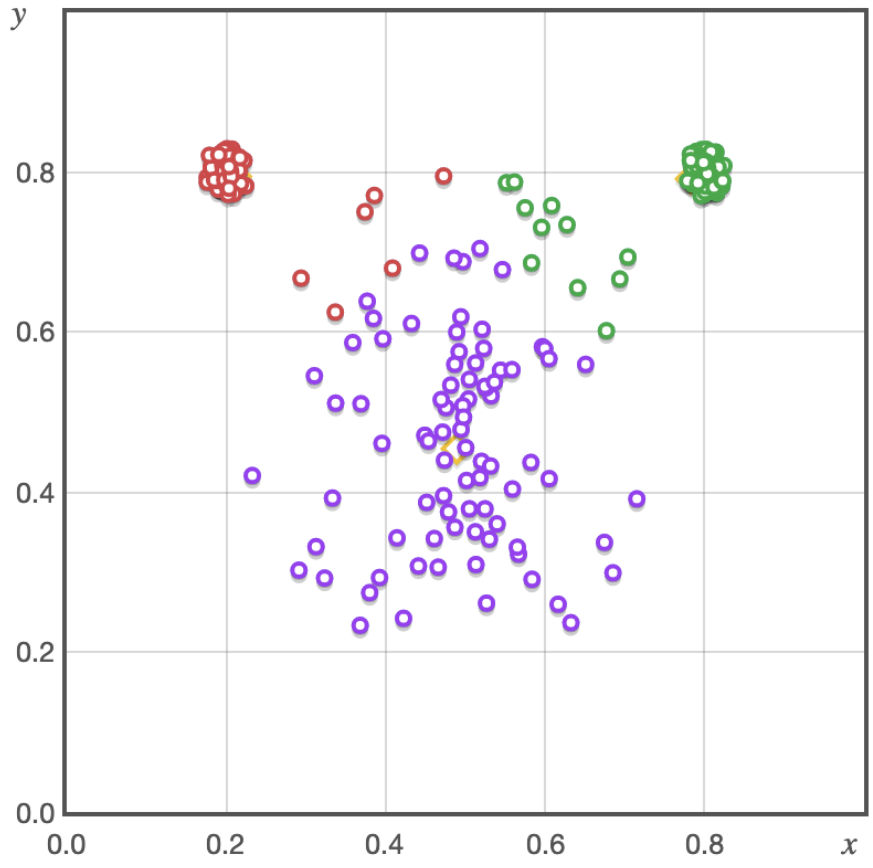
In practice

- Choose multiple initial sets of representative vectors $\mathcal{C}_0 = \{c_1, \ldots, c_k\}$
- Run the iterative k-means procedure
- Choose the partition associated with the smallest final cost

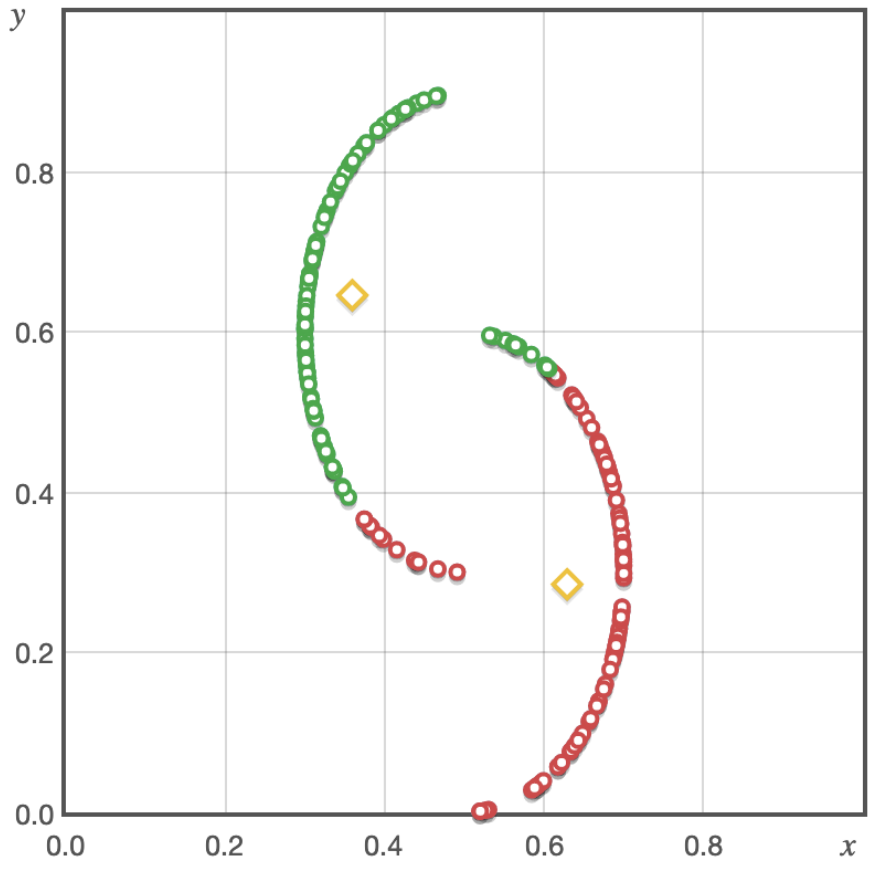
Example: http://onmyphd.com/?p=k-means.clustering.

K-Means tends to perform best when clusters are spherical, similar in variance and size

3-means (onmyphd.com)



2-Means (onmyphd.com)



Agglomerative Clustering

Binary Trees

1. Distinguished node called the **root** with zero or two children but no parent

2. Every other node has one parent and zero or two children

   - Nodes with no children are called **leaves**
   - Nodes with two children are called **internal**

Note: Tree usually drawn upside-down, with root node at the top

Agglomerative Clustering

Stage 0: Assign each object x_i to its own cluster

Stage k:

- Find the two *closest* clusters at stage $k-1$
- Combine them into a single cluster

Stop: When all objects x_i belong to a single cluster

Output: Binary tree T called a *dendrogram*

Note: Closeness of clusters C, C' can be measured in different ways

Distances Between Clusters

Single Linkage

$$d_s(C, C') = \min_{x_i \in C,\, x_j \in C'} d(x_i, x_j)$$

Average Linkage

$$d_a(C, C') = \frac{1}{|C|\,|C'|} \sum_{x_i \in C,\, x_j \in C'} d(x_i, x_j)$$

Total Linkage

$$d_t(C, C') = \max_{x_i \in C,\, x_j \in C'} d(x_i, x_j)$$

Dendrogram

Binary tree associated with the agglomerative clustering procedure: it is a graphical record of the clustering process

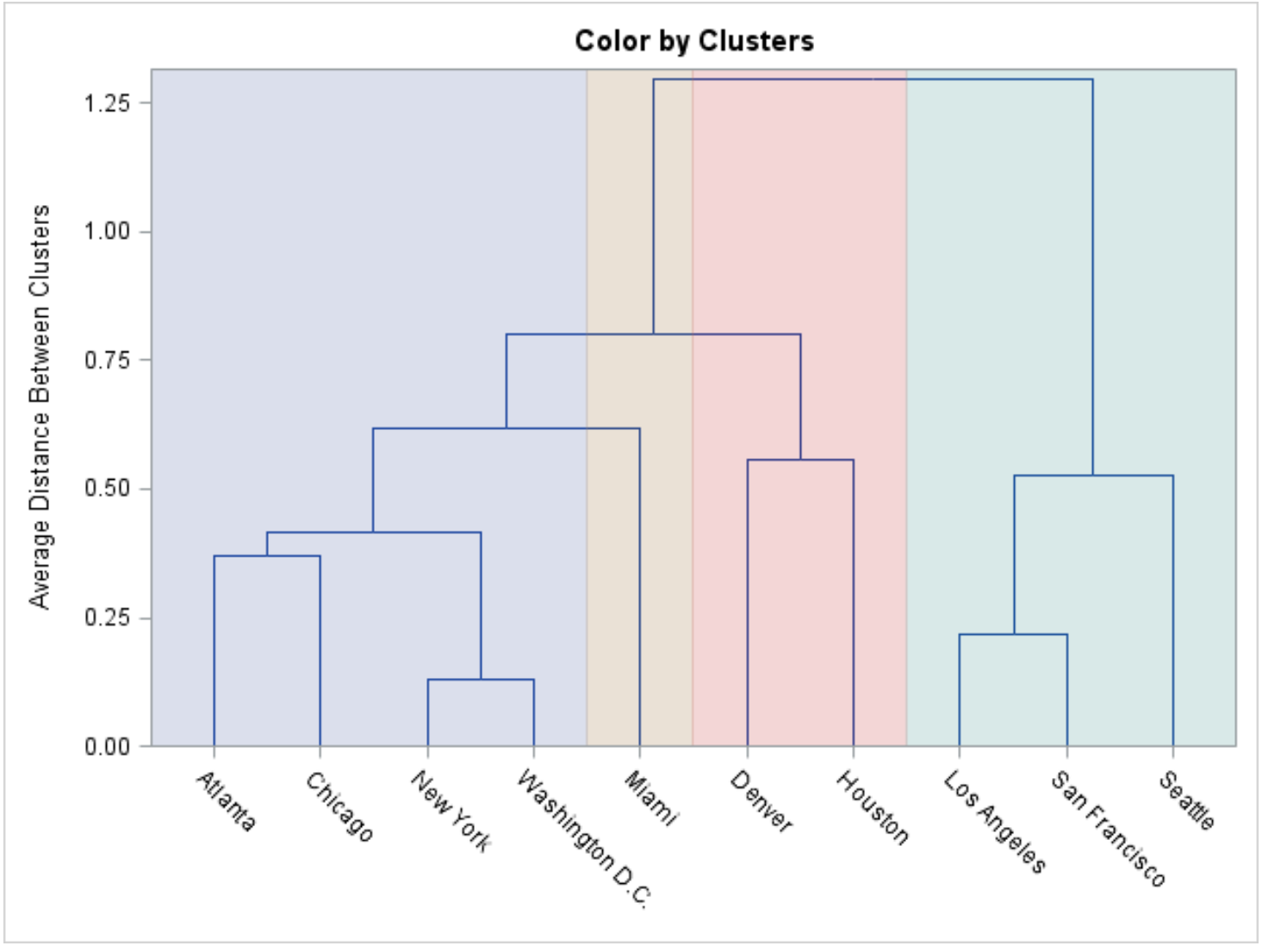
Initialize: Each singleton cluster $\{x_i\}$ corresponds to a node at height 0

Update: If two clusters C, C' are combined, their respective nodes are joined to a parent node at height $d(C, C')$

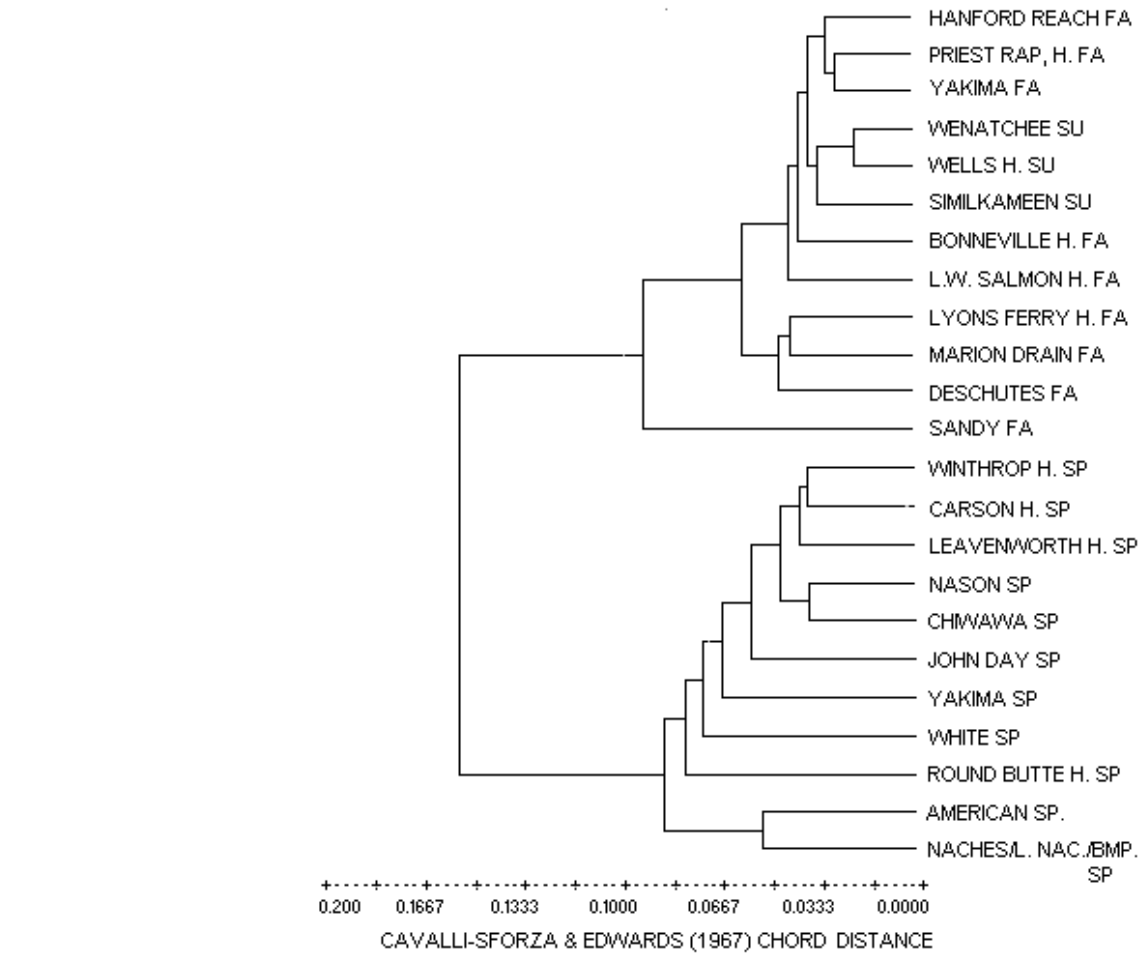
Each node of dendrogram corresponds to a set of objects. Objects associated with two nodes are merged when forming their parent

- Leaves correspond to individual objects
- The root corresponds to all objects

Cities by Distance (blogs.sas.com)



Salmon by Genetic Similarity



Dendrogram, cont.

Note: Dendrogram T represents many possible clusterings, one for each (rooted) subtree.

Methods for selecting a clustering/subtree

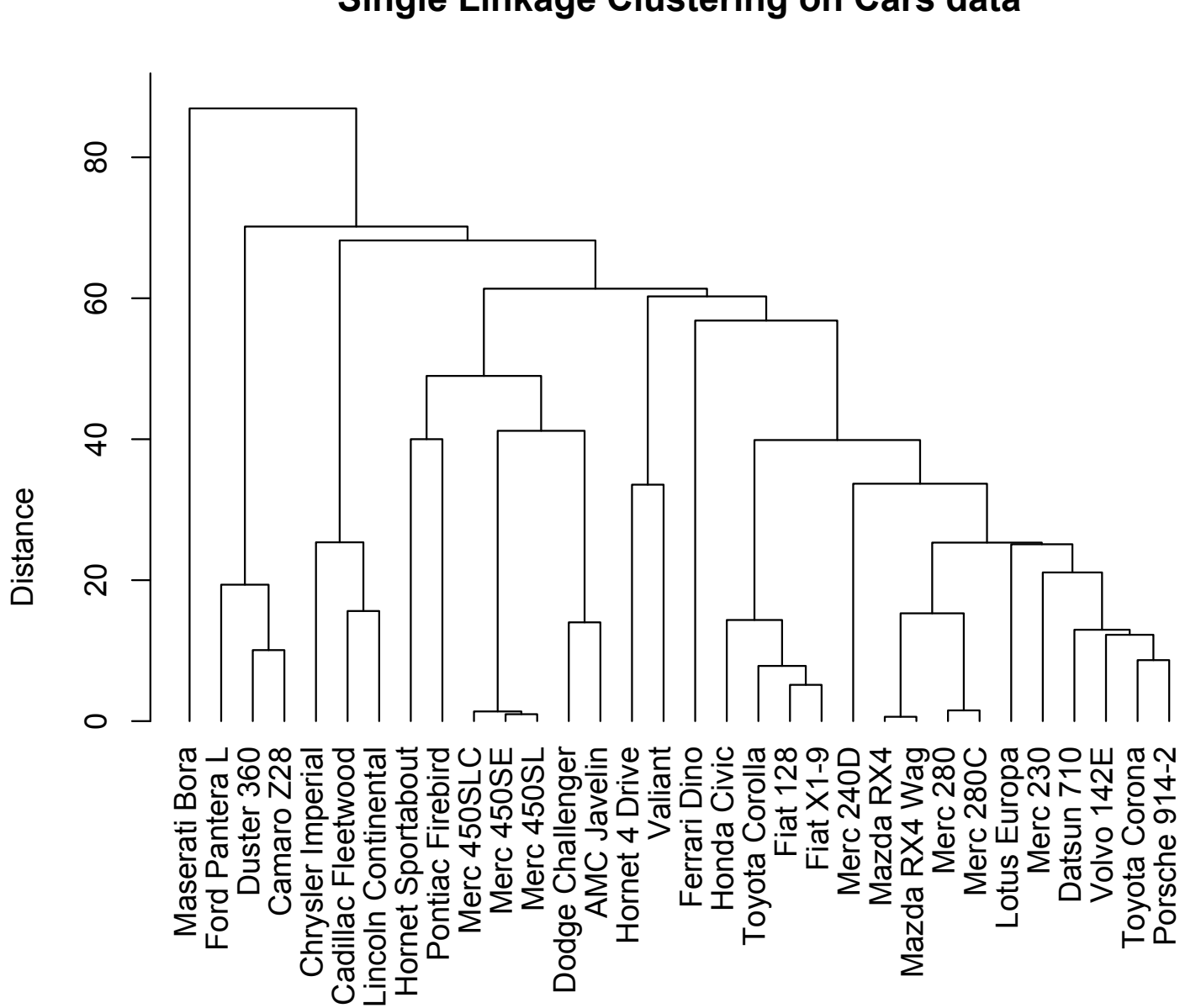
- Ad hoc selection (by eye)
- "Cutting" dendrogram at fixed level
- Penalized pruning

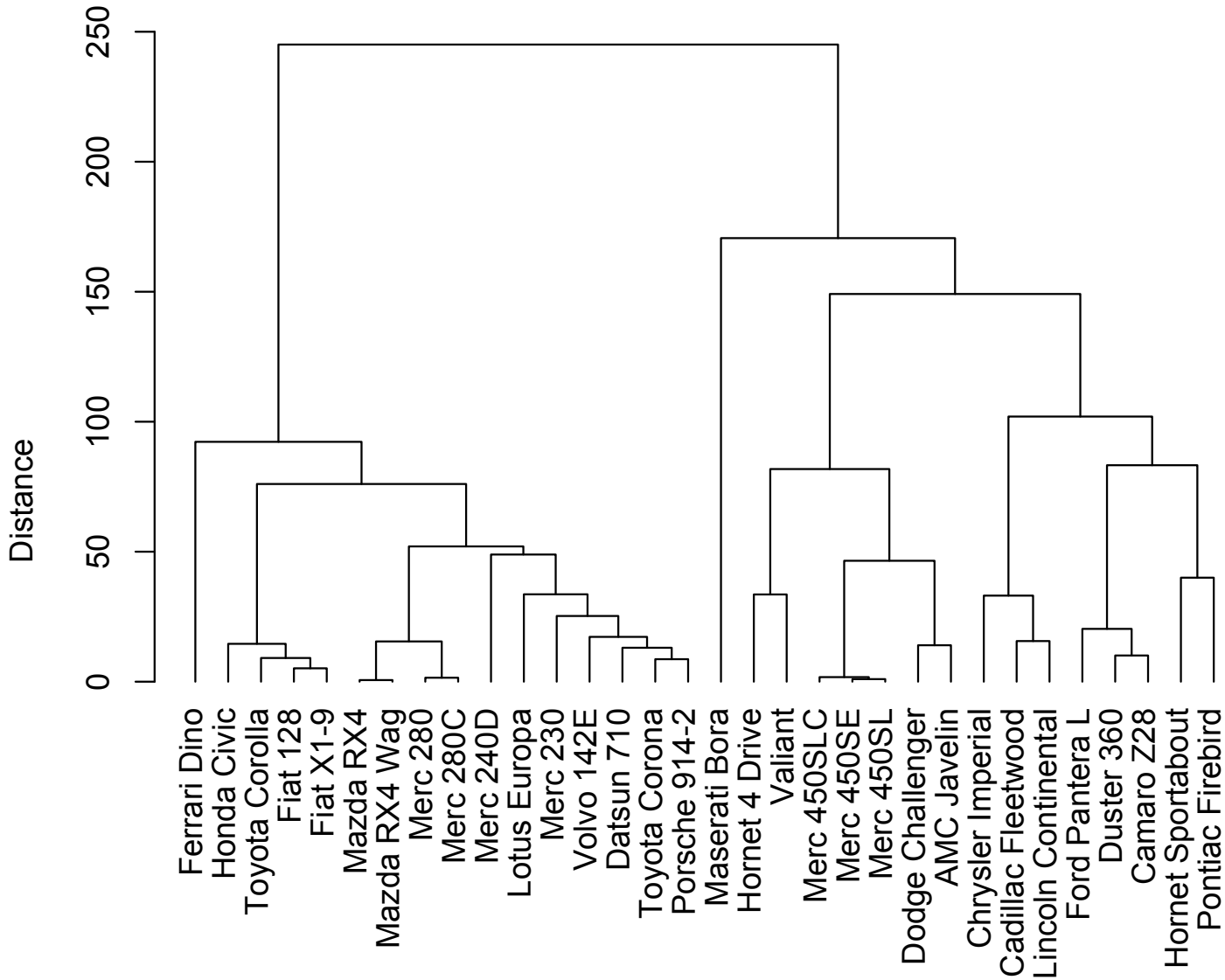
Visualization of clustering structure

- Order objects in the same way as the leaves of the dendrogram
- Caveat: many orderings possible

Cars Data

- **Samples**: 32 unique cars
- **Variables**: 11 descriptive variables, including gas mileage, horsepower, number of cylinders, etc.
- Freely available in **R**: *data(mtcars)*


Single Linkage Clustering on Cars data
Distance
0
20
40
60
80
Maserati Bora
Ford Pantera L
Duster 360
Camaro Z28
Chrysler Imperial
Cadillac Fleetwood
Lincoln Continental
Hornet Sportabout
Pontiac Firebird
Merc 450SLC
Merc 450SE
Merc 450SL
Dodge Challenger
AMC Javelin
Hornet 4 Drive
Valiant
Ferrari Dino
Honda Civic
Toyota Corolla
Fiat 128
Fiat X1-9
Merc 240D
Mazda RX4
Mazda RX4 Wag
Merc 280
Merc 280C
Lotus Europa
Merc 230
Datsun 710
Volvo 142E
Toyota Corona
Porsche 914-2

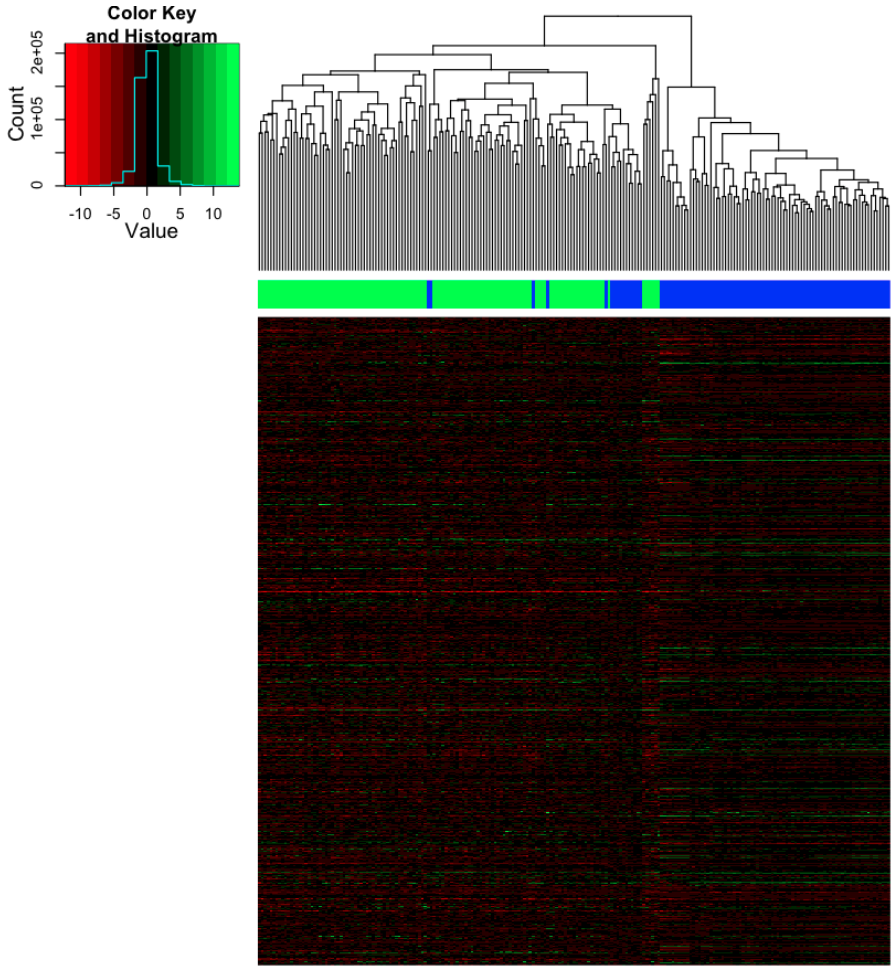

Average Linkage Clustering on Cars data
Distance
0
50
100
150
200
250
Ferrari Dino
Honda Civic
Toyota Corolla
Fiat 128
Fiat X1-9
Mazda RX4
Mazda RX4 Wag
Merc 280
Merc 280C
Merc 240D
Lotus Europa
Merc 230
Volvo 142E
Datsun 710
Toyota Corona
Porsche 914-2
Maserati Bora
Hornet 4 Drive
Valiant
Merc 450SLC
Merc 450SE
Merc 450SL
Dodge Challenger
AMC Javelin
Chrysler Imperial
Cadillac Fleetwood
Lincoln Continental
Ford Pantera L
Duster 360
Camaro Z28
Hornet Sportabout
Pontiac Firebird

TCGA Data

Gene expression data from The Cancer Genome Atlas (TCGA)

- **Samples**
  - 95 Luminal A breast tumors
  - 122 Basal breast tumors
- **Variables**: 2000 randomly selected genes

TCGA Data



- Clustered samples (breast tumor subtype)
- Colors: Luminal A and Basal

Important Questions

- What is the right number of clusters?
- What is right measure of distance?
- What is the best clustering method for the data?
- How robust is an observed clustering to small perturbations of the data?
- What significance can be assigned to the clusters?