3 Clustering

Clustering refers to a broad set of techniques for finding *subgroups* in a data set.

We seek to partition observations into distinct groups so that
- observations within a group are similar
- observations in different groups are dissimilar

need to define
Depend on domain!

For instance, suppose we have a set of n observations, each with p features. The n observations could correspond to tissue samples for patients with breast cancer and the p features could correspond to measurements collected for each tissue sample:

- clinical measurements, e.g. tumor stage or grade
- gene expression measurements.

diverse in character

We may have reason to believe there is heterogeneity among the n observations.

e.g. different unknown subtype of cancer.

This is *unsupervised* because

We are trying to discover structure (distinct clusters) in the absence of a response.

vs.

Supervised problems we have the goal of prediction of a response.

Both clustering and PCA seek to simplify the data via a small number of summaries.

- PCA – finds a low dimensional representation of the observations that explain a good fraction of the variance.
- Clustering – finds homogenous subgroups among observations

Since clustering is popular in many fields, there are many ways to cluster.

We will focus on 2 best-known clustering approaches.

- K-means clustering

  Seeks to partition the observations into a pre-specified # of clusters.

- Hierarchical clustering

  We do not know in advance how many clusters we want.

  We obtain clusterings for $1, \ldots, n$ # of clusters

  ↳ can view in a tree-like visualization called a "dendrogram"

In general, we can ① cluster observations on the basis of features or ② we can cluster features on the basis of observations.

① ↓ identify subgroups among observations

② ↓ identify subgroups among the features.

We will focus on ①

But we can perform ② by transposing the data matrix.

$X \longrightarrow X^T \longrightarrow$ clustering.

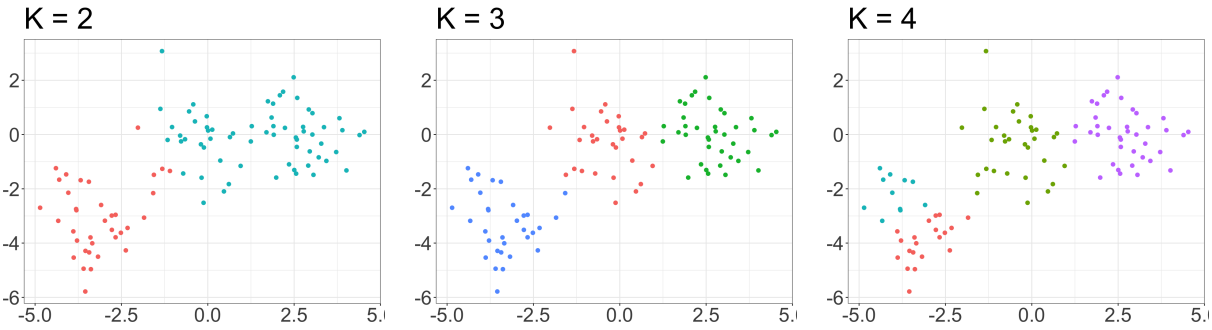
$n \times p$ $\quad p \times n$

3.1 K-Means Clustering

Simple and elegant approach to parition a data set into K distinct, non-overlapping clusters.

We must first specify how many clusters K.
Then K-means assigns each observation to one of the K clusters.

eg. clustering $n=100$ observations into K clusters using $p=2$ features.



The K-means clustering procedure results from a simple and intuitive mathematical problem. Let $C_1, \ldots, C_K$ denote sets containing the indices of observations in each cluster. These satisfy two properties:

e.g. if observation i is in cluster k, $i \in C_k$

1. $C_1 \cup \cdots \cup C_K = \{1, 2, \ldots, n\}$

   each observation belongs to one of the K clusters.

2. $C_k \cap C_{k'} = \emptyset \quad \forall k \neq k'$

   The clusters are non overlapping.

Idea: "good clustering" is one for which the within-cluster variation is as small as possible.

The *within-cluster variation* for cluster C_k is a measure of the amount by which the observations within a cluster differ from each other.

Call this $W(C_k)$.

Then we want to solve the problem:

$$\underset{C_1,\dots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} W(C_k) \right\}$$ ← we want to partition observations into K clusters s.t. total within-cluster variation is minimized.

To solve this, we need to define within-cluster variation.

Many way we can do that.

Most common way: squared euclidean distance:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

↑ # obs in k^{th} cluster.

This results in the following optimization problem that defines K-means clustering:

$$\underset{C_1,\dots,C_K}{\text{minimize}} \left\{ \sum_{k=1}^{K} \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}$$

objective function

This is very difficult to solve exactly! $\approx K^n$ ways to partition n obs. into K clusters.

A very simple algorithm has been shown to find a local optimum ("pretty good solution") to this problem:

1. randomly assign a number from 1 to K to each observation
   these are initial cluster assignments for the observations

2. Iterate until cluster assignments stop changing.
   (a) For each of the K clusters compute cluster centroid ← vector of the p feature means for observations in each cluster (K).
   (b) assign each observation to the closest centroid cluster.
   ↑ euclidean distance.

Algorithm is guaranteed to decrease value of objective function at each step.
When cluster assignments stop changing this is a local minimum.
↳ not necessarily global min ⇒ clustering depends on initial (random) cluster values (step 1).

What to do? Run the algorithm multiple times from different initial configurations and choose clustering w/ smallest object function.

Problem is we still must choose K! More later...

3.2 Hierarchical Clustering

One potential disadvantage of K-means clustering is that it requires us to specify the number of clusters K. *Hierarchical clustering* is an alternative that does not require we commit to a particular K.

ahead of time

hierarchical clustering also results in a tree-based representation of the observations.

clusters getting larger.

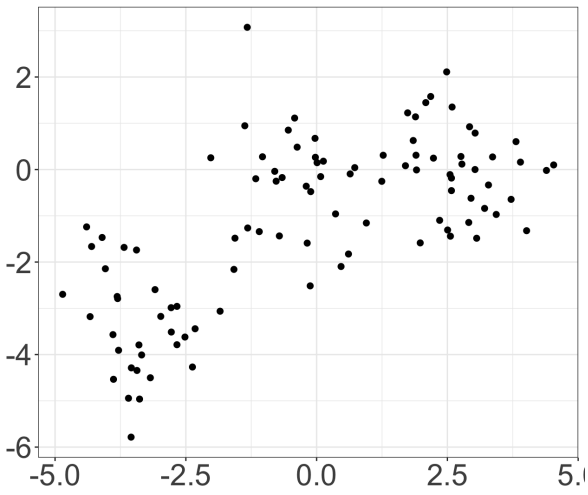
We will discuss *bottom-up* or "*agglomerative*" clustering.

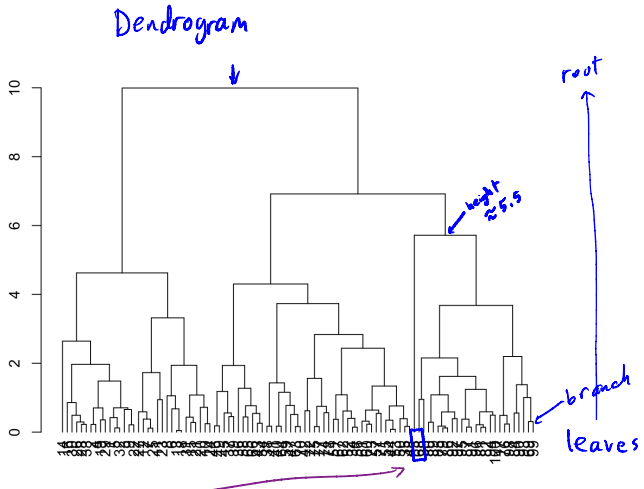
start with every observation in its own cluster and merge (fuse) clusters until all observations are in a single cluster (n clusters of size 1 ⟶ 1 cluster of size n).

"bottom-up" refers to representation of clusters in tree diagram w/ leaves on the bottom.

3.2.1 Dendrograms

Same simulated data as before n=100 observations w/ p=2 features.





branches = clusters w/ more than 1 observation

leaves = cluster w/ 1 observation

even though these observations are right next to each other on x-axis of dendrogram, they are quite different b/c height of first fusion.

Each *leaf* of the dendrogram represents one of the 100 simulated data points.

As we move up the tree, leaves begin to fuse into branches, which correspond to observations that are similar to each other.

- as we move higher up the tree, branches fuse with other branches or w/ leaves.
- the lower a fusion occurs, the more similar the observations are.
- observations that fuse high up in the tree can be quite different

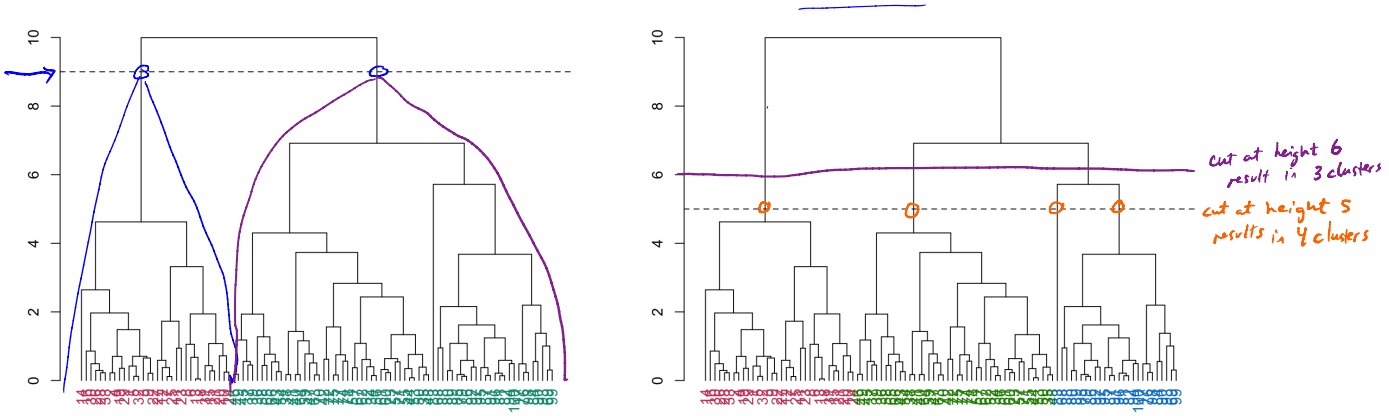
More precisely: For any two observations, we can look for the point in the tree where branches containing those two observations are first fused.

the height of this fusion indicates how different they are!

We draw conclusions about similarity of two observations based on the location on the vertical axis where branches containing those observations are first fused.

How do we get clusters from the dendrogram?

We make horizontal "cuts" across the dendrogram.



We can cut at a height that corresponds to $1, \ldots, n$ clusters. (i.e. height of cut is similar to K in K-means).

$\Rightarrow$ A single dendrogram can be used to obtain any number of clusters!

In practice: people inspect dendrogram and choose where to cut based on heights of fusion and # cluster resulting (subjective).

The term *hierarchical* refers to the fact that clusters obtained by cutting the dendrogram at a given height are necessarily nested within the clusters obtained by cutting the dendrogram at a greater height.

This hierarchical assumption may or may not be realistic.

e.g. suppose have group of observations 50-50 split M/F and evenly split American, Japanese, French.

Maybe 2 clusters results in clustering by sex
3 clusters results in a clustering by nationality } not nested.

3.2.2 Algorithm

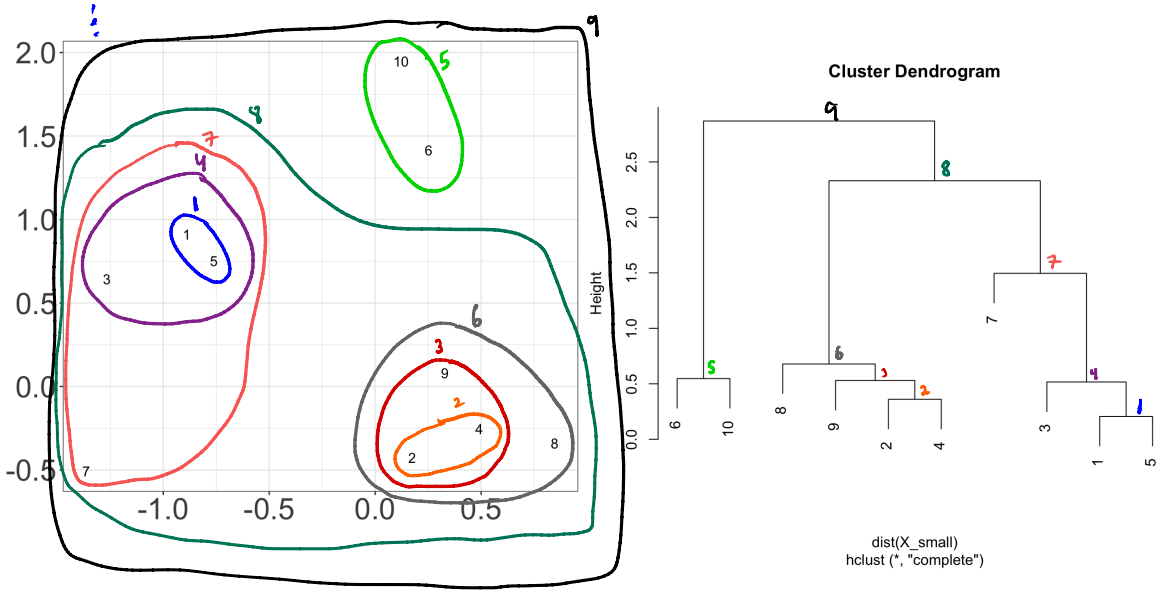
First, we need to define some sort of *dissimilarity* metric between pairs of observations.

Most often Euclidean distance is used.

Then the algorithm proceeds iteratively.

start at the bottom of dendrogram each obs. is in own cluster

find 2 most similar clusters and fuse.



More formally,

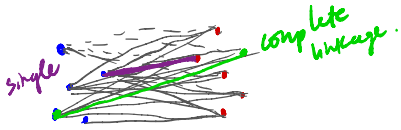
1. Begin with n observations and a measure of all $\binom{n}{2}$ pairwise dissimilarities (e.g. euclidean distance). Treat each observation as its own cluster.

2. For $i = n, n-2, \ldots, 2$
   (a) Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair that is least dissimilar. Fuse (merge) these two clusters. The dissimilarity between these two clusters is the height at which the fusion should be placed.
   (b) Compute new pairwise inter-cluster dissimilarities among the $i-1$ remaining clusters.

detail
One issue has not yet been addressed.
inter-cluster dissimilarity?
How to fuse in step 4? cluster {1,5} and {3}?
We have dissimilarity between pairwise observations not clusters!

How do we determine the dissimilarity between two clusters if one or bother of them contains multiple observations?

We develop the notion of "linkage" – defines dissimilarity between groups of observations.



Most common types:

most used stat/ml

* 1. Complete: maximal inter cluster dissimilarity
   Compute all pairwise dissimilarity between points in two clusters choose max
2. Single: minimal intercluster dissimilarity.
   all pairwise dissimilarities between 2 clusters choose min.
* 3. Average: mean intercluster dissimilarity
   average all pairwise dissimilarities btw/ 2 clusters
4. Centroid: dissimilarity btw centroid of two clusters.
   ↳ can lead to inversions

used in genomics.

3.2.3 Choice of Dissimilarity Metric

- So far we have used Euclidean distance.
- Could alternatively use correlation-based $(1 - |corr|)$

Choice of dissimilarity because it has a strong result on the dendrogram.
↳ choose via type of data and scientific question!

3.3 Practical Considerations in Clustering

In order to perform clustering, some decisions should be made.

- Should observations be scaled? centered?
  if variables are measured on different scales, probably.
- Hierarchical clustering:
  - What dissimilarity metric?
  - What type of linkage?
  - Where to cut dendrogram?

  Some linkages will produce dendrograms with undesirable characteristics
  ↳ try a different linkage.
- K-means:
  how many clusters should we have?

Each of these decisions can have a strong impact on the results obtained. What to do?

There is no 1 right answer. Any clustering that results in some "interesting" structure is valid.

There are some ways "validate" clusterings vs. what would expect to see by chance.

involve comparing within cluster variation to between cluster variation.

e.g. "Dunn index"