Lecture 2: Classification & Clustering

STATS 202: Data Mining and Analysis

Linh Tran

tranlm@stanford.edu

Department of Statistics
Stanford University

June 28, 2023

Announcements

- Any issues with Piazza/Gradescope?

Outline

- Classification
  - K-nearest neighbors
  - Naive Bayes
- Clustering
  - K-means
  - Hierarchical clustering

Bayes classifier

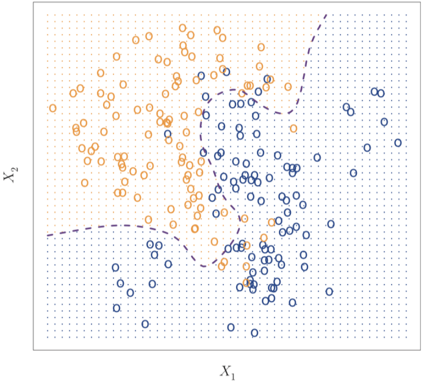
- f_0 gives us a probability of the observation belonging to each class.
- To select a class, we can just pick the element in $f_0 = [p_1, p_2, ..., p_K]$ that's the largest
  - Called the Bayes Classifier
- As a classifier, produces the lowest error rate

Bayes error rate

$$1 - \mathbb{E}_0 \left[\max_y \mathbb{P}_0[Y = y | X_1, X_2, ..., X_p] \right] \tag{1}$$

Analogous to the irreducible error described previously

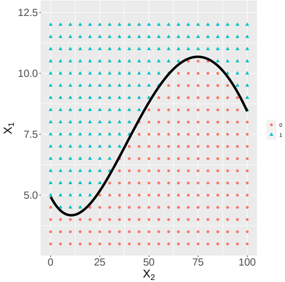
Example: Classifying in 2 classes with 2 features.



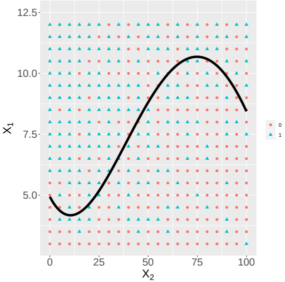
The Bayes error rate is 0.1304.

Note: $\mathcal{C}(\mathbf{x}) = \arg\max_{y} f_0(y)$ may seem easier to estimate

- Can still be hard, depending on the distribution f_0, e.g.



Bayes error $= 0.0$

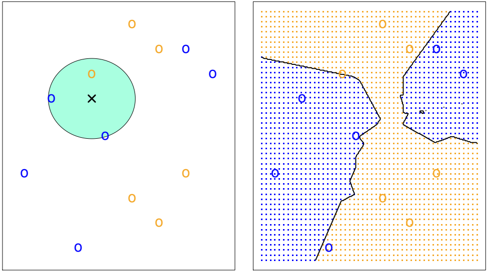


Bayes error $= 0.3$

K-nearest neighbors

How do we estimate Bayes classifier $\mathcal{C}(\mathbf{x})$?

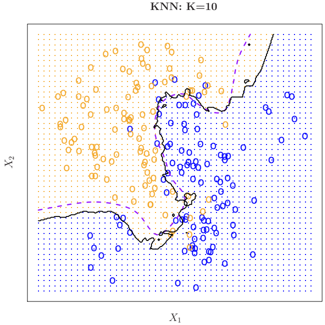
- Could just vote based on the K nearest neighbors (where K is some positive integer)



The KNN approach, using $K = 3$.

Using KNN (i.e. $\hat{f}_n^{knn}$) as a classifier $\mathcal{C}(\mathbf{x})$, we can estimate Bayes boundary f_0^*.

- Despite simplicity, $\hat{f}_n^{knn}$ can be surprisingly close



The KNN ($K = 10$) and Bayes decision boundaries.

K-nearest neighbors

Mathematically, we can represent KNN as

K-nearest neighbors

$$\mathbb{P}(Y = j | X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} \mathbb{I}(y_i = j) \tag{2}$$

We can apply Bayes rule to the resulting probabilities to get our classifier.

K-nearest neighbors

Some details to consider in our KNN implementation:

- Are all our X_i's on the same scale?
  - Typically, will standardize all features to be mean 0 and variance 1.

Some details to consider in our KNN implementation:

- Are all our X_i's on the same scale?
  - Typically, will standardize all features to be mean 0 and variance 1.
- How do we measure distance?
  - Typically, the Euclidean distance is used, e.g.

$$d_{(i)} = ||x_{(i)} - x_0|| \tag{3}$$

K-nearest neighbors

Some details to consider in our KNN implementation:

- Are all our X_i's on the same scale?
  - Typically, will standardize all features to be mean 0 and variance 1.
- How do we measure distance?
  - Typically, the Euclidean distance is used, e.g.

$$d_{(i)} = ||x_{(i)} - x_0|| \tag{3}$$

- Ties are typically broken randomly

K-nearest neighbors

Some details to consider in our KNN implementation:

- Are all our X_i's on the same scale?
  - Typically, will standardize all features to be mean 0 and variance 1.
- How do we measure distance?
  - Typically, the Euclidean distance is used, e.g.

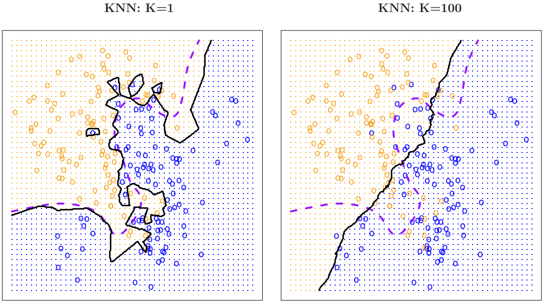
$$d_{(i)} = ||x_{(i)} - x_0|| \tag{3}$$

- Ties are typically broken randomly
- What size K do we use?
  - Estimated with e.g. test set

K-nearest neighbors

Higher values of K will result in smoother decision boundaries

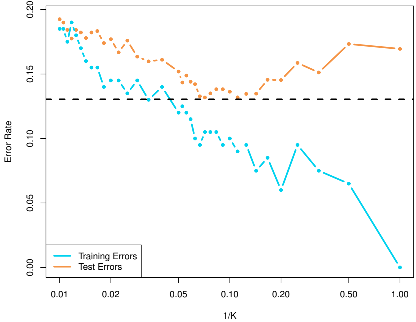
- You're trading off higher variance for higher bias



Two KNN boundary estimates ($K = 1$ and $K = 100$).

More flexibility (i.e. lower K) will result in over-fitting

- Similar to regression setting



KNN training/test errors as a function of K. Black line is Bayes error.

Naive bayes

Another simple estimator is *Naive Bayes*.

By Bayes Theorem, we have

$$\begin{aligned}
\mathbb{P}_0(Y|X_1, X_2) &= \frac{\mathbb{P}_0(Y)\mathbb{P}_0(X_1, X_2|Y)}{\mathbb{P}_0(X_1, X_2)} && (4) \\
&= \frac{\mathbb{P}_0(X_1, X_2, Y)}{\mathbb{P}_0(X_1, X_2)} && (5)
\end{aligned}$$

We only care about the numerator

- It's a function of Y

Naive bayes

Typically, we have

$$\mathbb{P}_0(X_1, X_2, Y) = \mathbb{P}_0(Y) \cdot \mathbb{P}_0(X_1|Y) \cdot \mathbb{P}_0(X_2|X_1, Y) \tag{6}$$

However, we "naively" assume independence such that

$$\mathbb{P}_0(X_2|X_1, Y) \approx \mathbb{P}_0(X_2|Y) \tag{7}$$

Consequently, we have

$$\begin{aligned} \mathbb{P}_0(Y|X_1, X_2) &\propto \mathbb{P}_0(Y)\mathbb{P}_0(X_1, X_2|Y) \qquad (8) \\ &\approx \mathbb{P}_0(Y)\prod_{i=1}^{2}\mathbb{P}_0(X_i|Y) \qquad (9) \end{aligned}$$

Naive bayes

$$\mathbb{P}_0(Y|X_1, X_2) \;\approx\; \frac{1}{Z}\mathbb{P}_0(Y)\prod_{i=1}^{2}\mathbb{P}_0(X_i|Y)$$

We can estimate $\mathbb{P}_0$ empirically

- e.g. *kernel density estimation*
- Could also use parametric models (e.g. Gaussian distribution)
- Question: What if the feature is categorical?

Naive bayes

$$\mathbb{P}_0(Y|X_1, X_2) \approx \frac{1}{Z}\mathbb{P}_0(Y)\prod_{i=1}^{2}\mathbb{P}_0(X_i|Y)$$

We can estimate $\mathbb{P}_0$ empirically

- e.g. *kernel density estimation*
- Could also use parametric models (e.g. Gaussian distribution)
- Question: What if the feature is categorical?

Remark: we don't need Z if we're just classifying

- Just take the class with the max value, e.g.

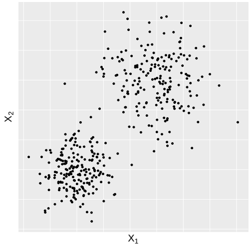
Example naive bayes classifier

$$\hat{y}_n = \mathcal{C}(X_1, X_2) = \underset{y \in \{\text{Orange}, \text{Blue}\}}{\arg\max} \mathbb{P}_0(y)\prod_{i=1}^{2}\mathbb{P}_0(X_i|y) \tag{10}$$

Clustering

Sometimes, we do not have the classes as our output Y.
But we still want to assign each observation to a group.

- This is referred to as *Clustering*
- Falls into unsupervised learning (i.e. no clearly defined outcome of interest)
- Our goal is to find homogeneous subgroups among the observations



Clustering

There are many types of clustering algorithms.

We will cover three:

- K-means clustering
- Hierarchical clustering
- Expectation maximization algorithm
  - Beyond scope of our class

K-means clustering

Clusters all observations into K clusters

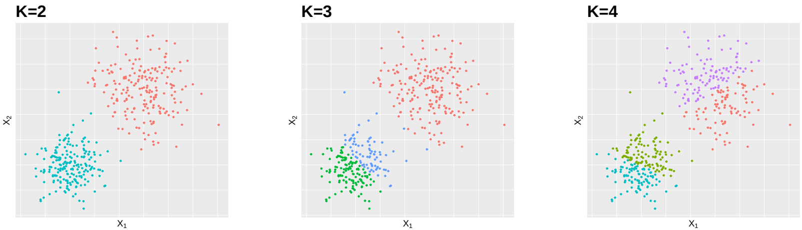
- K must be specified a-priori
- Algorithm then assigns every point to one of the K clusters
- Object is to minimize the *within-cluster variation*, i.e.

K-means clustering

$$\min_{C_1, C_2, \ldots, C_k} \sum_{\ell=1}^{K} W(C_\ell) \quad : \quad W(C_\ell) = \frac{1}{|C_\ell|} \sum_{i,j \in C_\ell} \mathcal{D}^2(\mathbf{x}_i, \mathbf{x}_j)$$

$\mathcal{D}(\mathbf{x}, \mathbf{y})$ measures the distance between $\mathbf{x}$ and $\mathbf{y}$ (typically the Euclidean distance, i.e. $\sqrt{\sum_{j=1}^{p} (x_j - y_j)^2}$).

K-means clustering



Results from applying K-means clustering with different K's.

K-means clustering

Algorithm steps

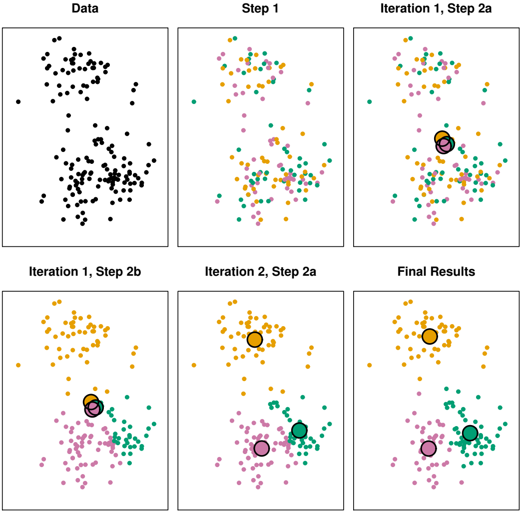
K-means clustering

1. Assign each observation (randomly) to one of the K clusters.
2. Iterate the 2 following steps until cluster assignments stop changing:
   a. Find the centroid of each of the K clusters

$$\bar{\mathbf{x}}_\ell = \frac{1}{|C_\ell|} \sum_{i \in C_\ell} \mathbf{x}_i \tag{11}$$

   b. Reassign each sample to the nearest centroid (using $\mathcal{D}^2(\mathbf{x}, \mathbf{y})$)

K-means clustering



Visualization of k-means at different steps.

K-means clustering

Some properties of K-means

- The algorithm always converges to a local minimum of

$$\min_{C_1, C_2, \ldots, C_k} \left\{ \sum_{\ell=1}^{K} \frac{1}{|C_\ell|} \sum_{i,j \in C_\ell} \mathcal{D}^2(\mathbf{x}_i, \mathbf{x}_j) \right\} \tag{12}$$

- The algorithm is random
  - Each initialization can result in a different minimum
  - Can run with with multiple initializations and select lowest minimum

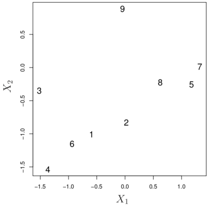
K-means clustering

Example of running K-means 6 different times ($K = 3$).

Hierarchical clustering

Most algorithms for hierarchical clustering are *agglomerative*.

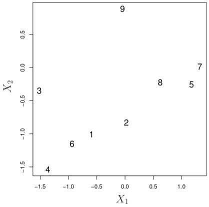
e.g.

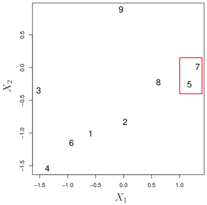


Hierarchical clustering

Most algorithms for hierarchical clustering are *agglomerative*.

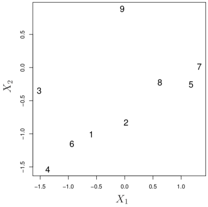
e.g.

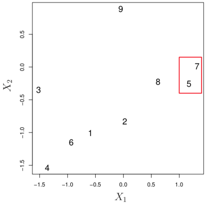


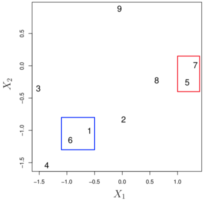


Most algorithms for hierarchical clustering are *agglomerative*.

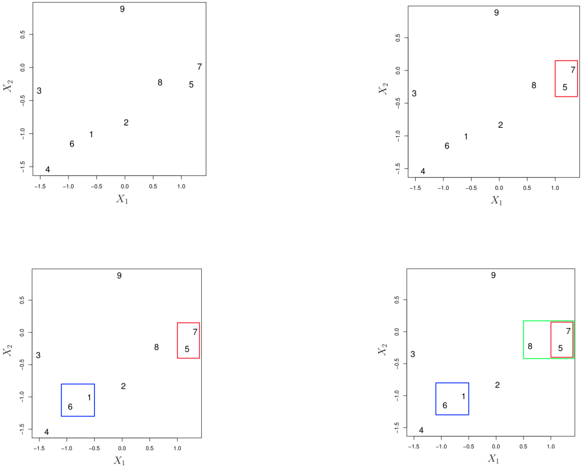
e.g.



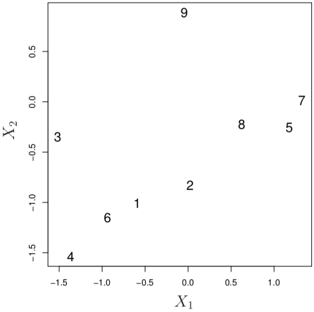


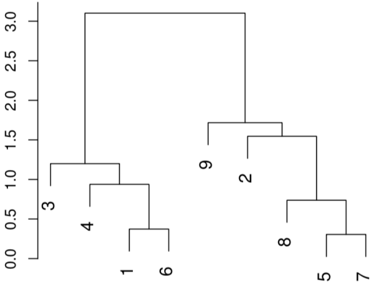


Most algorithms for hierarchical clustering are *agglomerative*.
e.g.



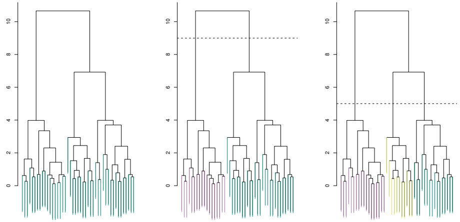
- The algorithm results in a *dendogram*
- *Hierarchical* in the sense that lower clusters are nested within higher clusters





Hierarchical clustering

- The number of clusters does not need to be specified a-priori
- Clusters created by cutting dendogram at a vertical point
- **Note**: Not all segmentation problems are nested clusters.
  - e.g. Market segmentation for consumers of 2 genders from 3 different nationalities.
  - Wierd to divide into 2 groups, and then to further divide 1 in half



Hierarchical clustering

In each iteration, we fuse the 2 clusters *closest* to each other.

- While we can use the Euclidean distance, what if a cluster has multiple observations?

Hierarchical clustering

In each iteration, we fuse the 2 clusters *closest* to each other.

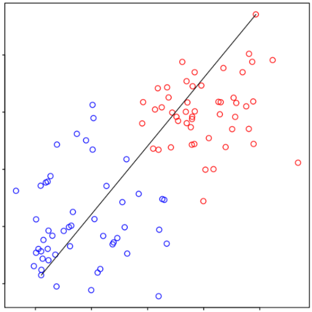
- While we can use the Euclidean distance, what if a cluster has multiple observations?

Linkage defines the dissimilarity between two clusters

Four primary types:

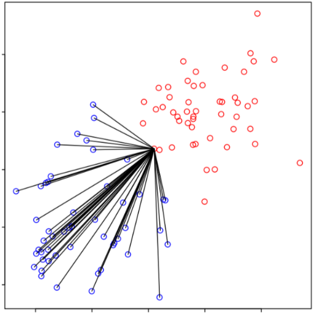
1. Complete
2. Average
3. Single
4. Centroid

Hierarchical clustering



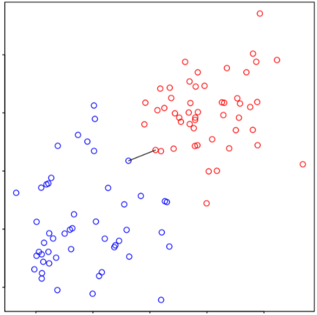
Complete linkage:

- The distance between 2 clusters is the maximum distance between any pair of samples, one in each cluster.



Average linkage:

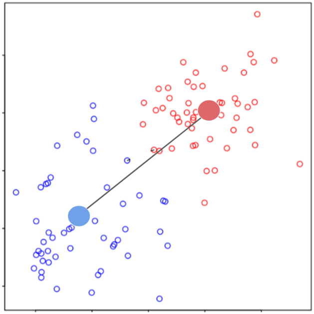
- The distance between 2 clusters is the average of all pairwise distances.



Single linkage:

- The distance between 2 clusters is the minimum distance between any pair of samples, one in each cluster.
- *Suffers from chaining phenomenon*

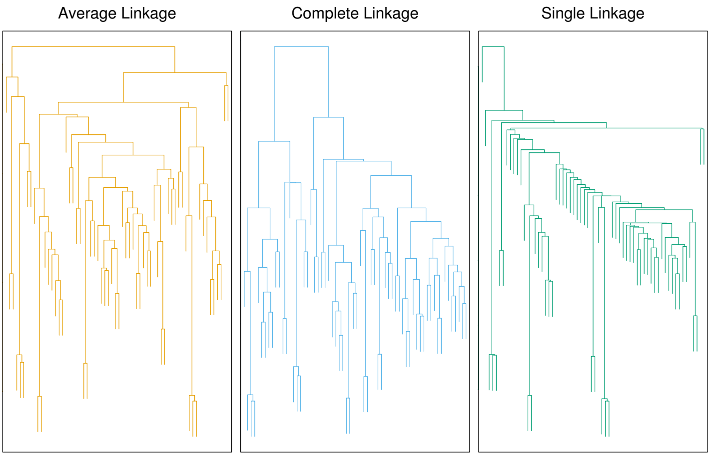
Hierarchical clustering



Centroid linkage:

- The distance between 2 clusters is the distance between each centroid.
- *Suffers from inversions*

Hierarchical clustering



Examples of hierarchical clustering using different linkages.

Clustering

Clustering is riddled with questions and choices

- Is clustering appropriate? i.e. Could a sample belong to more than one cluster?
  - Mixture models, soft clustering, topic models.

Clustering

Clustering is riddled with questions and choices

- Is clustering appropriate? i.e. Could a sample belong to more than one cluster?
  - Mixture models, soft clustering, topic models.
- How many clusters are appropriate?
  - Choose subjectively — depends on the inference sought.
  - Some formal methods based on gap statistics, mixture models, etc.

Clustering

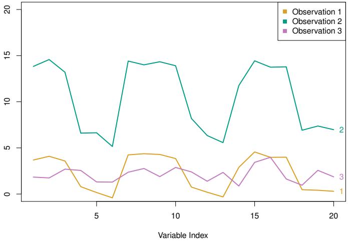
Clustering is riddled with questions and choices

- Is clustering appropriate? i.e. Could a sample belong to more than one cluster?
  - Mixture models, soft clustering, topic models.
- How many clusters are appropriate?
  - Choose subjectively — depends on the inference sought.
  - Some formal methods based on gap statistics, mixture models, etc.
- Are the clusters robust?
  - Run the clustering on different random subsets of the data. Is the structure preserved?
  - Try different clustering algorithms. Are the conclusions consistent?
  - Most important: temper your conclusions.

Questions on distance

- Should we scale the variables before doing the clustering.
  - Variables with larger variance have a larger effect on the Euclidean distance between two samples.
- Does Euclidean distance capture dissimilarity between samples?

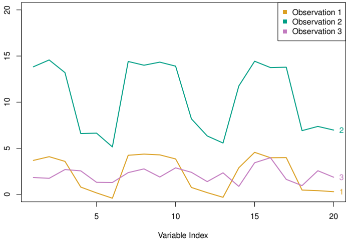
Correlation distance



Example: Suppose that we want to cluster customers at a store for market segmentation.

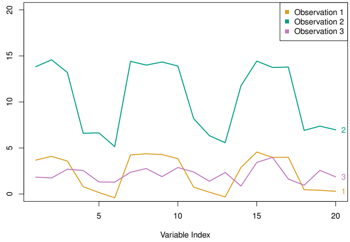
- Samples are customers
- Each variable corresponds to a specific product and measures the number of items bought by the customer during a year.

Correlation distance



Example: Suppose that we want to cluster customers at a store for market segmentation.

- We **could** use Euclidean distance
  - Would cluster all customers who purchase few things (orange and purple)



Example: Suppose that we want to cluster customers at a store for market segmentation.

- We **could** use Euclidean distance
  - Would cluster all customers who purchase few things (orange and purple)
- **What if**: we want to cluster customers who purchase *similar* things?
  - *Correlation distance* may be a more appropriate measure

References

[1] ISL. Chapters 2.2.3, 10.3

[2] ESL. Chapter 6.6.3