STA314 Summer 2023

Week 5: Clustering & Introduction to Bayesian method

Ziang Zhang

Department of Statistics, University of Toronto

Overview

Supervised and unsupervised learning

- Recall there are two types of ML methods: supervised and unsupervised.
- Supervised method uses the *labelled* data: $\{y_i, \mathbf{x}_i\}$, and aims to learn $P(y_i|\mathbf{x}_i)$.
- Based on $P(y_i|\mathbf{x}_i)$, supervised method makes prediction for a new outcome variable.
- On the other hand, un-supervised method uses the *unlabelled* data: $\{\mathbf{x}_i\}$, and aims to learn $P(\mathbf{x}_i)$.
- The target is to discover hidden structure within the distribution $P(\mathbf{x}_i)$.

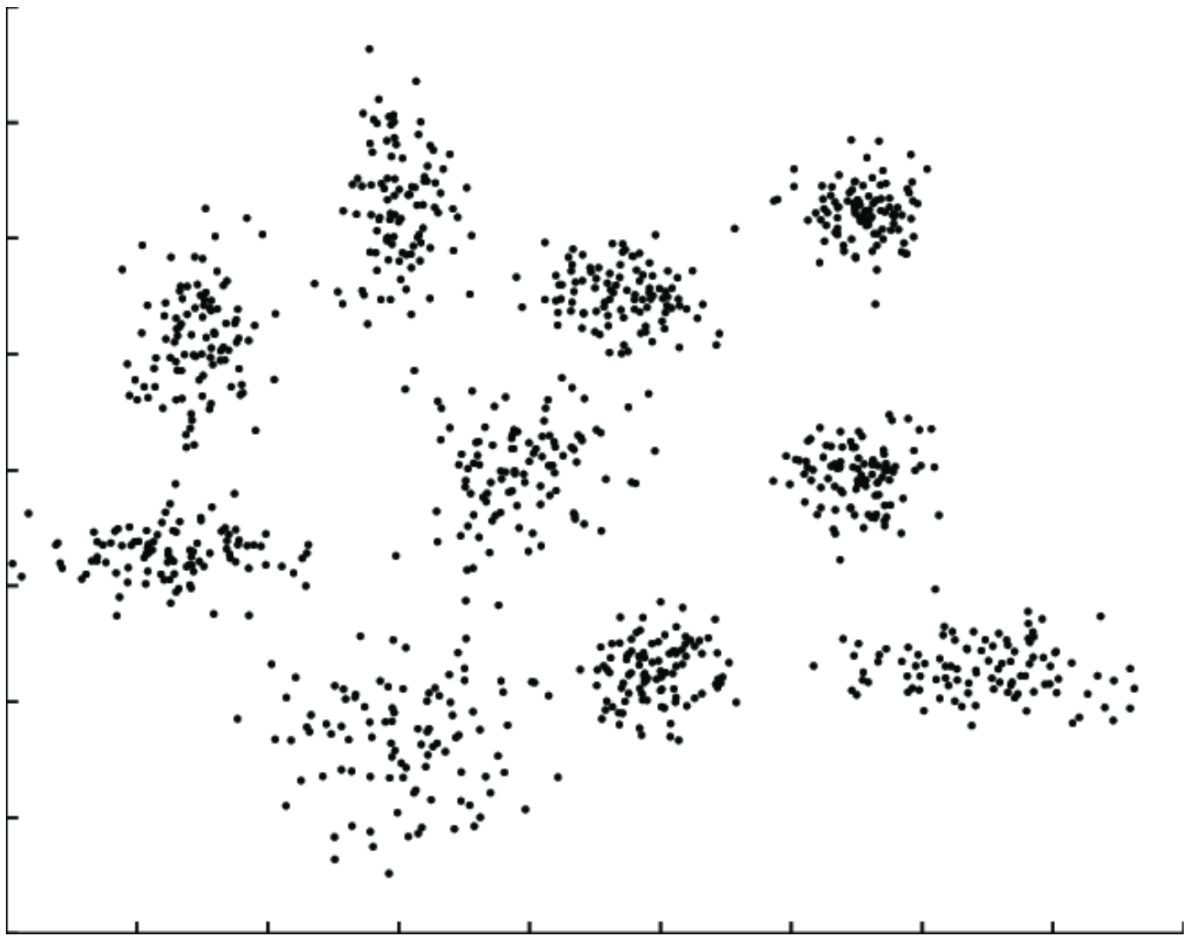
What is clustering?

- Previously, we have learned one unsupervised learning method, the PCA.
- Today, we will learn another type of unsupervised learning method, called the *clustering*.
- Clustering refers to a very broad set of techniques for finding *subgroups*, or *clusters*, in a data set.

Example

- Suppose we collect n observations of tissue samples of breast cancer patients: $\{\mathbf{x}_i\}_{i=1}^n$.
- Each observation $\mathbf{x}_i \in \mathbb{R}^p$ denotes the measurements collected for each tissue sample.
- For example, $\mathbf{x}_{i1}$ could be tumor size; $\mathbf{x}_{i2}$ could be gene expression...
- We may have a reason to believe that there is some heterogeneity among the n tissue samples.
- Perhaps there are a few different *unknown* subtypes of breast cancers...

Example



Clustering vs PCA

Both clustering and PCA learn and analyze the structure of $p(\mathbf{x})$, but their mechanisms are different:

- PCA looks to find a low-dimensional representation of the observations that explain a good fraction of the variance.
- Clustering looks to find *homogeneous subgroups* among the observations.

Clustering vs Classification

Both clustering and Classification try to put similar observations into the same group, but:

- In classification problem, you *know* which group (label) each training observation belongs to.
- In clustering problem, you *don't know* which group each observation comes from. In certain case, you might not even know how many groups exist in total!

Intuition

Suppose we know that there are likely K clusters (groups) of observations:

- Let each of $C_1, ..., C_K$ denote sets containing the indices of the observations in each cluster, such that:
  1. $C_1 \cup C_2 ... \cup C_K = \{1, 2, ..., n\}$
  2. $C_k \cap C_{k'} = \emptyset$ for $k \neq k'$
- How do we decide the observations in each set of the indices?
- A good clustering algorithm should ensure the *within-cluster variation* of each cluster C_k is small.

The notion of distance

More formally, we hope to minimize the following:

$$\sum_{k=1}^{K} W(C_k),$$

where $W(C_k)$ is the within-cluster variation of C_k.

The default choice is typically *squared Euclidean distance*:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,j \in C_k} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = \frac{1}{|C_k|} \sum_{i,j \in C_k} \sum_{d=1}^{p} (x_{id} - x_{jd})^2,$$

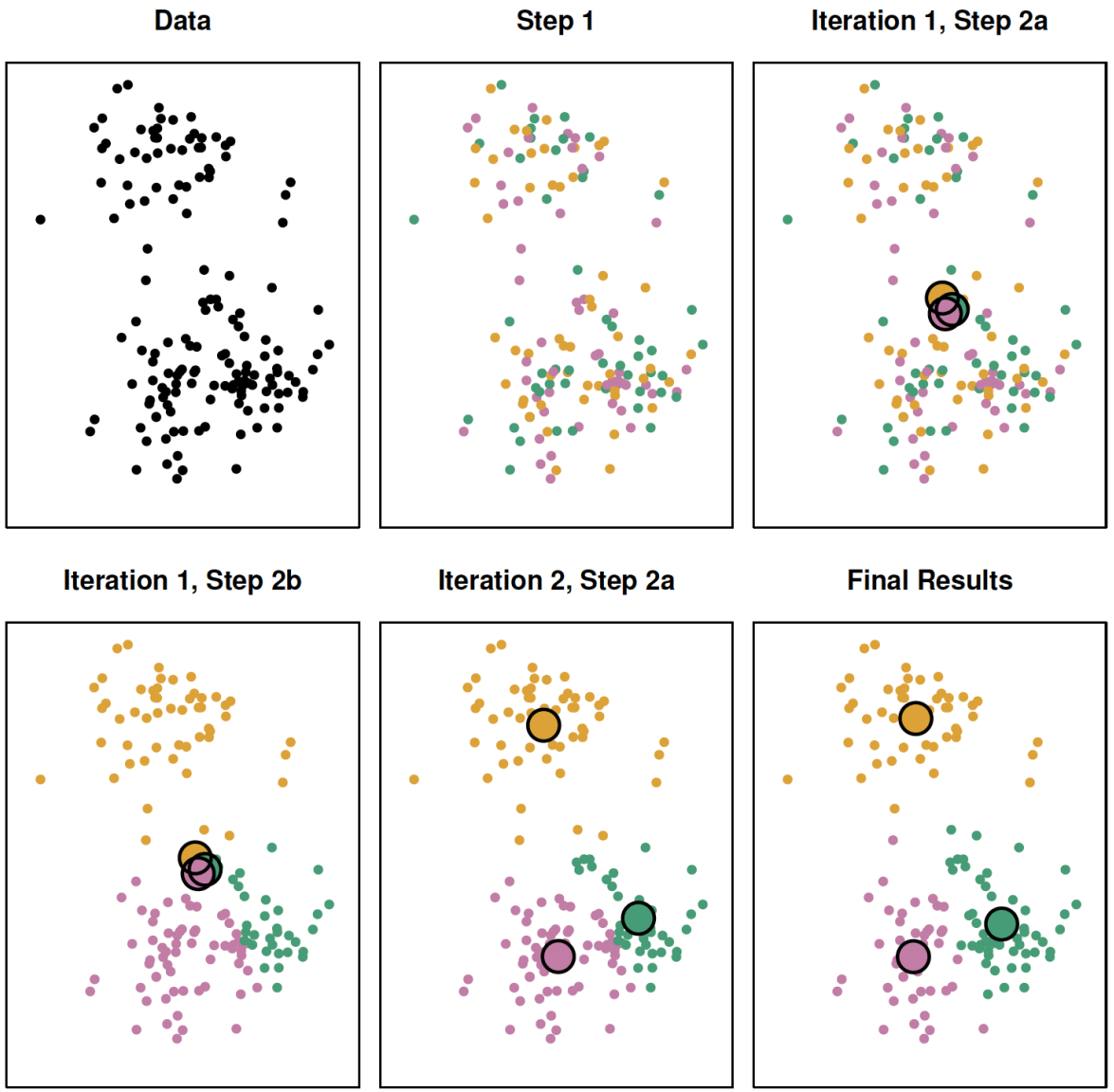
where $|C_k|$ is the number of observations in C_k.

K-means algorithm

The K-means algorithm proceeds as follows:

1. **Initialization:** Randomly assign a number from 1 to K to each of the observations. These serve as the initial cluster assignments for the observations.
2. **Iteration:** Until the cluster assignments stop changing:
   - **Updating (Refitting):** For each of the K clusters, compute the *cluster centroid*.
     (The kth cluster centroid is typically the mean for the observations in the kth cluster)
   - **Assignment:** Assign each observation to the cluster whose centroid is closest.
     (where *closest* is typically defined using Euclidean distance)

Illustration



Local optimum

The K-means algorithm guarantees the objective function $\sum_{k=1}^{K} W(C_k)$ to not increase in each iteration.

To understand that, note the following identity:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,j \in C_k} \|\mathbf{x}_i - \mathbf{x}_j\|^2 = 2 \sum_{i \in C_k} \|\mathbf{x}_i - \bar{\mathbf{x}}_k\|^2,$$

where $\bar{\mathbf{x}}_k$ denotes the sample mean (centroid) for cluster k.

- The update step computes the correct mean in each cluster.
- The assignment step will reduce each $\|\mathbf{x}_i - \bar{\mathbf{x}}_k\|^2$.
- Since each observation is assigned to the closest centroid.

Exercise: Prove the above identity.

Local optimum

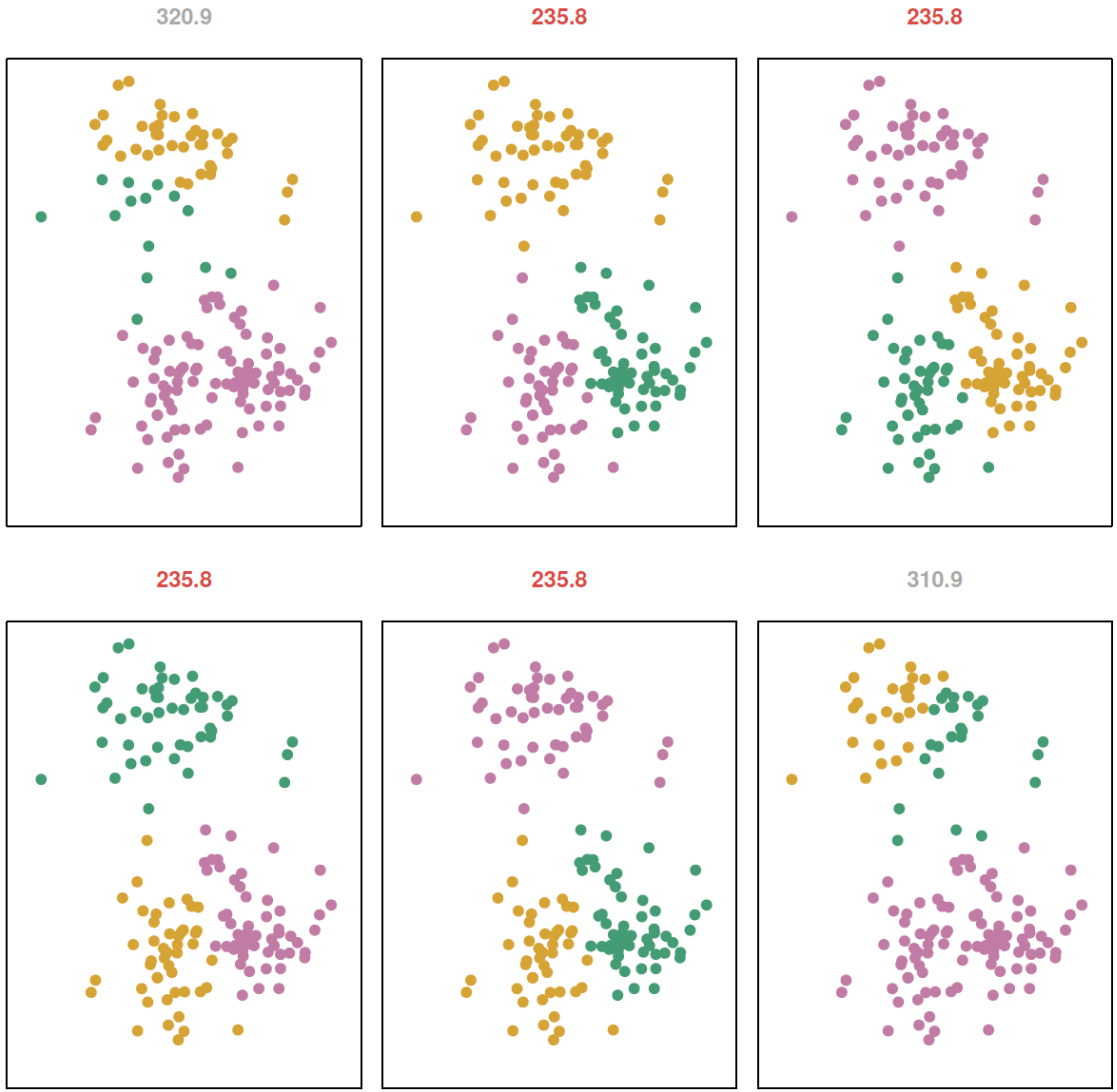
- The K-means algorithm iteratively reduces the objective function.
- Once the algorithm converges, it reaches a local minimum.
- Because the K-means algorithm finds a local rather than a global optimum, the results obtained will heavily depend on the *initialization*.

Multiple starts

One way to address this problem is through multiple starts:

- Randomly initialize the algorithm multiple times.
- For each initialization, run the K-means algorithm.
- Each run will likely result in a different final set of clusters.
- Choose the result with the *lowest* within-cluster variation.
- This method can reduce the chance of obtaining a poor local minimum, at the cost of computational time.

Multiple starts: Illustration



The limitation of K-means

- To use the K-means algorithm, you need to pre-specify the number of clusters.
- In practice, it is often hard to determine the K.
- *Hierarchical Clustering* is an alternative that does not require the number of clusters to be pre-specified.
- In Hierarchical clustering, the clusters are obtained from cutting the *Dendrogram* at a given height.

The Dendrogram

The Dendrogram is a tree-based representation of observations.

- Each unique observation in the dataset is a leaf in the tree.
- As we move up the tree, some leaves begin to *fuse* into branches.
- As we move higher up the tree, branches themselves fuse, either with leaves or other branches.
- The earlier (lower in the tree) fusions occur, the more similar the groups of observations are to each other.
- The height of this fusion, as measured on the vertical axis, indicates how different the two observations are.
- This is also called bottom-up or agglomerative clustering.

The Dendrogram

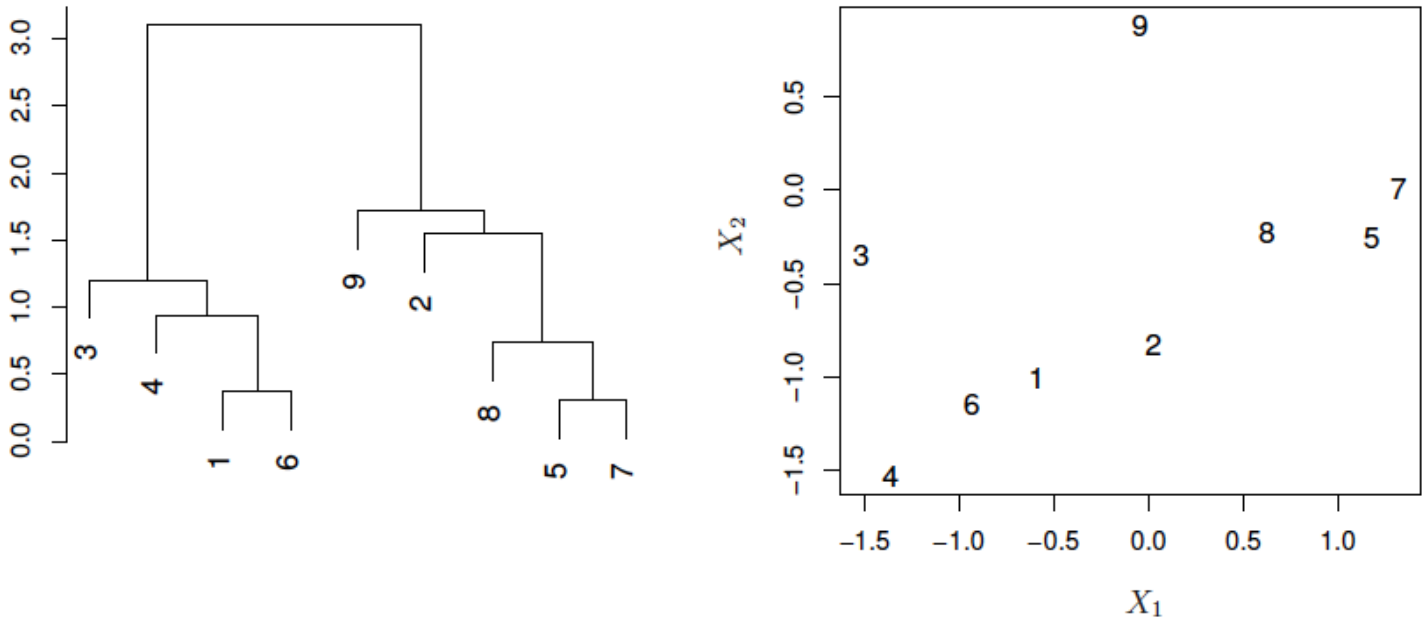


FIGURE 12.12. *An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space.* Left: *a dendrogram generated using Euclidean distance and complete linkage. Observations* 5 *and* 7 *are quite similar to each other, as are observations* 1 *and* 6. *However, observation* 9 *is* no more similar to *observation* 2 *than it is to observations* 8, 5, *and* 7, *even though observations* 9 *and* 2 *are close together in terms of horizontal distance. This is because observations* 2, 8, 5, *and* 7 *all fuse with observation* 9 *at the same height, approximately* 1.8. Right: *the raw data used to generate the dendrogram can be used to confirm that indeed, observation* 9 *is no more similar to observation* 2 *than it is to observations* 8, 5, *and* 7.

Inter-cluster dissimilarity

To actually construct a Dendrogram, we need a measure of dissimilarity.

- Defining the dissimilarity for a pair of observations is easy.
- But how do we generalize the notion of dissimilarity for a pair of *clusters*?
- For that, we need a notion called Linkage.

Linkage

Linkage	*Description*
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

TABLE 12.3. *A summary of the four most commonly-used types of linkage in hierarchical clustering.*

Hierarchical Clustering Algorithm

The Hierarchical Clustering Algorithm can be outlined as follows:

1. Begin with n observations and a measure (such as Euclidean distance) of all the $n(n-1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.
2. For $i = n, n-1, \ldots, 2$:
   - Identify the pair of clusters that are closest to each other, and merge them. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion line should be drawn.
   - Compute the new pairwise inter-cluster dissimilarities among the remaining $i-1$ clusters.
3. Draw a horizontal line across the dendrogram to obtain the final clusters.

Illustration

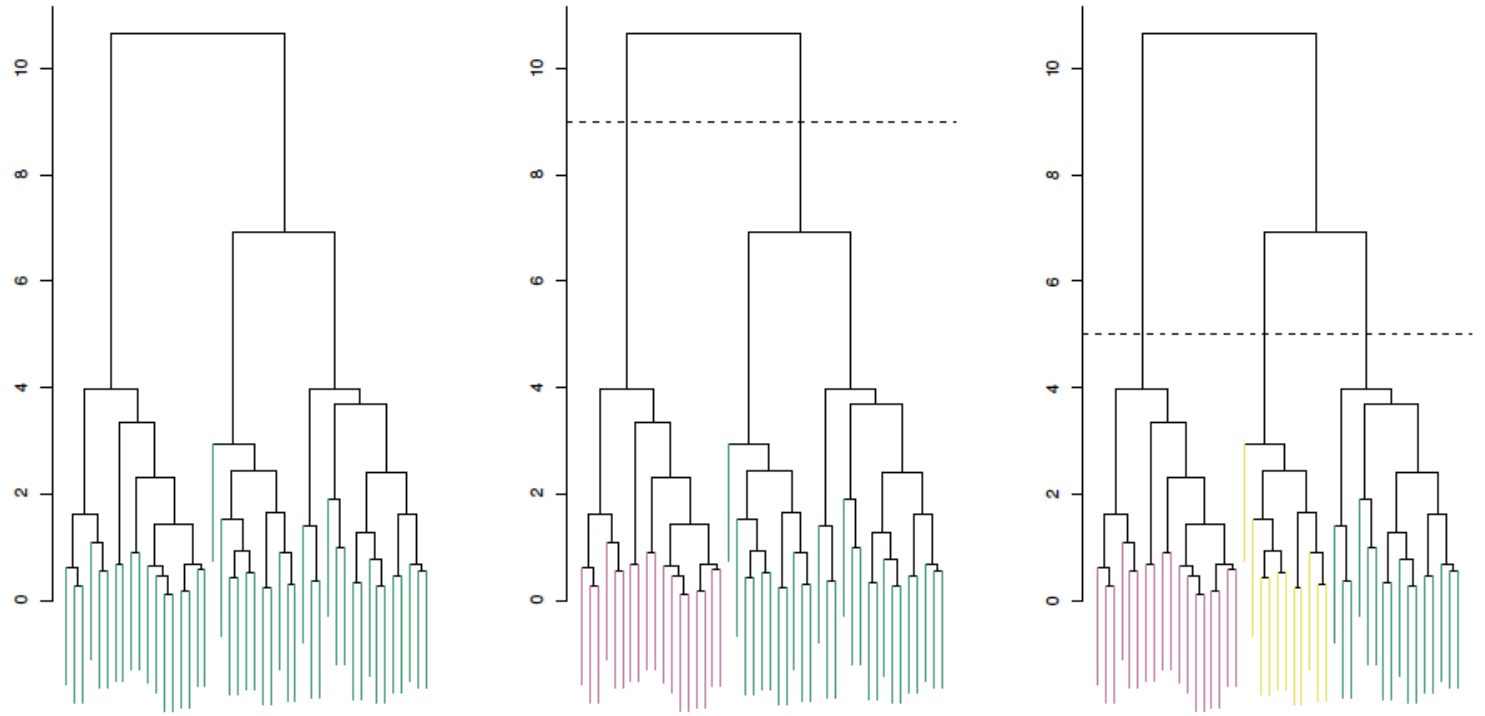


FIGURE 12.11. Left: *dendrogram obtained from hierarchically clustering the data from Figure 12.10 with complete linkage and Euclidean distance.* Center: *the dendrogram from the left-hand panel, cut at a height of nine (indicated by the dashed line). This cut results in two distinct clusters, shown in different colors.* Right: *the dendrogram from the left-hand panel, now cut at a height of five. This cut results in three distinct clusters, shown in different colors. Note that the colors were not used in clustering, but are simply used for display purposes in this figure.*

Illustration

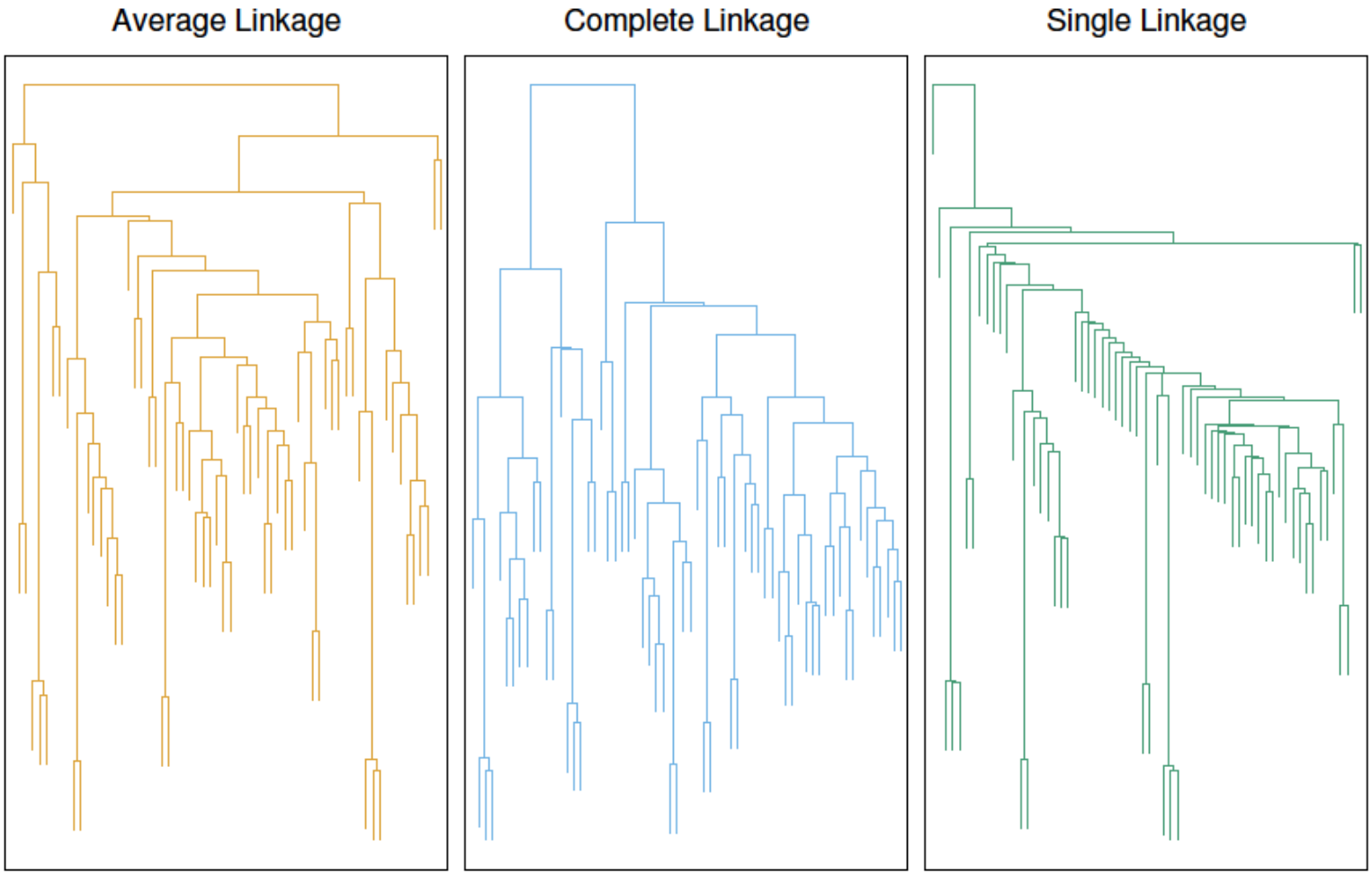


FIGURE 12.14. *Average, complete, and single linkage applied to an example data set. Average and complete linkage tend to yield more balanced clusters.*

Practical issues with clustering

There are many choices to make in practical clustering problem:

- For Hierarchical clustering:
  - Which linkage should be used?
  - Where should we cut the dendrogram?
- For K-means clustering:
  - How many K should we use?
  - How many multiple starts should we do?
- For both methods:
  - Should all the features be standardized first?
  - What dissimilarity measure should we use?

What is Bayesian inference?

Bayesian methods rely on the use of the Bayes theorem.
Let Z be the latent variable, Y be the observed variable:

$$P(Z = z|Y = y) = \frac{P(Z = z, Y = y)}{P(Y = y)} = \frac{P(Y = y|Z = z)P(Z = z)}{P(Y = y)}.$$

- $P(Z = z)$ is the prior, which is our belief of Z before seeing any data.
- $P(Y = y|Z = z)$ is the likelihood function.
- The inference is based on the *posterior* $P(Z = z|Y = y)$.
- The posterior can be viewed as an update of the prior, once we see the data.

Bayesian method vs Frequentist method

In traditional Frequentist method, a quantity can be an unknown constant (parameter), a random variable, or a known constant.

In a fully Bayesian approach, a quantity is either a random variable, or a known constant.

In other words, if a parameter is unknown, it should be assigned with a prior to describe our prior belief for it.

Bayesian method vs Frequentist method

For example, suppose we collect n $i.i.d$ data $\mathbf{y} = (y_1, ..., y_n)$, from a distribution $P(Y|\theta)$ where θ is an unknown parameter.

In Frequentist method, we estimate θ by the MLE:

$$\hat{\theta}_{ML} = \underset{\theta}{\operatorname{argmax}}\, L(\mathbf{y}|\theta) = \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{n} P(y_i|\theta).$$

In Bayesian method, we estimate θ with:

$$P(\theta|\mathbf{y}) = \frac{L(\mathbf{y}|\theta)P(\theta)}{P(\mathbf{y})} \propto L(\mathbf{y}|\theta)P(\theta).$$

The point estimate can then be obtained by maximizing $P(\theta|\mathbf{y})$, which is called the MAP.

Bayesian method vs Frequentist method

To predict a new data y_{new}, Frequentist method uses:

$$\hat{y}_{new} = \mathbb{E}(Y|\hat{\theta}_{ML}),$$

where the expectation is taken over $Y \sim P(Y|\theta = \hat{\theta}_{ML})$.

In Bayesian method, one can compute the **posterior predictive distribution**:

$$\begin{aligned} P(y_{new}|\mathbf{y}) &= \int_{\theta} P(y_{new}|\theta, \mathbf{y}) P(\theta|\mathbf{y}) d\theta \\ &= \int_{\theta} P(y_{new}|\theta) P(\theta|\mathbf{y}) d\theta. \end{aligned} \tag{1}$$

A point prediction can be obtained as

$$\hat{y}_{new} = \mathbb{E}(Y_{new}|\mathbf{y}).$$

Bayesian Hierarchical Model

Bayesian method also provides a convenient approach to handle the hyper-parameter λ:

$$\begin{aligned} \lambda &\rightarrow \theta \rightarrow \mathbf{y} \\ P(\lambda) &\rightarrow P(\theta|\lambda) \rightarrow L(\mathbf{y}|\theta) \\ P(\lambda, \theta|\mathbf{y}) &= \frac{P(\lambda)P(\theta|\lambda)L(\mathbf{y}|\theta)}{P(\mathbf{y})} \propto P(\lambda)P(\theta|\lambda)L(\mathbf{y}|\theta). \end{aligned} \tag{2}$$

- $P(\lambda)$ is a prior on the hyperparameter.
- λ can be integrated out from the posterior.
- This is a **hierarchical Bayesian model**, also called **multi-level model**.

Empirical Bayes

For Bayesian hierarchical model,

$$P(\lambda, \theta|\mathbf{y}) \propto P(\lambda)P(\theta|\lambda)L(\mathbf{y}|\theta)$$

- In some case, integrating out λ is hard.
- One shortcut is to fix the hyperparameter at its posterior mode:

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}}\, P(\lambda|\mathbf{y}) = \underset{\lambda}{\operatorname{argmax}} \int_{\theta} L(\mathbf{y}|\theta)P(\theta|\lambda)P(\lambda)d\theta$$

- If we further assume the prior on λ is uniform:

$$\begin{aligned} \underset{\lambda}{\operatorname{argmax}}\, P(\lambda|\mathbf{y}) &= \underset{\lambda}{\operatorname{argmax}} \int_{\theta} L(\mathbf{y}|\theta)P(\theta|\lambda)d\theta \\ &= \underset{\lambda}{\operatorname{argmax}}\, L(\mathbf{y}|\lambda). \end{aligned} \tag{3}$$

- This is called the *Empirical Bayes*.

Example: Bayesian linear regression

Suppose we want to do linear regression model given n data points $\{y_i, \mathbf{x}_i\}_{i=1}^n$,

$$y_i = \mathbf{x}_i^T \boldsymbol{\beta} + \epsilon_i,$$

where $\epsilon_i \sim N(0, 1)$.

- Assume the features $\mathbf{x}_i$ is fixed constant, so we can simply write $P(y_i|\mathbf{x}_i, \boldsymbol{\beta}) = P(y_i|\boldsymbol{\beta})$.
- The only unknown parameter is $\boldsymbol{\beta}$, and we use a normal prior:

$$\boldsymbol{\beta} \sim N(\mathbf{0}, \frac{1}{\lambda}\mathbf{I})$$

- Let's compute the MAP estimate of $\boldsymbol{\beta}$.

Example: Bayesian linear regression

Recall the normal density function for a random variable x with mean μ and variance σ^2 is given by:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

For a multivariate case with a d-dimensional random vector $\mathbf{x}$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$:

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}-\boldsymbol{\mu})}.$$

Hence the log likelihood and the log prior are respectively:

$$\log L(\mathbf{y}|\boldsymbol{\beta}) = -\frac{1}{2}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2 \quad \log P(\boldsymbol{\beta}) = -\frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}.$$

Example: Bayesian linear regression

Since $\text{argmax}_\beta P(\boldsymbol{\beta}|\mathbf{y}) = \text{argmax}_\beta \log P(\boldsymbol{\beta}|\mathbf{y})$:

- Maximizing the log posterior (up to a constant) is equivalent to:

$$\log L(\mathbf{y}|\boldsymbol{\beta}) + \log P(\boldsymbol{\beta}) \equiv -\frac{1}{2}\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2 - \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta} \quad (4)$$

- Maximizing the posterior is equivalent to minimizing:

$$\sum_{i=1}^{n}(y_i - \mathbf{x}_i^T\boldsymbol{\beta})^2 + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}.$$

- The Ridge regression is equivalent to the MAP in the Bayesian linear regression with Gaussian prior.

Why use Bayesian methods for ML?

There are many reasons that make Bayesian methods useful in ML:

- The use of the prior allows the existing knowledge to be integrated into the modeling.
- The use of the prior can serve the purpose of model regularization.
- The existence problem of MLE is no longer an issue. You can always compute $P(\theta|\mathbf{y})$ as long as $P(\theta)$ is well-defined.
- The posterior predictive distribution accounts for the uncertainty of *all* the unknown quantities in the model.

Example of Bayesian methods in ML

There are many famous Bayesian methods in the ML field:

- Bayesian Neural Network (BNN)
- Bayesian Network (BN) (STA414)
- Bayesian Optimization
- Bayesian Regression and PCA (STA414)

Difficulty of the Bayesian methods

Recall that

$$P(\theta|\mathbf{y}) = \frac{L(\mathbf{y}|\theta)P(\theta)}{P(\mathbf{y})} \propto L(\mathbf{y}|\theta)P(\theta)$$

The major difficulty in Bayesian methods is the integration:

- If we want the entire distribution $P(\theta|\mathbf{y})$, we need to know the value of the marginal likelihood $P(\mathbf{y})$.
- That requires us to compute $P(\mathbf{y}) = \int_\theta L(\mathbf{y}|\theta)P(\theta)d\theta$.
- In most applications, that integral is not tractable.
- Numerical methods work well for θ with small dimension, but less well when the dimension is large.

Solution in Bayesian methods

$$P(\theta|\mathbf{y}) = \frac{L(\mathbf{y}|\theta)P(\theta)}{P(\mathbf{y})} \propto L(\mathbf{y}|\theta)P(\theta)$$

There are three main approaches to tackle this problem:

1. Markov Chain Monte Carlo (MCMC)
2. Approximation method such as Variational Inference.
3. Conjugate Prior

The first two methods are the main topics in STA414.
We will only cover the easiest approach through the use of conjugate prior.

Conjugacy

Given a likelihood function, We say the prior is a conjugate prior for that likelihood *when the prior and the posterior have the same form.*

Conjugacy

For example, if the likelihood is binomial:

$$L(y|\theta) = \binom{n}{y}\theta^y(1-\theta)^{n-y} \propto \theta^y(1-\theta)^{n-y}.$$

And we use a beta-prior for θ:

$$P(\theta) = \text{Beta}(\theta|\alpha, b) = \frac{\theta^{\alpha-1}(1-\theta)^{b-1}}{B(\alpha, b)} \propto \theta^{\alpha-1}(1-\theta)^{b-1},$$

where $B(\alpha, b) = \int_0^1 \theta^{\alpha-1}(1-\theta)^{b-1}\, d\theta$.

The posterior distribution must be proportional to:

$$\begin{aligned} P(\theta|y) &\propto P(\theta)L(y|\theta) \\ &\propto \theta^{\alpha-1}(1-\theta)^{b-1}\theta^y(1-\theta)^{n-y} \\ &\propto \theta^{y+\alpha-1}(1-\theta)^{n-y+b-1} \end{aligned} \tag{5}$$

Conjugacy

We have shown that $P(\theta|y) \propto \theta^{y+\alpha-1}(1-\theta)^{n-y+b-1}$.

We just need to normalize the density:

$$P(\theta|y) = \frac{\theta^{y+\alpha-1}(1-\theta)^{n-y+b-1}}{\int_0^1 \theta^{y+\alpha-1}(1-\theta)^{n-y+b-1}d\theta}.$$

However, we don't need to compute the normalization constant, since $\theta^{y+\alpha-1}(1-\theta)^{n-y+b-1}$ is the functional form of Beta distribution.

The functional form uniquely determines the distribution, since any distribution must integrate to 1.

Conjugacy

In conclusion, we know that:

$$P(\theta|y) = \text{Beta}(\theta|\alpha^*, b^*) = \frac{\theta^{\alpha^*-1}(1-\theta)^{b^*-1}}{B(\alpha^*, b^*)},$$

where $\alpha^* = y + \alpha, \quad b^* = n - y + b.$

Specifically, we know that:

$$\int_0^1 \theta^{y+\alpha-1}(1-\theta)^{n-y+b-1} d\theta = B(\alpha^*, b^*).$$

Hence, Beta distribution is the conjugate prior to the binomial likelihood.
Quick Exercise: Is Beta prior still conjugate to the Bernoulli likelihood?

Example: Bayesian linear regression

Now let's return to our Bayesian linear regression example.

This time, instead of just finding its MAP, let's find the entire posterior distribution.

Derivation of Posterior Distribution - Part 1

Recall our setup:

- Likelihood: $L(\mathbf{y}|\boldsymbol{\beta}) = \prod_{i=1}^{n} N(y_i|\mathbf{x}_i^T\boldsymbol{\beta}, 1)$.
- Prior: $P(\boldsymbol{\beta}) = N(\boldsymbol{\beta}|\mathbf{0}, \frac{1}{\lambda}\mathbf{I})$.

The posterior distribution is proportional to

$$P(\boldsymbol{\beta}|\mathbf{y}) \propto L(\mathbf{y}|\boldsymbol{\beta})P(\boldsymbol{\beta}) = \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) - \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right). \tag{6}$$

Derivation of Posterior Distribution - Part 2

Expanding the squares in the exponential, we get:

$$\begin{aligned} P(\boldsymbol{\beta}|\mathbf{y}) &\propto \exp\left(-\frac{1}{2}(\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} - 2\boldsymbol{\beta}^T\mathbf{X}^T\mathbf{y} + \mathbf{y}^T\mathbf{y}) - \frac{\lambda}{2}\boldsymbol{\beta}^T\boldsymbol{\beta}\right) \\ &\propto \exp\left(-\frac{1}{2}\boldsymbol{\beta}^T(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\boldsymbol{\beta} + \boldsymbol{\beta}^T\mathbf{X}^T\mathbf{y}\right). \end{aligned} \tag{7}$$

Note that the density of a general multivariate Gaussian can be written as:

$$\begin{aligned} f(\mathbf{x}) &\propto \exp\left(-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu})\right) \\ &\propto \exp\left(-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} + \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}\right). \end{aligned} \tag{8}$$

Derivation of Posterior Distribution - Part 3

The above expression implies that Gaussian prior is a conjugate prior to the Gaussian likelihood, hence

$$P(\boldsymbol{\beta}|\mathbf{y}) = N(\boldsymbol{\beta}|\mu^*, \Sigma^*),$$

for solve posterior mean μ^* and posterior covariance Σ^*.

Solving the above, we get:

- Posterior Covariance Σ^*:

$$\Sigma^* = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}.$$

- Posterior Mean μ^*:

$$\mu^* = \Sigma^*\mathbf{X}^T\mathbf{y} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}.$$

For the final exam

- The final will be held at Aug 23: 7:00 - 10:00 PM, at the room EX100.
- It will cover all the materials from lecture 1 to lecture 5.
- The setting will be similar to the midterm: the only allowed aid is one non-programmable calculator.
- An representative practice final will be posted soon.

For the final exam

- There will also be additional instructor/TA office hour before the final exam.
- Thank you for all the efforts that you have put into this course!
- Good luck with your final exam!