Hierarchical Models

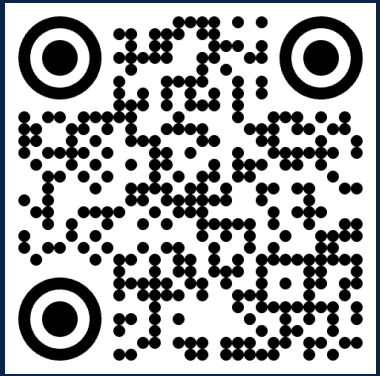
The Advanced R Series: Clustering & Classification

Slides and code created by Jesse Ghashti

January 14, 2026

Centre for Scholarly Communication
The University of British Columbia | Okanagan Campus | Syilx Okanagan Nation Territory

New Here?

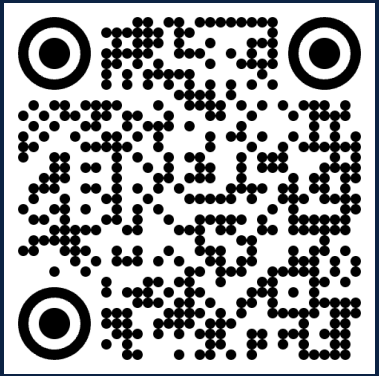


<——————-

Check out our other CSC workshops!

GitHub code and slides for today's workshop (and previous workshops)

——————>



Alternatively, code/slides available at the bottom of `https://csc-ubc-okanagan.github.io/workshops/`

Workshop Series Overview

Session	Topic	Date/Time
1	**Hierarchical Models**	**Jan 14, 12:00 PM**
2	Centroid-Based Models	Jan 19, 3:30 PM
3	Fuzzy Clustering	Jan 28, 1:00 PM
4	Distribution-Based Models	Feb 2, 3:30 PM
5	Density-Based Models	Feb 9, 3:30 PM
6	Graph-Based Models	Feb 23, 3:30 PM
7	Mixed-Type Data Quantification	Mar 4, 1:00 PM
8	Mixed-Type Data Clustering	Mar 11, 1:00 PM
9	Bias Reduction and Fairness	Mar 18, 1:00 PM
10	Dimensionality Reduction 1 of 2	Mar 23, 3:30 PM
11	Dimensionality Reduction 2 of 2	Mar 30, 3:30 PM

From Regression to Clustering & Classification

Where We Left Off (Regression)

Last term, we focused on **regression models**:

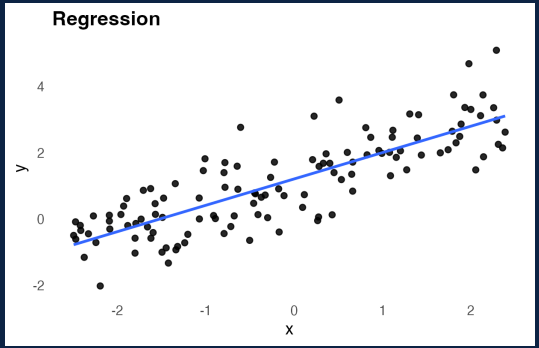
- Predict an **outcome** y from features X
- Learn a relationship: $y \approx f(X)$
- Evaluate with prediction error (MSE, R^2, …)

Now: Different Questions

In clustering and classification, we often ask

- **Classification:** given labelled examples, can we predict the correct **category**?
- **Clustering:** with no labels, can we discover **groups/structure** in the data?

Regression



We are not always predicting a response anymore — sometimes we predict a **label**, and sometimes there are **no labels at all**.

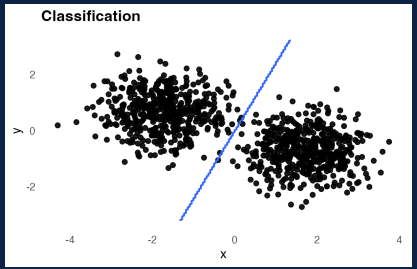
Clustering vs Classification

Classification (Supervised)

Data features X and known labels y
Goal learn a rule $\hat{y} = f(X)$ to predict labels

Common Uses

- Species identification
- Disease status (positive/negative)
- Habitat type classification



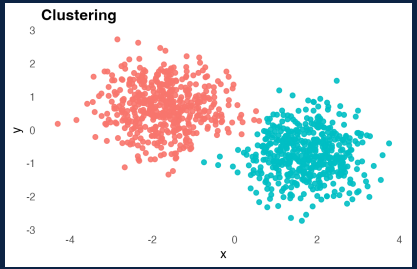
Here labels are known, so we can measure performance.

Clustering: Learning Structure Without Labels

Clustering (Unsupervised)

Data features X only (no labels) **Goal** partition observations into groups that are "similar"

- No ground-truth categories
- Many plausible answers
- Evaluate internally: cohesion vs separation



Is it more challenging?

- "Correct" clusters may not exist
- Choice of k is not uniquely determined

Today: Hierarchical Clustering

Today we will...

- Understand how different linkage methods work
- Learn the elbow method and silhouette analysis for choosing k
- Apply hierarchical clustering to the Iris dataset
- Compare Complete and Ward linkage methods
- Evaluate clustering as a classification problem
- Calculate ARI and clustering accuracy metrics

Today we require...

```
library('ggplot2')     # visualization
library('cluster')     # clustering algorithms
library('factoextra') # clustering visualization
library('dplyr')       # data manipulation
library('mclust')      # for ARI (Adjusted Rand Index)
```

Understanding Linkage Methods

The Core Question

In hierarchical clustering, we merge the two **closest** clusters at each step.
But how do we measure distance between **clusters** (not just points)?

Five common approaches:

1. Single Linkage (minimum distance)
2. Complete Linkage (maximum distance)
3. Average Linkage (mean distance)
4. Ward's Method (minimize variance)
5. DIANA (divisive, not agglomerative)

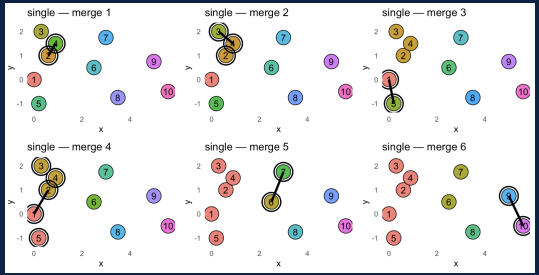
Single Linkage: Nearest Neighbour

Definition

$$d(A, B) = \min_{i \in A, j \in B} d(i, j)$$

Distance = shortest connection

- Finds closest pair of points and merge.
- Can detect elongated clusters
- Prone to “chaining” effect



Single linkage often creates long chains instead of compact clusters

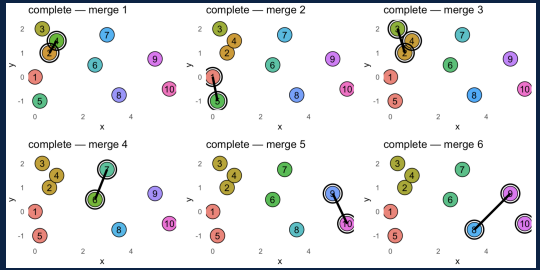
Complete Linkage: Farthest Neighbour

Definition

$$d(A, B) = \max_{i \in A, j \in B} d(i, j)$$

Distance = longest connection

- Finds farthest pair of points and merges
- Creates compact, spherical clusters
- Sensitive to outliers
- Good for well-separated groups



Complete linkage tends to produce clusters with similar diameters

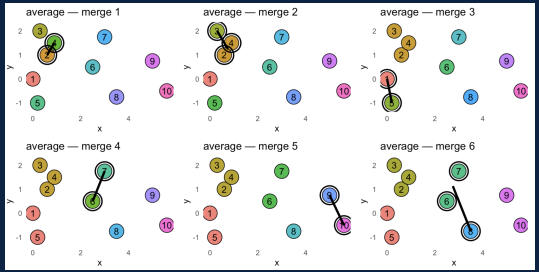
Average Linkage: Compromise

Definition

$$d(A, B) = \frac{1}{|A||B|} \sum_{i \in A} \sum_{j \in B} d(i, j)$$

Distance = mean of all pairs

- Considers all pairwise distances
- Compromise between single/complete
- More robust to outliers



Average linkage connects clusters based on their centroids

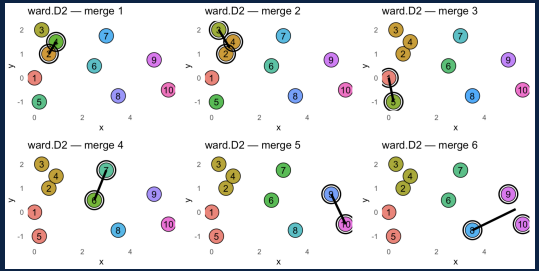
Ward's Method: Minimum Variance

Definition

$$\Delta(A, B) = \frac{|A||B|}{|A| + |B|} ||\bar{x}_A - \bar{x}_B||^2$$

Minimize within-cluster variance

- Creates compact, spherical clusters
- Similar to k-means (later)
- Assumes equal cluster sizes



Often the most popular choice: often produces the most interpretable clusters

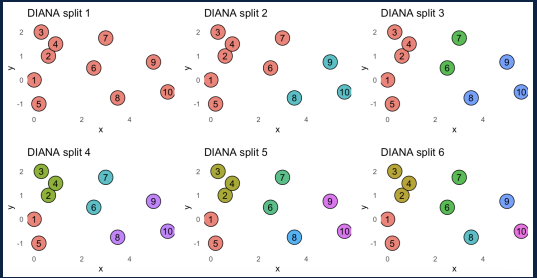
DIANA: Divisive Approach

Top-Down Clustering

Algorithm:

1. Start with all points in one cluster
2. Find ‘best’ split into two
3. Choose cluster to split next
4. Repeat until k clusters

- Opposite of agglomerative
- Can find main structure first
- Computationally expensive



Choosing k: The Elbow Method

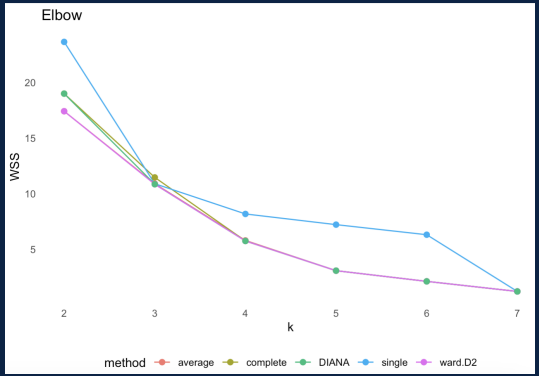
Within-Sum-of-Squares

$$WSS = \sum_{k=1}^{K} \sum_{i \in C_k} ||x_i - \bar{x}_k||^2$$

The idea

- Plot WSS vs number of clusters
- Look for “elbow” in the curve
- Diminishing returns after elbow
- Balances complexity vs fit

WSS always decreases with more clusters - look for where it levels off



Choosing k: Silhouette Analysis

Silhouette Width

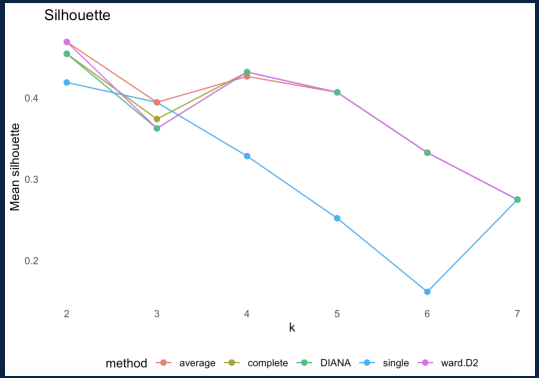
For each point i:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

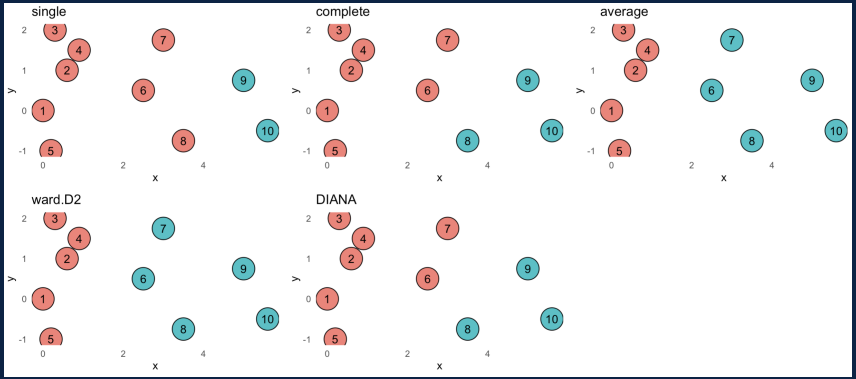
- $a(i)$ = avg distance within cluster
- $b(i)$ = min avg distance to other clusters

Interpretation

- $s(i) \approx 1$: well clustered
- $s(i) \approx 0$: on boundary
- $s(i) < 0$: wrong cluster

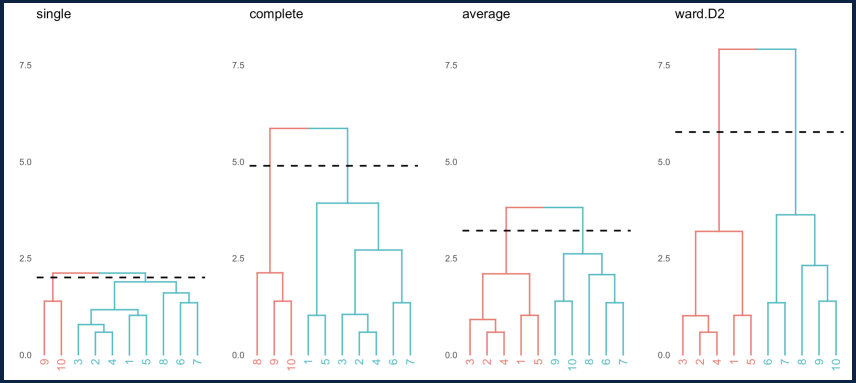


Toy Example Final Clusters



Notice how different linkage methods produce different clusterings from the same data.

Toy Example Dendrograms



Dendrograms show the hierarchical structure — cut at different heights for different k

Application: Iris Dataset

Why Iris Without Scaling?

All measurements in centimeters

- Sepal Length (cm)
- Sepal Width (cm)
- Petal Length (cm)
- Petal Width (cm)

Same units = comparable magnitudes

- No need to standardize
- Preserves biological relationships
- Natural interpretation of distances

We'll focus on Complete and Ward linkage

Iris Analysis: Setup

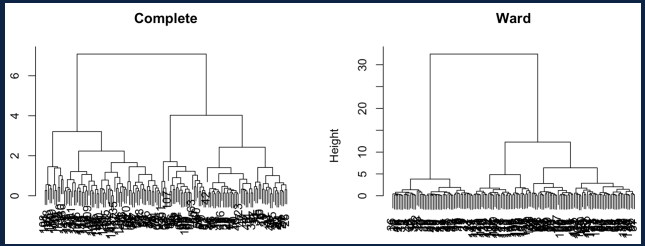
```r
# Load and prepare data
library(ggplot2)
library(cluster)
library(factoextra)
library(dplyr)

# Use iris dataset
X <- as.matrix(iris[, 1:4])
trueLabels <- iris$Species

# Create distance matrix
D <- dist(X, method = "euclidean")

# Fit Complete and Ward
hcComplete <- hclust(D, method = "complete")
hcWard <- hclust(D, method = "ward.D2")

# Plot dendrograms
plot(hcComplete, main = "Complete Linkage",
     xlab = "", sub = "")
plot(hcWard, main = "Ward's Method",
     xlab = "", sub = "")
```



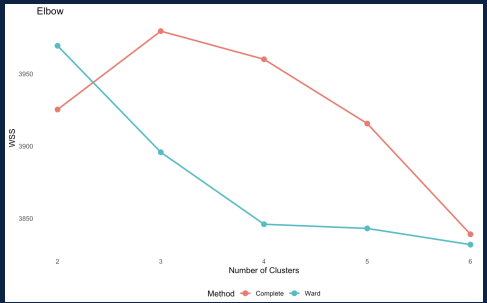
Determining best k

```r
# Function to calculate WSS
calcWSS <- function(X, clusters) {
  wss <- 0
  for(k in unique(clusters)) {
    clusterData <- X[clusters == k, ]
    center <- colMeans(clusterData)
    wss <- wss + sum(rowSums(
      (clusterData - center)^2))
  }
  return(wss)
}
# Evaluate k = 2 to 6
kVals <- 2:6
wssComplete <- numeric(length(kVals))
wssWard <- numeric(length(kVals))

for(i in seq_along(kVals)) {
  k <- kVals[i]
  clComplete <- cutree(hcComplete, k)
  clWard <- cutree(hcWard, k)
  wssComplete[i] <- calcWSS(X, clComplete)
  wssWard[i] <- calcWSS(X, clWard)
}
```

```r
# Create elbow plot
elbowData <- data.frame(k = rep(kVals, 2),
  WSS = c(wssComplete, wssWard),
  Method = rep(c("Complete", "Ward"), each = length(kVals)))

ggplot(elbowData, aes(x = k, y = WSS, color = Method)) +
  geom_line(size = 1) + geom_point(size = 3) +
  labs(title = "Elbow",
       x = "Number of Clusters",
       y = "Within-Sum-of-Squares") +
  theme_minimal()
```



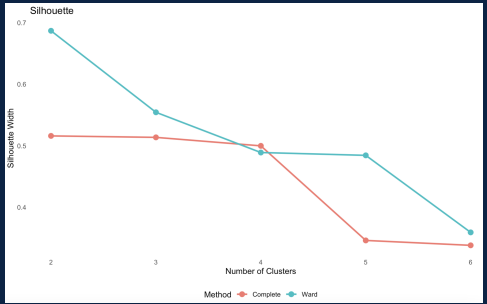
Silhouette Analysis

```r
# Calculate silhouette for each k
silComplete <- numeric(length(kVals))
silWard1 <- numeric(length(kVals))

for(i in seq_along(kVals)) {
  k <- kVals[i]
  # Complete
  clComplete <- cutree(hcComplete, k)
  silComp <- silhouette(clComplete, D)
  silComplete[i] <- mean(silComp[, 3])
  # Ward
  clWard <- cutree(hcWard, k)
  silWard <- silhouette(clWard, D)
  silWard1[i] <- mean(silWard[, 3])
}
# Find best k (max silhouette)
kOptComp <- kVals[which.max(silComplete)]
kOptWard <- kVals[which.max(silWard1)]
```

```r
# Plot silhouette values
silData <- data.frame(k = rep(kVals, 2),
  Silhouette = c(silComplete, silWard1),
  Method = rep(c("Complete", "Ward"), each = length(kVals)))

ggplot(silData, aes(x = k, y = Silhouette, color = Method)) +
  geom_line(size = 1) + geom_point(size = 3) +
  labs(title = "Silhouette Analysis",
       x = "Number of Clusters",
       y = "Mean Silhouette Width") +
  theme_minimal()
```

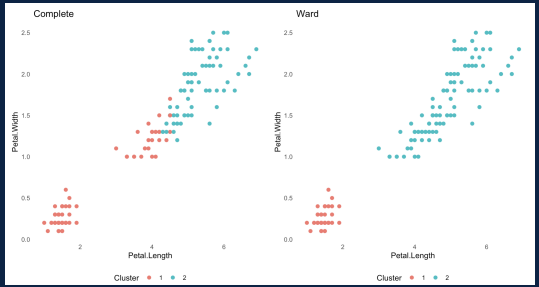


Final Clustering Results

```r
kFinal <- 2
clusComplete <- cutree(hcComplete, kFinal)
clusWard <- cutree(hcWard, kFinal)

irisPlot <- iris
irisPlot$clusterComplete <- factor(clusComplete)
irisPlot$clusterWard <- factor(clusWard)

# plot complete linkage
p1 <- ggplot(irisPlot, aes(x = Petal.Length, y = Petal.Width,
      color = clusterComplete)) +
  geom_point(size = 2) +
  labs(title = "Complete Linkage (k=2)",
       color = "Cluster") +
  theme_minimal()
# plot Ward
p2 <- ggplot(irisPlot, aes(x = Petal.Length, y = Petal.Width,
      color = clusterWard)) +
  geom_point(size = 2) +
  labs(title = "Ward's Method (k=2)",
       color = "Cluster") +
  theme_minimal()
```



Clustering as a Classification Problem

When when we know true labels...

Since Iris has known species labels, we can evaluate clustering as a classification problem:

Metrics

- **Clustering Accuracy** Match predicted labels to true labels
- **Confusion Matrix** See which species get confused
- **Adjusted Rand Index (ARI)** Similarity between two partitions

Careful... cluster labels (1,2,3) don't necessarily match species order

We need to find the best matching between predicted labels and true species

Confusion Matrix and Accuracy

```r
# confusion matrices
clusComplete <- cutree(hcComplete, 3)
clusWard <- cutree(hcWard, 3)
confMatComp <- table(trueLabels,clusComplete)
confMatWard <- table(trueLabels, clusWard)
print(confMatComp)
print(confMatWard)

# calculate accuracy (with best matching)
library(clue)  # for solve_LSAP
calcAccuracy <- function(true, pred) {
  tab <- table(true, pred)
  matches <- solve_LSAP(tab, maximum = TRUE)
  sum(diag(tab[, matches])) / length(true)
}

accComplete <- calcAccuracy(trueLabels, clusComplete)
accWard <- calcAccuracy(trueLabels, clusWard)
```

```
            1  2  3
setosa     50  0  0
versicolor  0 23 27
virginica   0 49  1

            1  2  3
setosa     50  0  0
versicolor  0 49  1
virginica   0 15 35
```

Adjusted Rand Index (ARI)

Measuring Agreement Between Partitions

Rand Index $RI = \frac{a+d}{a+b+c+d}$. For all pairs of points

- a = both in same cluster in both partitions
- d = in different clusters in both partitions
- b, c = disagreements

Adjusted Rand Index Corrected for chance

$$ARI = \frac{RI - E[RI]}{1 - E[RI]}$$

Interpretation

- ARI = 1: Perfect agreement
- ARI = 0: Random clustering
- ARI < 0: Worse than random

Calculating ARI

```r
# Calculate ARI
library(mclust)
ariComp <- adjustedRandIndex(trueLabels, clusComplete)
ariWard <- adjustedRandIndex(trueLabels, clusWard)

# Also calculate for k=2 (common result)
clusComplete_k2 <- cutree(hcComplete, 2)
clusWard_k2 <- cutree(hcWard, 2)

ariComp_k2 <- adjustedRandIndex(trueLabels, clusComplete_k2)
ariWard_k2 <- adjustedRandIndex(trueLabels, clusWard_k2)
```

With k=3 (matches true number):

- Complete Linkage: 0.642
- Ward's Method: 0.731

With k=2 (silhouette best):

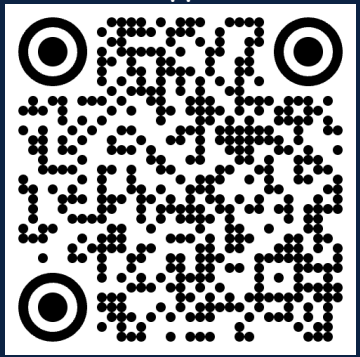
- Complete Linkage: 0.422
- Ward's Method: 0.568

Key Takeaways

- **Linkage matters:** Different methods give different results
- **Ward and Complete** are usually most reliable choices
- **Single linkage** often creates chains - use with caution
- **Elbow method:** Look for diminishing returns in WSS
- **Silhouette:** Measures cluster separation and cohesion
- **Domain knowledge** can override statistical metrics
- **Scaling decision** depends on variable units and meaning
- **ARI** measures agreement when true labels known
- **Clustering accuracy** requires best label matching
- **Confusion matrices** show which groups get mixed up

What's Next?

Additional Questions?
Book an Appointment!



Next Workshop:

Centroid Based Models
–> January 19, 3:30 PM

Thank You!

Questions?

Workshop Materials:
`https://csc-ubc-okanagan.github.io/workshops/`

Contact:
jesse.ghashti@ubc.ca