CS 660: Mathematical Foundations for Analytics

Dr. Francis Parisi

Pace University

Spring 2017

Course Overview

1. Part I: Data Science Fundamentals
   - Data Science Concepts and Process
   - The R Language
   - Exploratory Data Analysis
   - Cleaning & Manipulating Data
   - Presenting Results
2. Part II: Graphs & Statistical Methods
   - Basic Graphics
   - Advanced Graphics
   - Probability & Statistical Methods
3. Part III: Modeling Methods
   - Model Selection and Evaluation
   - Linear and Logistic Regression
   - Unsupervised Methods
   - Advanced Modeling Methods

Unsupervised Methods

- Unlike supervised learning, in *unsupervised* learning the data do not contain a “right” classification
- This makes unsupervised learning more challenging
- Often unsupervised learning is used as part of exploratory data analysis
- And validating results is difficult because we do not know the *true* answer

Unsupervised Methods

- Unsupervised learning techniques are growing in importance, especially in certain fields of study
- Cancer researchers may look for subgroups among breast cancer samples to get a better understanding of the disease
- Online shopping sites may try to identify groups of shoppers with similar histories and show items in which the shopper is likely to have an interest
- Or one may study how the population moves around based on sociioeconomic factors

Cluster Analysis

- Cluster analysis is a data-reduction technique used to uncover subgroups within a dataset
- Roughly speaking, *cluster* is a group of observations that are more similar to each other than they are to the observations in the other groups
- The two most common clustering approaches are hierarchical clustering and partitioning clustering

Cluster Analysis

- In hierarchical clustering each observation is its own cluster to start then we combine observations into larger clusters until we reach a single cluster
- In partitioning approaches, you specify the number of clusters k, then the observations are divided into the k groups
- There are many algorithms within each of these approaches, and we will explore a couple

Cluster Analysis

- We will look at some data sets within the `flexcust` and `rattle` packages
- We will make use of functions from the following packages: `cluster`, `NbClust`, `flexclust`, `fMultivar`, `ggplot2`, and `rattle`

Steps in Cluster Analysis

1. Select the variables that you believe are important in understanding differences among the groups
2. Scale the data – it is best to "standardize" your data so that variables with larger ranges do not dominate your results
3. Check for outliers – either eliminate them if possible or use an approach that is robust like partitioning around medoids
4. Calculate the distances
5. Select a clustering algorithm
6. Obtain one or more solutions
7. Determine the number of clusters – try several different numbers and find the "best" one

Steps in Cluster Analysis....cont'd

8. Find the final clustering solution based on 6 and 7 above
9. Visualize your results – this will help you understand the solution and its usefulness
10. Interpret the clusters – e.g., what do the observations in each cluster have in common?
11. Validate the results – Do these groupings make sense, are they real?

Cluster Analysis

- A critical step in every cluster analysis is the calculation of the distance, or dissimilarity between each observation, or the complement, proximity
- The most common measure of distance is Euclidean distance
- The Euclidean distance between two observations is given by

$$d_{ij} = \sqrt{\sum_{k=i}^{p} (x_{ik} - x_{jk})^2}$$

where i and j are observations and p is the number of variables

Cluster Analysis

The `nutrients` dataset in the `flexclust` package includes measurements on 27 types of foods

```
> data(nutrient, package="flexclust")
> head(nutrient, 4)
                energy protein fat calcium iron
BEEF BRAISED       340      20  28       9  2.6
HAMBURGER          245      21  17       9  2.7
BEEF ROAST         420      15  39       7  2.0
BEEF STEAK         375      19  32       9  2.6
```

The Euclidean distance between beef braised and hamburger is

$$d = \sqrt{(340-245)^2 + (20-21)^2 + (28-17)^2 + (9-9)^2 + (26-27)^2}$$

or $d = 95.64$

Cluster Analysis

Or we can use the `dist()` function in R...

```
> d <- dist(nutrient, method = "euclidean") # default
> as.matrix(d)[1:4,1:4]
              BEEF BRAISED HAMBURGER BEEF ROAST BEEF STEAK
BEEF BRAISED           0.0      95.6        80.9        35.2
HAMBURGER             95.6       0.0       176.5       130.9
BEEF ROAST            80.9     176.5         0.0        45.8
BEEF STEAK            35.2     130.9        45.8         0.0
```

which gives us the same result of course

Cluster Analysis

Other distance and similarity measures include

- Hamming distance: for categorical variables, counts the number of mismatches
- Manhattan (or city block) distance

$$d_{ij} = \sum_{k=i}^{p} |x_{ik} - x_{jk}|$$

- Cosine similarity commonly used in text analysis

$$\text{similarity} = \cos(\theta) = \frac{\sum_{k=1}^{n} x_{ik} x_{jk}}{\sqrt{\sum_{k=1}^{n} x_{ik}^2 x_{jk}^2}}$$

and

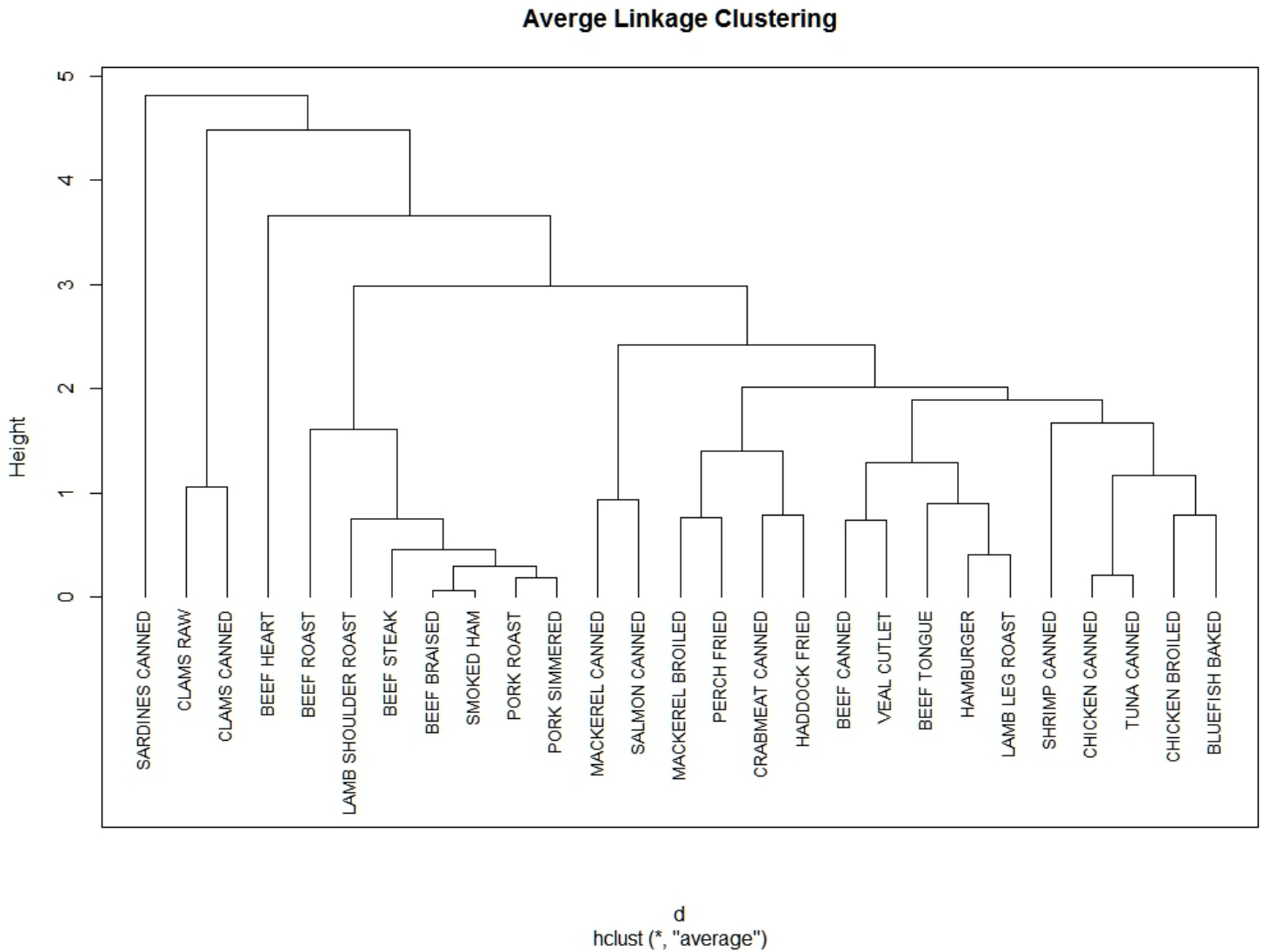
$$\text{distance} = cos^{-1}(\text{similarity})/\pi$$

Cluster Analysis

Hierarchical Cluster Analysis

1. Define each observation (row, case) as a cluster
2. Calculate the distances between every cluster and every other cluster
3. Combine the two clusters that have the smallest distance - this reduces the number of clusters by one Repeat steps 2 and 3 until all clusters are merged into into a single cluster

Let's look at R



Cluster Analysis

- The previous result helps us understand the similarities among food groups based on their nutrients
- If we want to classify these food into a number of smaller groups we need to do more analysis
- The `NbClust` package will help us out –back to R

Partitioning Cluster Analysis

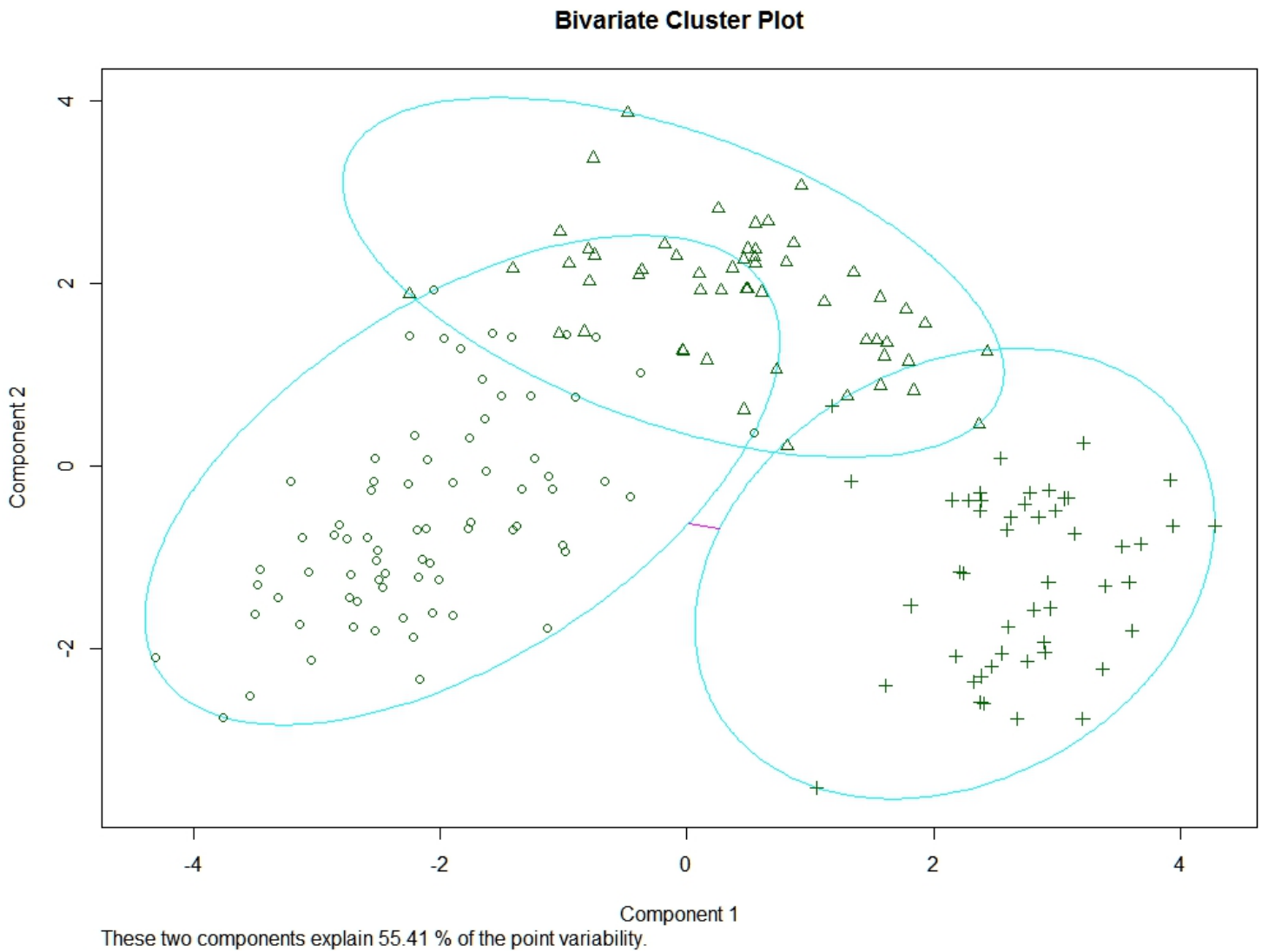
- Partitioning divides the observations into k groups
- The groups are shuffled to make up the most cohesive clusters possible according to the criterion
- Two popular approaches are k-means and partitioning around medoids (PAM)

k-means Clustering

1. Select k centroids (rows) chosen at random
2. Assign each data point to its closest centroid
3. Recalculate all the centroids by averaging all the data points in each cluster
4. Assign the data points to clusters based on the new centroids
5. Continue 3 and 4 until the observations don't reassign anymore
6. Let's go to R for another example

Partitioning Around Medoids

1. Randomly select K observations (call each a medoid)
2. Calculate the distance/dissimilarity of every observation to each medoid
3. Assign each observation to its closest medoid
4. Calculate the sum of the distances of each observation from its medoid (total cost)
5. Select a point that isnt a medoid, and swap it with its medoid
6. Reassign every point to its closest medoid
7. Calculate the total cost
8. If this total cost is smaller, keep the new point as a medoid
9. Repeat steps 5-8 until the medoids dont change
10. Let's look at an example in R


Bivariate Cluster Plot
Component 2
Component 1
4
2
0
-2
-4
-2
0
2
4
These two components explain 55.41 % of the point variability.

Cluster Analysis – Summary

- There are many methods for cluster analysis
- Cluster analysis helps us discover subgroups within our data
- Common methods we covered in R are hierarchical cluster analysis, and partitioning methods that include k-means and partitioning around medoids

References

James, G., Hastie, T., Witten, D. and Tibshirani, R. (2013).
An Introduction to Statistical Learning with Applications in R.
Springer, second edition.
6th Printing 2015.

Kabacoff, R. I. (2015).
R in Action.
Manning, Shelter Island, NY, second edition.

Lander, J. P. (2014).
R for Everyone.
Addison-Wesley, Upper Saddle River.

Zumel, N. and Mount, J. (2014).
Practical Data Science with R.
Manning, Shelter Island, NY, second edition.