# Identify Leaf clusters

Addison James, Bin Zhuo, Feifei Lei, Nandhita Narendra Babu

June 4th, 2014

# Contents

# Overview

- To identify the leaf clusters from leaf data set
- Leaf dataset
  - Collection of shape and texture features extracted from digital images of leaf specimens
  - A total of 40 different plant species
  - Only 30 of those were present in the data provided
  - 340 observations

# Methods

"All methods are concerned with using the inherent structures in the data to best organize the data into groups of maximum commonality"
- Jasonb in "A Tour of Machine Learning Algorithms"

- **K-means Clustering**
  To make 30 clusters to see how well this approach recovers the real data
- **Model-based clustering**
  To group the leaves at higher level

# K-Means Clustering

# Method Description

- Definition $\rightarrow$ Assigning $n$ observations into $k$ clusters
- Requires to specify the number of clusters $k$ to extract
- Similarity criterion $\rightarrow$ proximity to the mean of each cluster
- Assignment method $\rightarrow$ minimize euclidean distance from the data to the means of the clusters

# Algorithm Description

1. Set Initial Cluster Means
2. Assign each datum to the cluster with the nearest mean
3. Calculate the new mean of each cluster
4. Repeat steps 2-3 until convergence

# Procedure

- function *kmeans* : To cluster leaves into species
- $k = 30$ clusters $\rightarrow$ cluster all the observations into 30 separate species based on the 14 covariates available
- Covariates : eccentricity, aspect ratio, elongation, solidity, stochastic convexity, isoperimetric factor, maximal indentation depth, lobedness, average intensity, average contrast, smoothness, third moment, uniformity, and entropy

# K-means Clustering result

Two possible solutions

- First solution
  - examine each cluster and assigns a species number to each cluster based on the most frequent species in the cluster
  - can fail to create a one-to-one relationship between the species and cluster $\rightarrow$ there may be a tie or two different clusters may be assigned to the same species
- Second solution
  - examine each species and assign the species to the clusters, but reversing the role of clusters and species

# K-means Clustering result

- Result from First Method
  - 138 of the 340 observations (41%) ended up in the correct cluster
  - successfully assigned 19 of the 30 clusters (63%) to a species
- Result from Second Method
  - less optimistic evaluation
  - 126 observations were correctly clustered (37%)
  - 17 of the 30 species were assigned a cluster (57%)
  - poor job of grouping leaves of the same species based on the known covariates

# Model-based Clustering

# Method description

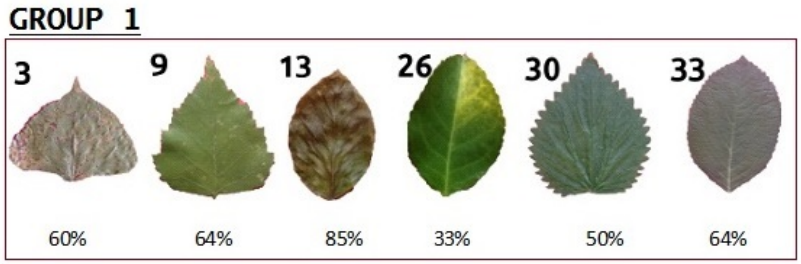
- Basic idea: Clustering as probability estimation
- One model for each cluster
- Generative model:
  - Probability of selecting a cluster
  - Probability of generating an object in cluster
- Use EM algorithm

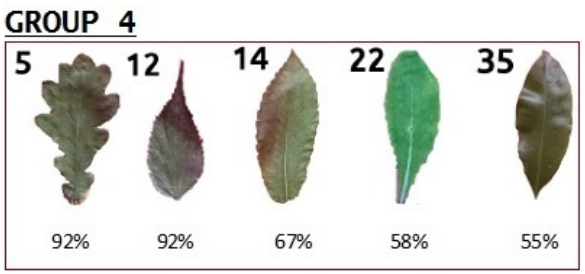
# Procedure

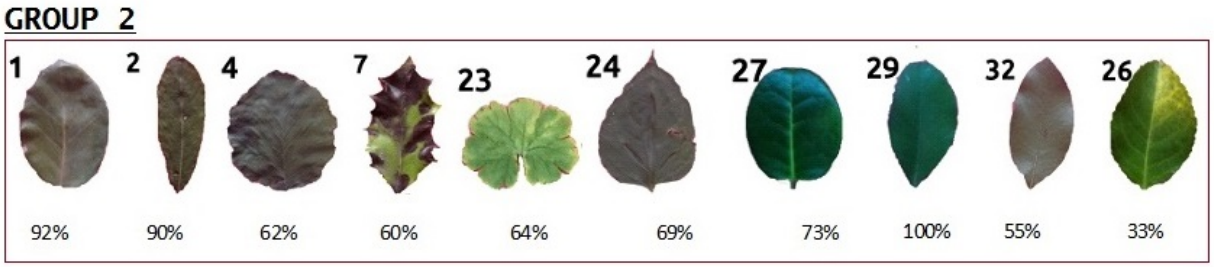
- Covariance parameterization and number of clusters are selected via BIC
- Clustering using normal mixture modeling via EM algorithm
- *Mclust* function
  - Selects the optimal model according to BIC for EM initialized by Guassian Mixture Models
  - Chooses the model and number of clusters with the largest BIC

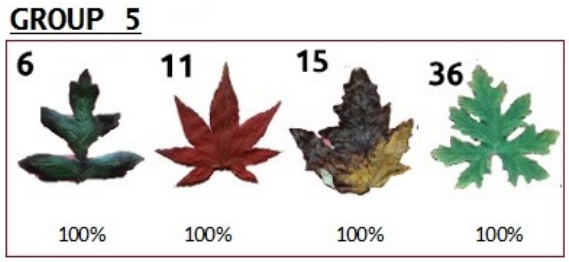
# Model-based clustering results

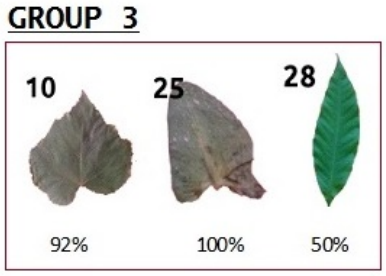
Optimal cluster number $\rightarrow$ 6

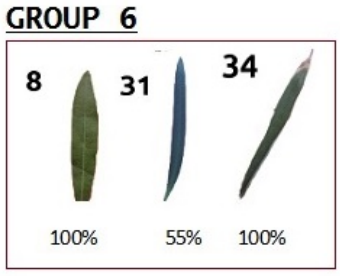


# Assumptions/Limitations and Scalability

# Assumption/Limitations

- Assumptions
  - 14 variables are normally distributed
  - Each cluster follows a multivariate normal distribution
  - Can model all clusters simultaneously as a mixed multivariate normal
- Limitations
  - K-means Clustering
    - comparable size of clusters
  - Model-based clustering
    - provides only 10 of the 14 possible variance-covariance structures
    - Cannot handle 'NA' problem

# Scalability

- k-Means is faster than hierarchical clustering
- function *mclust* is able to cope with large datasets
- Mclust can use sampled data in the hierarchical phase before applying EM to extend the method to larger datasets

Questions ?

# EM Algorithm Description

## EM Algorithm

- **Initialization**: Choose means at random
- **E step**:
  - For all points and means, compute P(point|mean)
  - P(mean|point) = P(mean) P(point|mean) / P(point)
- **M step**:
  - Each mean = Weighted avg. of points
  - Weight = Prob(mean|point)
- Repeat until convergence
- Guaranteed to converge to local optimum