An Overview of Clustering:
Finding and Extracting Group Structure
in (High-Dimensional) Data

Goal: build foundation for understanding a variety of clustering methods; be able to identify the types of problems and which literature might be helpful; learn which questions to ask

Timeline: (subject to change depending on audience needs)

- 8:00-8:30am: Intro, Motivation; Goals; Data
- 8:30-9:15am: Distance-based methods (Linkage Clustering, K-means, K-medoids)
- 9:15-10:00am: Density-based clustering (model-based clustering)
- 10:00-10:15am: Break (for all short courses)
- 10:15-10:30am: Density-based clustering (nonparametric clustering)
- 10:30-11:00am: Spectral Clustering, Variable Selection
- 11:00am-11:30am: Visualization, Diagnostics
- 11:30am-12:00pm: Text (Document) Clustering

We will also take brief breaks as needed during the blocks of material.

Short Course Website: `http://www.stat.cmu.edu/~rnugent/CSP2015`

Contact Info:

Rebecca Nugent, Samuel L. Ventura
Dept of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213
rnugent@stat.cmu.edu, sventura@stat.cmu.edu
`http://www.stat.cmu.edu/~rnugent`, ~sventura

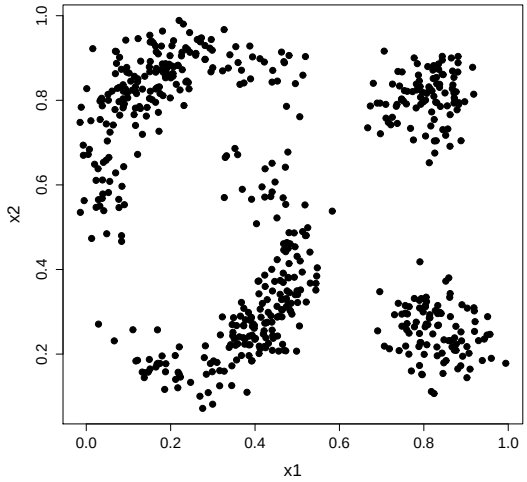
Clustering, in General:

- example of "Unsupervised Learning" - learning without labels
- Given vectors $\mathbf{X} = (X_1, X_2, ..., X_p)$, goal is to "understand" or describe the joint distribution $p(\mathbf{X})$ of these vectors
- Organize, Summarize, Categorize, Explain
- Infer properties of $p(\mathbf{X})$ without any labels
- Dimension is often higher than supervised learning problems
- Could be interested in identifying lower dimension manifold;
  are there a few latent variables/traits that summarize the higher dimensional information?
- Are the variables associated with each other? How?
- Could just want to know how many groups are in the data
- Locate the regions of high density
  (both in continuous and categorical data)
- Can compare *agreement* of different results;
  Need labels to return misclassification rate
- No one measure of success, can be dependent on application
- Trying to characterize the "structure" in the data
- Might define "success" as method that "best captures" the structure

Clustering Applications:

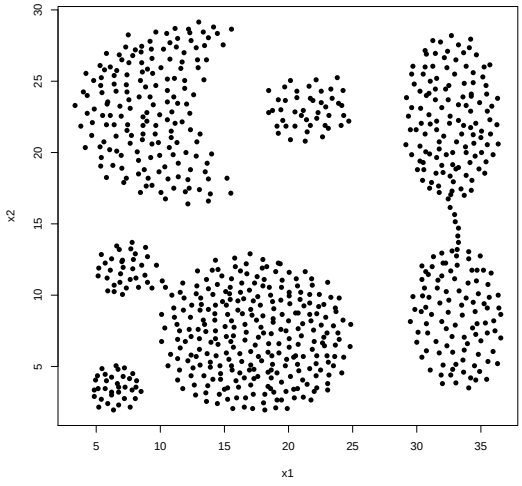
Datasets:

- Four groups, two dimensions; well-separated



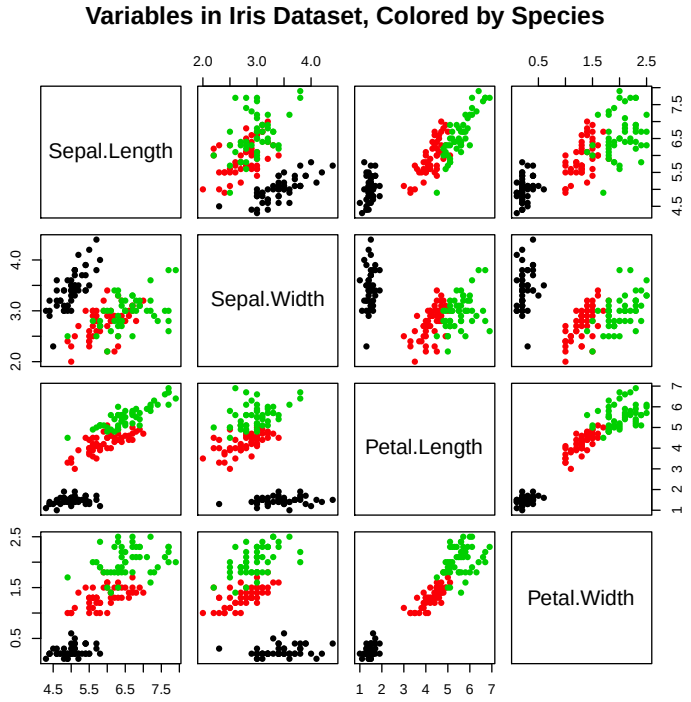

```
four.groups<-read.table("fourgroups.dat"); dim(four.groups)
plot(four.groups,xlab="x1",ylab="x2",pch=16)
```

- Seven groups, two dimensions; varying separation and shapes




```
aggregation<-read.table("aggregation.txt")
aggr.data<-aggregation[,1:2]
aggr.labels<-aggregation[,3]
plot(aggr.data,xlab="x1",ylab="x2",pch=16)
```

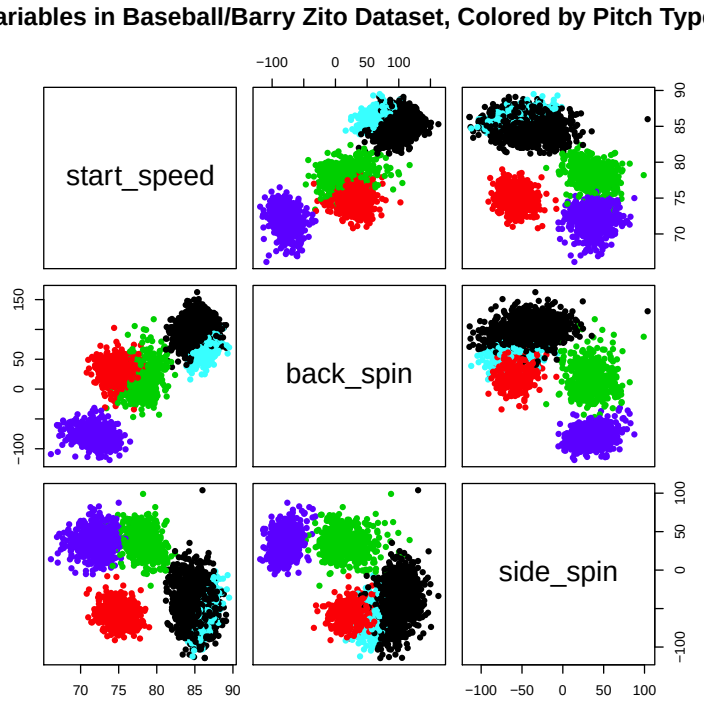
- Iris Species *(Fisher, 1936)*:
  - Groups: three different Iris species, 50 flowers in each
  - One well-separated group, two overlapping groups
  - Covariates: lengths and widths of sepals and petals
  - For more information, see
    `https://archive.ics.uci.edu/ml/datasets/Iris`
  - Also found in MASS library in R (`iris`)




```
#  Load Iris dataset into R
iris <- read.csv("iris.csv")[,-1]
##or library(MASS); help(iris)

#  Plot variables in Iris dataset, colored by species
pairs(iris[,1:4], col = iris$Species, pch = 16,
main = "Variables in Iris Dataset, Colored by Species")
```

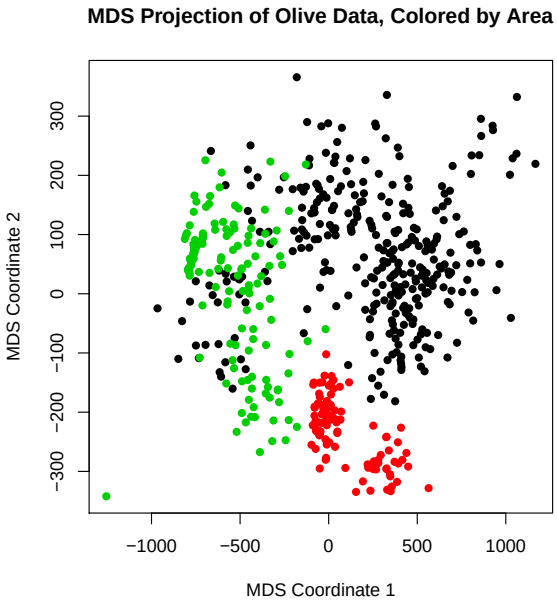
- Baseball Dataset *(Pane et al, 2013)*:
  - Pitcher: Barry Zito, SF Giants, 2010–2011
  - Groups: 5–7 different pitch types
  - Some well-separated pitch types (e.g. fastball and curveball)
  - Some overlapping pitch types (e.g. curveball and slider)
  - Covariates: speed, horizontal spin/break, vertical spin/break
  - For more information, see `http://arxiv.org/abs/1304.1756`

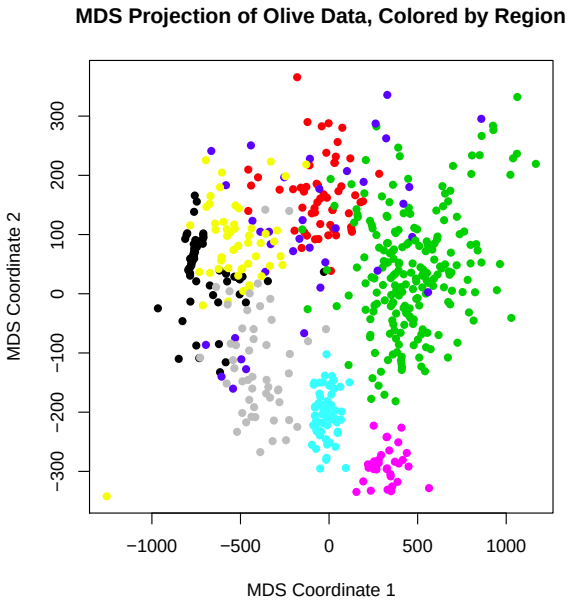



```
#  Load Baseball / Barry Zito pitch dataset into R
zito <- read.csv("zito.csv", header = T)[,-1]

#  Plot variables in Baseball dataset, colored by pitch type
pairs(zito[,1:3], col = zito$pitch_type, pch = 16,
main = "Variables in Baseball/Barry Zito Dataset, Colored by Pitch Type")
```

- Olive Oil *(Forina, et al, 1983)*
  - Chemical compositions of 572 olive oils from Italy
  - Groups: Two sets of geographic labels
    1) three regions and 2) nine specific areas (contained in regions)
  - Covariates: eight chemical measurement variables
  - For more information, see
    `http://artax.karlin.mff.cuni.cz/r-help/library/pdfCluster/html/oliveoil.`



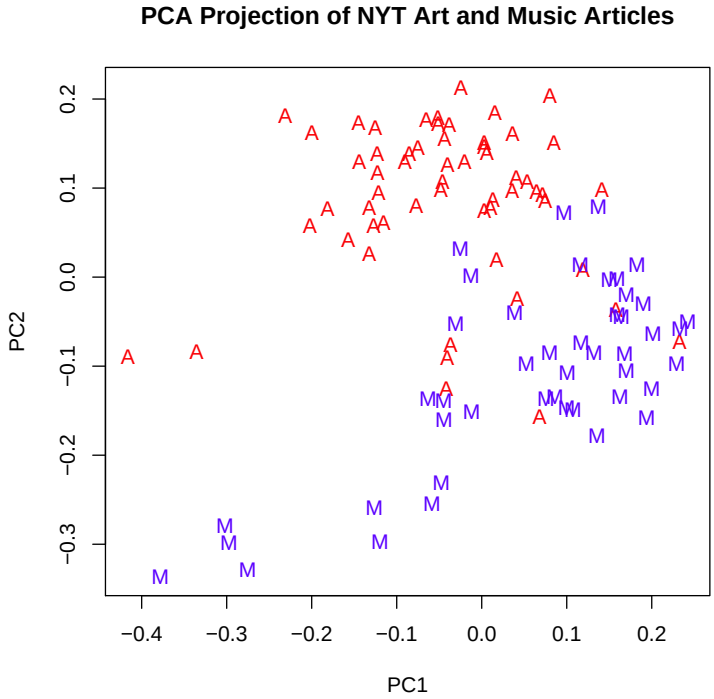



```
#  Load Olive dataset into R
olive <- read.csv("olive.csv", header = T)[,-1]

#  Calculate Multidimensional Scaling Projection into a lower (2-D) dimension
olive.mds <- cmdscale(dist(olive[,2:10]), 2)

#  Plot the two MDS coordinates
par(mfrow = c(1,2))
plot(olive.mds, col = olive$area, xlab = "MDS Coordinate 1", ylab = "MDS
Coordinate 2", main = "MDS Projection of Olive Data, Colored by Area", pch = 16)
plot(olive.mds, col = olive$region, xlab = "MDS Coordinate 1", ylab = "MDS
Coordinate 2", main = "MDS Projection of Olive Data, Colored by Region", pch = 16)
```

- New York Times Corpus of Art and Music Articles *(Shalizi, 2014)*
  - Groups: 100 NYT Art and Music articles
  - Covariates: TF-IDF values for 4432 unique words across articles




```
#  Load New York Times article corpus into R
nyt <- read.csv("nyt.csv", header = T)[,-1]

#  Check dimensionality
dim(nyt) # 102 rows (documents/articles), 4432 columns (unique words)

#  Need to reduce dimensionality; use Principal Components Analysis (PCA)
nyt.pca <- prcomp(nyt[,-1])

#  Plot first two components, color by article type
#  Arts stories with red As
#  Music stories with blue Ms
plot(nyt.pca$x[,1:2],type="n", main = "PCA Projection of NYT Art and Music Articles
points(nyt.pca$x[nyt[,"class.labels"]=="art",1:2],pch="A",col="red")
points(nyt.pca$x[nyt[,"class.labels"]=="music",1:2],pch="M",col="blue")
```

Looking for Group Structure in Data: **Clustering**

Goal: partition observations such that those in the same cluster are "more similar" to each other than they are to those in other clusters

Characterizing a Group/Cluster: want to summarize the structure

- Center:
- Spread:
- Shape:

Also need an *assignment list*; which observations belong to the cluster?

Distances/Dissimilarities

To understand/measure structure in a group of variables or feature vectors, need an idea of how observations relate/compare to each other.

Notation:

Measuring Distance: Common to describe the relationship between two observations by their "distance" or "dissimilarity": $d(i, j)$

Properties of a Distance:

Often expect $d(i, j)$ to increase as obs become more different/dissimilar. We can store this information in a *distance/dissimilarity* matrix.

Euclidean Distance: commonly used distance; "as the crow flies"

Can sometimes visualize the structure in the distance matrix.

Heat Map: multicolored representation of a matrix of values; color spectrum represents the range of values (e.g. red = low; yellow = high)

Why is the structure evident? What happens in practice?

What if obs are not ordered by group? What if there are outliers?

Potential issues with distances

- distance can change if measurement units change
- variables can have different scales and/or variances

Other distances: Manhattan (city block distance); L-infinity or Maximum distance; Hamming distance among others

Hierarchical Linkage Clustering

Hierarchical Partitioning: Agglomerative vs Divisive

(Agglomerative) Hierarchical Linkage Clustering: algorithm linking observations/groups in order of closeness in a hierarchical structure; generates n possible partitions for $K = 1, 2, ..., n$ clusters

This hierarchical structure is stored in a *dendrogram*.

We determine the clusters/partition by cutting the dendrogram.
Can be difficult to choose the partition when structure not obvious.

Single Linkage: intergroup distance: smallest possible distance

Characterized by "chaining", nearest neighbor effect, good at picking out curvilinear/non-spherical groups

Complete Linkage: intergroup distance: largest possible distance

Characterized by splitting the data up into more compact subsets

Other types of Linkage:

- Average:

- Centroid:

- Ward's

- Minimax Linkage (based on prototypes; less well known)

Can use any type of distance/dissimilarity;
in R, need to pass in a `dist` structure or a full distance matrix.

What kind of dissimilarities might we use?

To group similar obs, some methods try to balance *minimizing* within-cluster distance and *maximizing* between-cluster distance.

Within-Cluster Distance:

Between-Cluster Distance:

K-means: algorithm to partition obs into K **spherical clusters**

Measure "quality" of clusters with *within-cluster squared-error criterion*

Required: Set the number of clusters, K, in advance.

Given a set of K initial cluster centers, alternate between:

- Assign each observation to the closest center
- Recompute the centers given the current assignments

Stop when the cluster assignments/centers no longer change.

Each step decreases the within-cluster criterion:

- Given the cluster centers:

- Given the current assignments:

In practice:

- First few steps correspond to large drops in the criterion; later steps correspond to negligible drops.
- Use K randomly chosen observations as the starting centers (but don't have to; can choose specific centers)
- Have an idea of what K should be in advance

What if we don't know K? How do we choose?
If we increase K, what happens to the within-cluster criterion?

We use an *elbow graph* to determine a "useful" K.

What do we look for in the elbow graph?

K-means is also dependent on the set of starting centers you choose; solutions can vary widely. How do we pick?

K-Means can be strongly influenced by outliers (since based on means).

K-Medoids: Partitioning around Medoids

Medoid: the observation in the data set (cluster) whose average distance to all the other observations is minimal; not as susceptible to outliers

Given a starting set of K observations (medoids), alternate between:

- Assign each observation to the closest medoid.
- For each cluster, find the observation that corresponds to the lowest criterion value for the cluster; reassign as medoid

until cluster assignments no longer change.

Much more computationally difficult; at each step, criterion has to be optimized over all obs *(which one is the new medoid?)*

So far, we have looked at distance-based approaches.
In contrast, we can adopt a *statistical approach:*

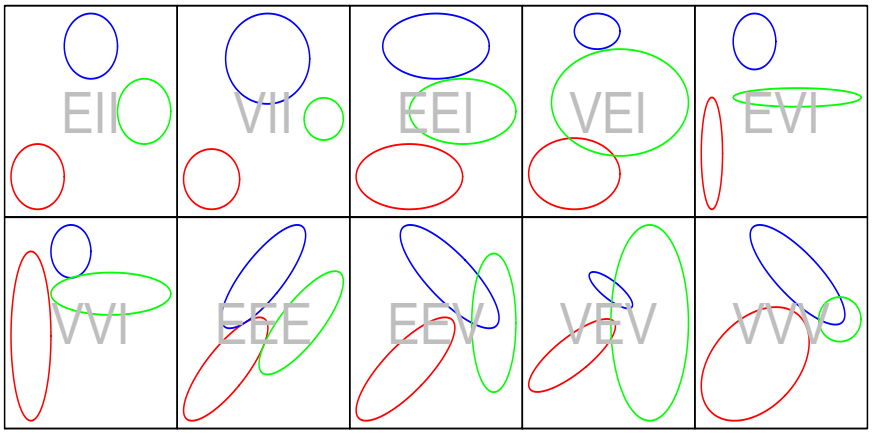
There are two subfields:

- Parametric

- Nonparametric

Model-Based (Parametric) Clustering: assumes that each population subgroup has its own density; overall pop is weighted combination

What type of densities do we fit?



Choosing the "Best" Model:

Pick the model that maximizes the Bayesian Information Criterion.

Looking at a Two Group Mixture:

To fit the model, we need to estimate three sets of parameters:

In particular, the covariance matrix can be parameterized to dictate the shapes, orientations, etc of the group densities.

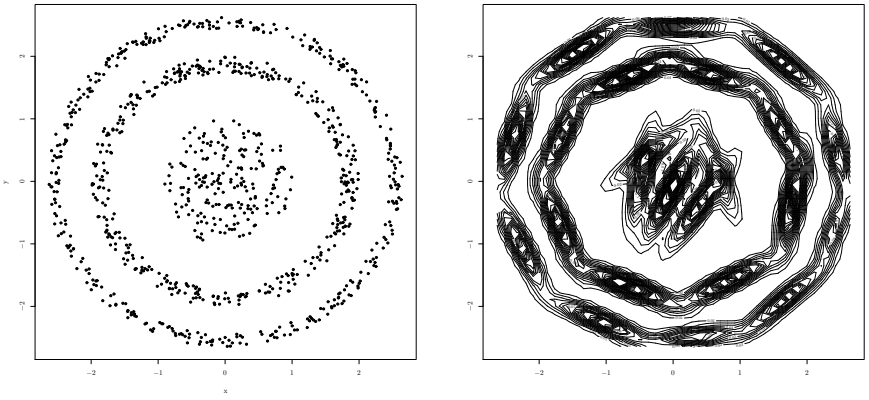
The models are fit using the Expectation-Maximization Algorithm:

After the final model is chosen (by the BIC), the procedure returns:

- the name of the model
- the estimated means and covariance
- the estimated membership probabilities
- the cluster assignments

Common assumption: each component represents a population group

If groups are not Gaussian, may overfit the number of components.



Need to think about how you decide to merge components. Options?

What about Gaussian clusters with noise?

Two options:

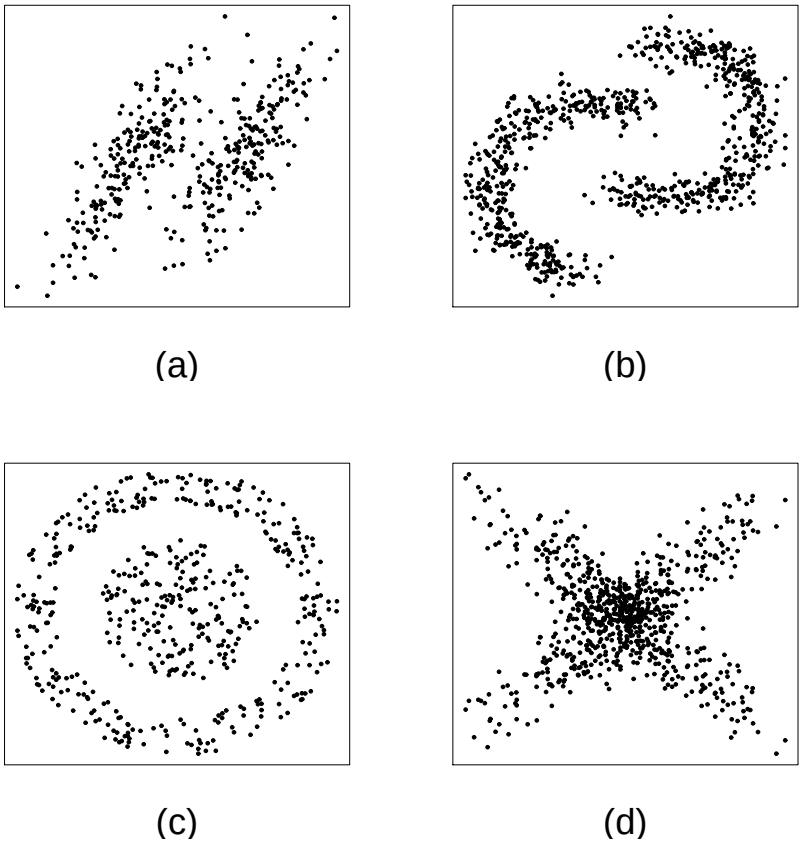
In general, need to be careful about how you interpret the components (whether or not they represent true groups in the population).

Nonparametric Clustering:

Often we just associate groups with high frequency areas.

Groups in the population correspond to modes of the density $p(x)$.

Gives the following definition: contiguous, densely populated areas of feature space, sep- arated by contiguous, relatively empty regions *(Carmichael, George, Julius).*



NP Goal: find the modes of a density $p(x)$ (or $\hat{p}(x)$); assign observations to the "domain of attraction" of a mode *(contrast with MBC)*

Finding Modes: associate presence of groups/modes with excess mass in one area surrounded by low mass areas.

Level Sets of a Density:

NP Approach: find the modes of a density $p(x)$ (or $\hat{p}(x)$); assign observations to "domain of attraction" of a mode; build cluster tree; *(NPEx.pdf)*

Unlike other clustering procedures, NP clustering is very dependent on the density estimate $\hat{p}(x)$.

Each mode of the density estimate $\iff$ cluster/population group

Kernel Density Estimate: common nonparametric density estimate

Choice of kernel:

- Gaussian
- Epanechikov
- Biweight/Triweight
- Triangular
- Box

Choosing a Bandwidth: Often trying to minimize an error measure; there are several reference rules (Scott or Silverman); could also use cross-validation; open research problem, no "one size fits all" choice

Spectral Clustering

Flexible method that analyzes the eigenvalues/vectors of an affinity/transition matrix to find clusters of arbitrary shape/size *(Ng, Jordan, Weiss)*

Need to measure the "connectivity" of the data;
how would we walk through the data via the "shortest" path?

How can we measure similarity?

Given a similarity matrix S, we build a *affinity* or *transition* matrix.

What would happen if we changed σ?

We use this transition matrix to find the structure in the data *(which groups of points are highly connected?)* via its eigenvalue decomposition (essentially project observations to find the structure).

The general algorithm is:

- Compute P
- Compute the eigenvectors $v_1, v_2, ..., v_n$ of P
- Select the first K eigenvectors
- Cluster the "new obs" using K-means

There are theoretical reasons for using k-means; it has been shown to minimize a spectral clustering criterion called the "gap" - the difference between the solution you choose and the best possible solution (Meila/Shi 2001).

Variable Selection

Often working with lots of variables; can be hard to find signal or see clusters in the presence of noisy or variables with no clustering info

Common solution for "too many variables" is *dimension reduction* (e.g. principal components, (non-metric) multi-dimensional scaling)

However, those techniques still include information from all variables. Important to consider approaches that reduce dimensionality by *selecting* the "right" variables for clustering

Examples:

- Spectral Clustering:

- Model-Based Clustering (Raftery, Dean):

- K-Means (Steinley):

Assessing/Comparing the Clusterings

Could use *percent correct* to characterize our results (if had labels).

Advantages/disadvantages:

What if the clustering algorithm is not completely deterministic?

Several clustering comparison criteria we could use (*also applies to comparing a set of clusters to the truth*); most are based on counting the pairs of observations on which two clusterings agree/disagree.

Fowlkes-Mallow Index: geometric mean of the probability that a pair of points in the same C_k are also in the same cluster in $C'_{k'}$

Rand Index:

Adjusted Rand Index (ARI, Hubert and Arabie): noticed that RI does not range over entire [0,1] interval. (min(RI)$>$ 0; RI tends toward 1)

Instead we adjust the index to have an expected value of zero under random partitioning (independent clusterings) with a max value = 1. Tends to give you credit for splitting a group into two clusters

Another way of thinking about percent correct is *misclassification error*:

(*Information-theoretic* point of view: entropy, mutual information, VI)

Using the Criteria:

- You can never compare values from different criteria;
  they measure different things
- We can compare the performance of two different clustering algorithms by comparing each against truth. Pick better (?) one.
- Compare the stability of a non-deterministic procedure by repeating several times and watching how the criteria change.

Visualization Diagnostics

Reminder of Model-Based Clustering:

After choosing our final model, each observation is assigned to the cluster that corresponds to the highest membership probability (`z`).

Maximum Membership Probability:

Uncertainty Index:

What would the uncertainty vector look like for a "good" set of clusters? What about a "bad" set of clusters?

When looking at other types of methods, we need some kind of "uncertainty measure". What would it mean to be "well-assigned" ?

We want to quantify the "closeness" of an observation to any cluster:

Silhouette Measure:

Given assignments, we find the silhouette value s_i for each observation (vector of length n); characterize cluster by its silhouette values

Longitudinal/Trajectory Clustering

Been looking at structure for observations with one set of measurements.

Sometimes observations may have sets of repeated measurements.

Can be characterized by a path or a *trajectory*. We're often interested in determining the "center trajectory" for a group of observations.

Can estimate the number of trajectories, the coordinates of the "center" trajectories, and the probability of belonging to each trajectory.

Text/Document Clustering

Can look for structure in emails, blog posts, letters, articles, etc by analyzing the words they use and how often they use them.

Bag of Words Model:

Difficult to cluster this type of data (binary, counts, high-dim); often convert to *Term-Frequency Inverse Document Frequency (TF-IDF).*

TF-IDF values can be affected by length of document; instead of using Euclidean distance to look at the similarity between documents, project to sphere and use cosine distance (Spherical K-means)

Have same issues with dimension reduction/variable selection. Often choose "important" words based on TF-IDF sums across documents.

Notes:

Notes:

Review/Takeaways