An Overview of Clustering:
Finding Group Structure in Educational Research 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)

- 9:00-9:15am: Intro, Motivation; Goals
- 9:15-10:00am: Distance-based methods (Linkage Clustering, K-means, K-medoids)
- 10:00-10:40am: Density-based clustering (model-based clustering)
- 10:45-11:00am: Break (for all tutorials/workshops)
- 11:00-11:30am: Density-based clustering (nonparametric clustering)
- 11:30am-12:00pm: Visualization, Diagnostics
- 12:00pm-12:30pm: Longitudinal Clustering/Text (Document) Clustering

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

Contact Info:

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

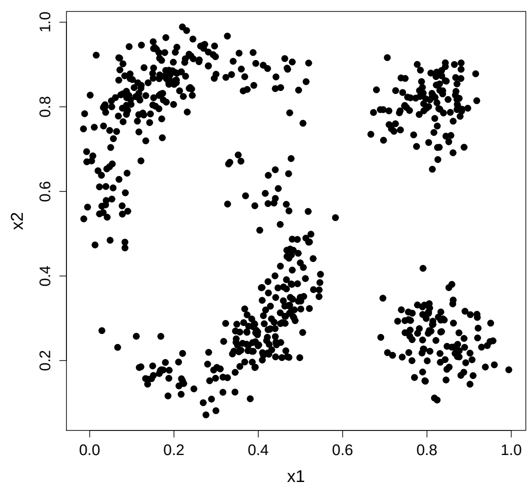
Clustering, in General:

- 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 in Education:

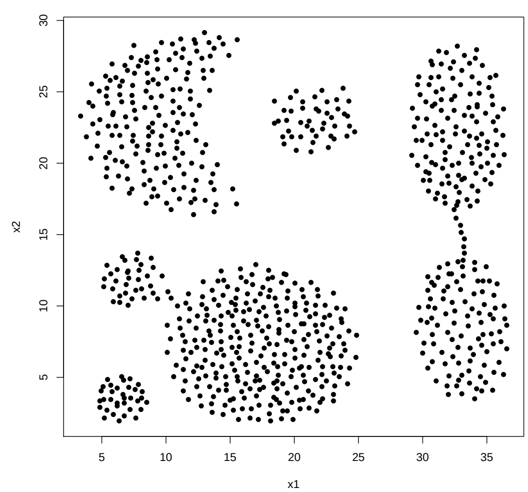
Datasets:

- 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



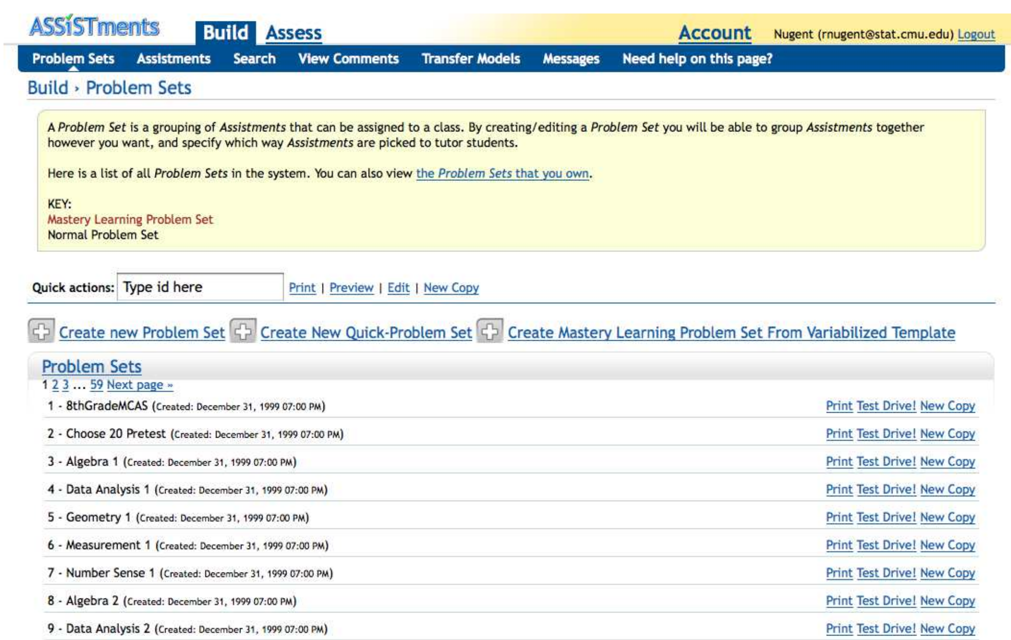

```
aggregation<-read.table("aggregation.txt")
aggr.data<-aggregation[,1:2]
aggr.labels<-aggregation[,3]
plot(aggr.data,xlab="x1",ylab="x2",pch=16)
```

The Assistments Project: `http://www.assistments.org`

- Web-based tutoring program developed by Carnegie Mellon University, Carnegie Learning, and Worcester Polytechnic Institute
- Blends tutoring "assistance" with "assessment" reporting
- Over 4000 students in Massachusetts and Pennsylvania utilized the system in 2007-2008
- System currently tracks/reports on about 120 skills per grade level

Goals:

- Help prepare students for end-of-year exams, e.g. MCAS
- Help teachers identify weaknesses/strengths in their students and in their curriculum
- Allow teachers to use their time more effectively
- Help researchers discover how students learn



Teachers can *build* questions or select from problem test banks. Students are assigned a set of questions online for practice.

Each question coded as a *main*, broken up into *scaffolds*, one per skill.

The student can

- Attempt to answer
- Ask for a hint

If the student is incorrect

- scaffold questions start
- students are prompted to answer steps
- after hints exhausted, system provides the answer

System tracks which scaffold questions students answer correctly, how many hints they need, how long it takes, and many other variables.

Problem Set "8thGradeMCAS" id:[1]

1) **Assistment #1474 "1474 - 1998MCASNum31a"**
At the end of every 2nd mile of the Boston Marathon, a typical marathon runner takes about 4 ounces of water. At this rate, how many ounces of water would an average runner take in an entire 26 mile marathon?
Fill in:
✓ 52.4
✓ 52

Scaffold:
First, you need to find out **how many times** a runner takes the water during the entire marathon.
Fill in:
✓ 13
✓ 13.1

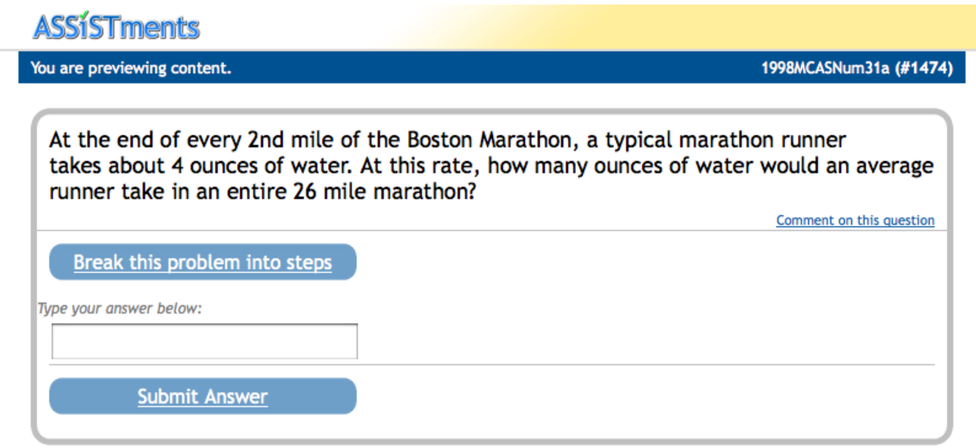
Hints:

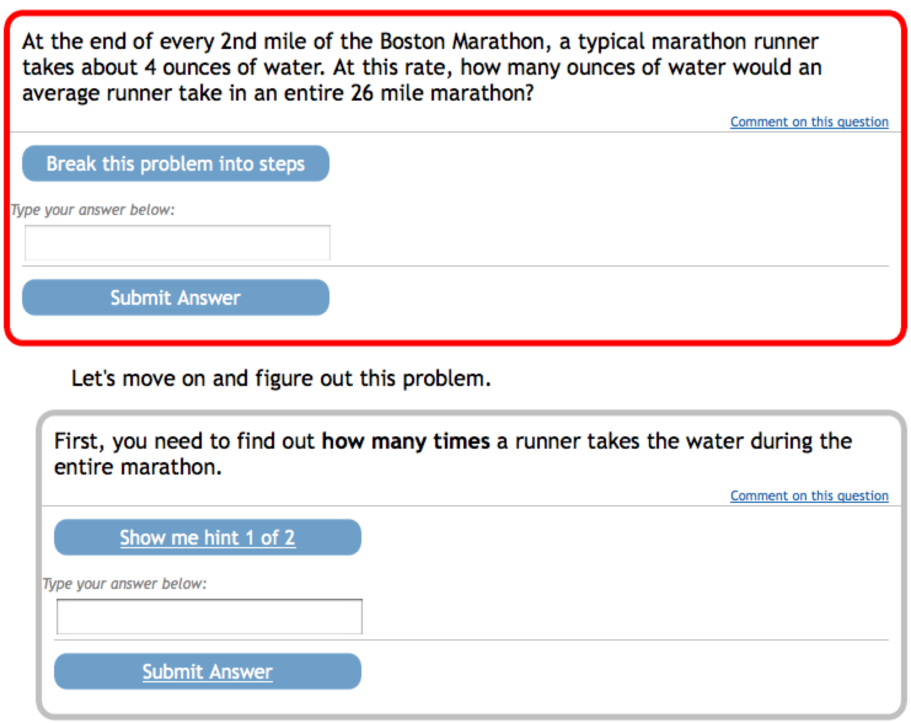
- A runner typically takes water every 2 miles.
  Divide 26 miles by 2 miles to get an estimate of how many times a runner takes water in the marathon.
- 26 divided by 2 is 13. Please enter 13

Scaffold:
Right. A runner will take water 13 times during the race. **How many ounces** of water would an average runner take in the entire 26 mile marathon?
Fill in:
✓ 52
✓ 52.4

Hints:

- You need to multiply the **number of times** a runner will take water by the **number of ounces** of water each time.
- A runner will take water **13** times during the marathon.
  A runner takes about **4** ounces of water each time.





The results all get summarized in several types of reports: teacher, class, student, skill, etc; online access to users, can study how they learn

Common goal: estimate skill mastery

<u>Long story short:</u> often use cognitive diagnosis models to estimate student skill mastery profiles, but high dim data makes this difficult.

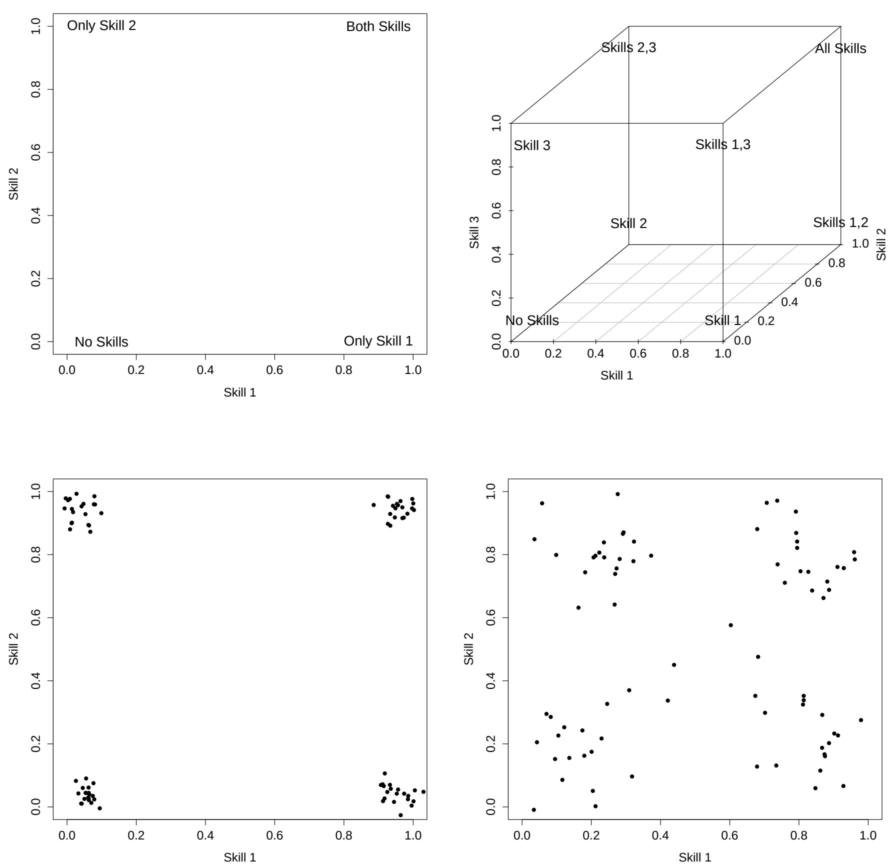
Ex: Dynamic Inputs, Noisy "and" Gate model (DINA):

$$P(Y_{ij} = 1 | \eta_{ij}, s_j, g_j) = (1 - s_j)^{\eta_{ij}} g_j^{1-\eta_{ij}}$$

where $\eta_{ij} = \prod_{k=1}^{K} \alpha_{ik}^{q_{jk}}$; $\alpha_{ik} = 1$ if student i has skill k, $= 0$ if not.

2^K possible skill set profiles $\alpha_i \in \{0,1\}^K$ (e.g. $\alpha_1 = (0,1,0)$).

True skill set profiles are corners of a K-dim hypercube.



The data we can collect:

- Student response matrix (Y)

$$Y = \begin{bmatrix} y_{1,1} & y_{1,2} & \dots & y_{1,J} \\ \vdots & \ddots & & \vdots \\ y_{N,1} & y_{N,2} & \dots & y_{N,J} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 1 \\ \vdots & \ddots & & \vdots \\ NA & 1 & \dots & 0 \end{bmatrix}$$

N students, J items

$Y_{ij} = 1$ if student i answered item j correctly; 0 if incorrectly; NA if not answered

- Assignment matrix of skills needed for each item (Q)

$$Q = \begin{bmatrix} q_{1,1} & q_{1,2} & \dots & q_{1,K} \\ \vdots & \ddots & & \vdots \\ q_{J,1} & q_{J,2} & \dots & q_{J,K} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ \vdots & \ddots & & \vdots \\ 0 & 1 & \dots & 1 \end{bmatrix}$$

J items, K skills

$Q_{jk} = 1$ if item j requires skill k; 0 if not.

One estimate for α_{ik} is the Capability Matrix (Nugent, Ayers, Dean)

$$B_{ik} = \frac{\sum_{j=1}^{J} I_{Y_{ij} \neq NA} \cdot Y_{ij} \cdot q_{jk}}{\sum_{j=1}^{J} I_{Y_{ij} \neq NA} \cdot q_{jk}}$$

B_{ik}: % of items student i answered correctly for skill k.

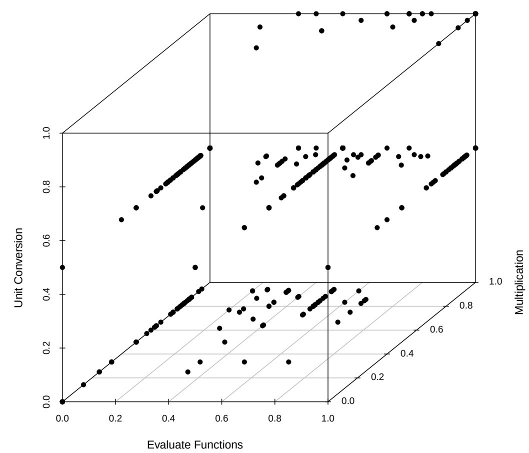
B_{ik} scales for the number of items seen; reduces influence of over-represented skills; incorporates missingness

$$B_{ik} = \hat{\alpha}_{ik} \in \{0, 1\}$$

Maps students into a unit hyper-cube (like CDM estimates).

Datasets:

- Assistments: 551 students, 3 Skills (Evaluating Functions, Multiplication, Unit Conversion)




```
assist3d<-read.table("assist3d.txt")
dim(assist3d)  ##551 students; 3 var
library(scatterplot3d)  ##need to install
scatterplot3d(assist3d,xlab="....",ylab="....",zlab="....",pch=16)
library(rgl)  ##need to install
plot3d(assist3d,xlab="....",ylab="...n",zlab="...",size=5)
```

- Assistments: 344 students; 13 skills

```
assist13d<-read.table("assist13d.txt")
dim(assist13d)  ##344 students; 13 var
pairs(assist13d)  ##scatterplots for each pair of variables
```

- Assistments: 1000 students; 20 skills

```
assist20d<-read.table("assist20d.txt")
dim(assist20d)  ##1000 students; 20 var
pairs(assist20d[,1:10])  ##just looking at a few
table(assist20d[,1]); table(assist20d[,2]); table(assist20d[,3])
```

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

<u>Goal:</u> 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 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: an algorithm that links observations/groups in order of closeness in a hierarchically linked structure; generates n possible partitions

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 <u>and</u> *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*

<u>Required:</u> 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:*

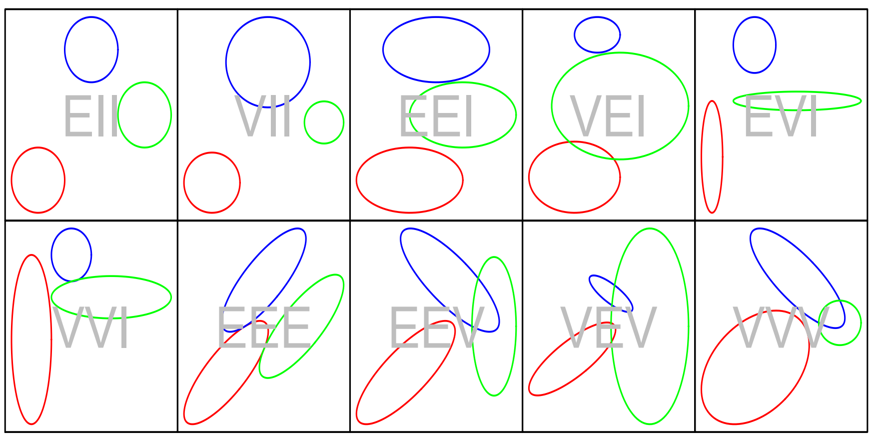
There are two subfields:

- 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.

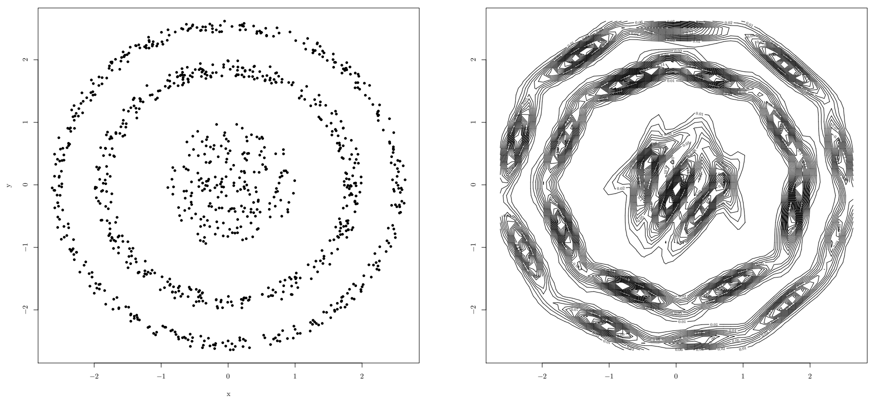
The models are fit using the Expectation-Maximization Algorithm:

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:

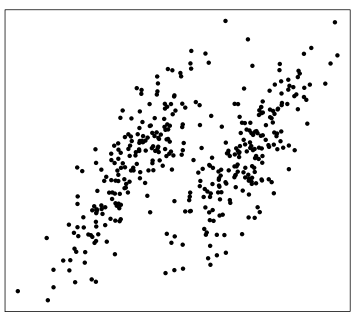
In general, need to be careful about how you interpret the components (whether or not they represent true groups in the population).

Nonparametric Clustering:

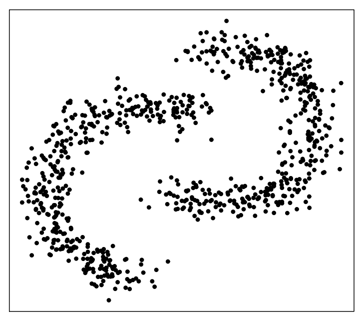
Often we just associate groups with high frequency areas.

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

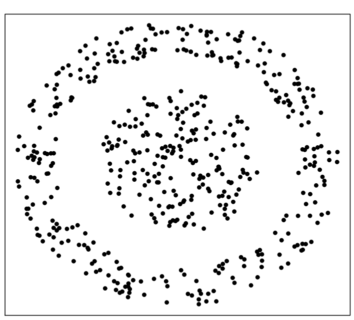
Gives the following definition: contiguous, densely populated areas of feature space, sep- arated by contiguous, relatively empty regions (Carmichael, George, Julius).



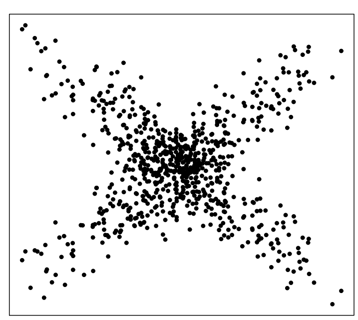
(a)



(b)



(c)



(d)

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

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

Each mode of the density estimate $\Longleftrightarrow$ 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

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 C_k are also in the same cluster in C'

Rand Index:

Adjusted Rand Index (ARI): motivated by seeing that RI does not range over the 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 of them against the truth. Pick the 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

We've only been looking at structure for observations that only have 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.

Notes:

Notes:

Review/Takeaways