Handout w/ Notes

## 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:*

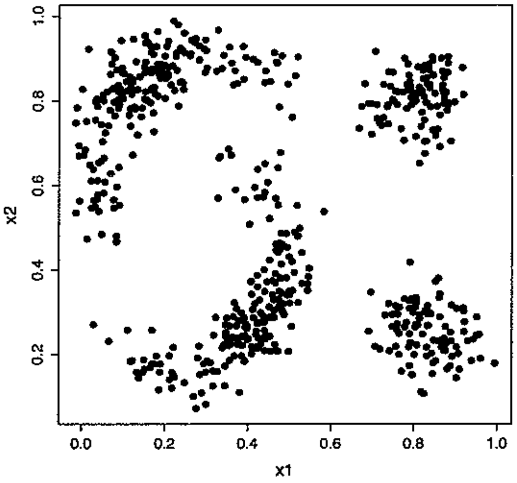
- example of "Unsupervised Learning" - learning without labels
- Given vectors $\mathbf{X} = (X_1, X_2, ..., X_p)$, goal is to "understand" or describe the joint distribution $p(\mathbf{X})$ of these vectors
- Organize, Summarize, Categorize, Explain
- Infer properties of $p(\mathbf{X})$ without any 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:*

Some ex) clustering log files
student behaviors
skill mastery
eye tracking
items, teachers, classes

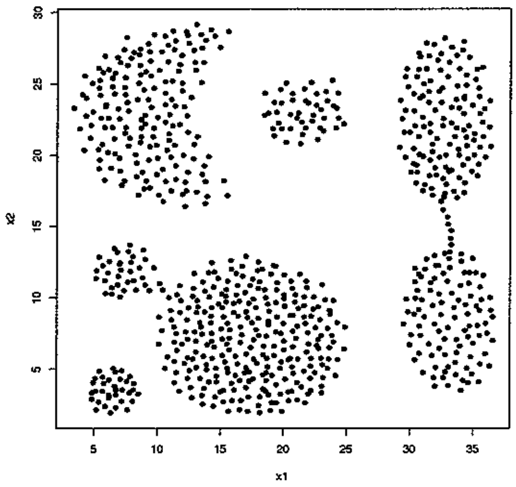
*Datasets:*

- 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



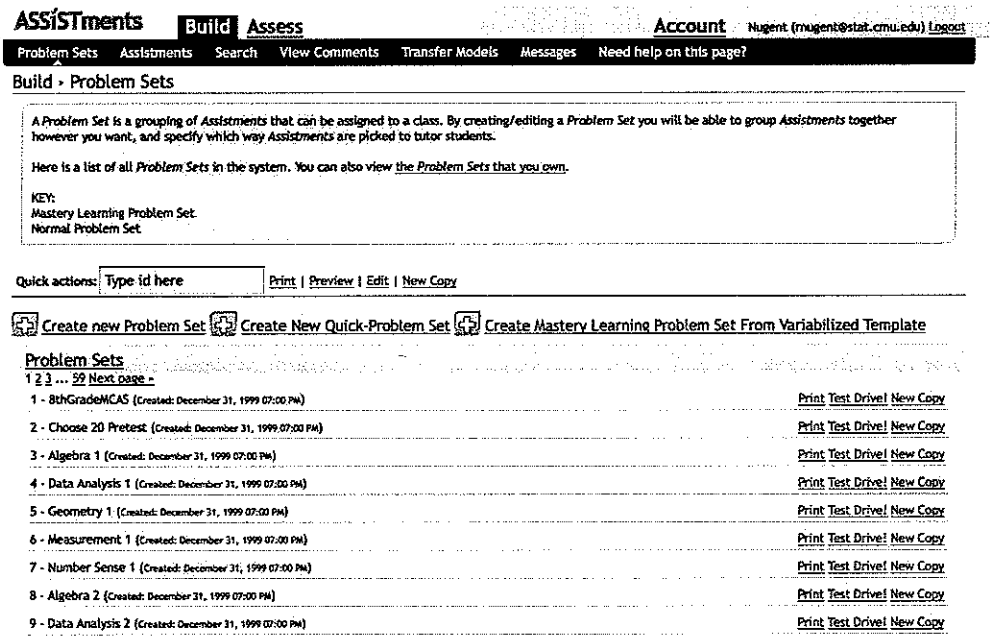
```
aggregation<-read.table("aggregation.txt")
aggr.data<-aggregation[,1:2]
aggr.labels<-aggregation[,3]
plot(aggr.data,xlab="x1",ylab="x2",pch=16)
```

## 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



Build › Problem Sets

A *Problem Set* is a grouping of *Assistments* that can be assigned to a class. By creating/editing a *Problem Set* you will be able to group *Assistments* together however you want, and specify which way *Assistments* are picked to tutor students.

Here is a list of all *Problem Sets* in the system. You can also view the *Problem Sets* that you own.

KEY:
Mastery Learning Problem Set
Normal Problem Set

Quick actions: Type id here Print | Preview | Edit | New Copy

Create new Problem Set Create New Quick-Problem Set Create Mastery Learning Problem Set From Variabilized Template

Problem Sets
1 2 3 ... 59 Next page ›

1 - 8thGradeMCAS (Created: December 31, 1999 07:00 PM) Print Test Drive! New Copy
2 - Choose 20 Pretest (Created: December 31, 1999 07:00 PM) Print Test Drive! New Copy
3 - Algebra 1 (Created: December 31, 1999 07:00 PM) Print Test Drive! New Copy
4 - Data Analysis 1 (Created: December 31, 1999 07:00 PM) Print Test Drive! New Copy
5 - Geometry 1 (Created: December 31, 1999 07:00 PM) Print Test Drive! New Copy
6 - Measurement 1 (Created: December 31, 1999 07:00 PM) Print Test Drive! New Copy
7 - Number Sense 1 (Created: December 31, 1999 07:00 PM) Print Test Drive! New Copy
8 - Algebra 2 (Created: December 31, 1999 07:00 PM) Print Test Drive! New Copy
9 - Data Analysis 2 (Created: December 31, 1999 07:00 PM) Print Test Drive! New Copy

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.

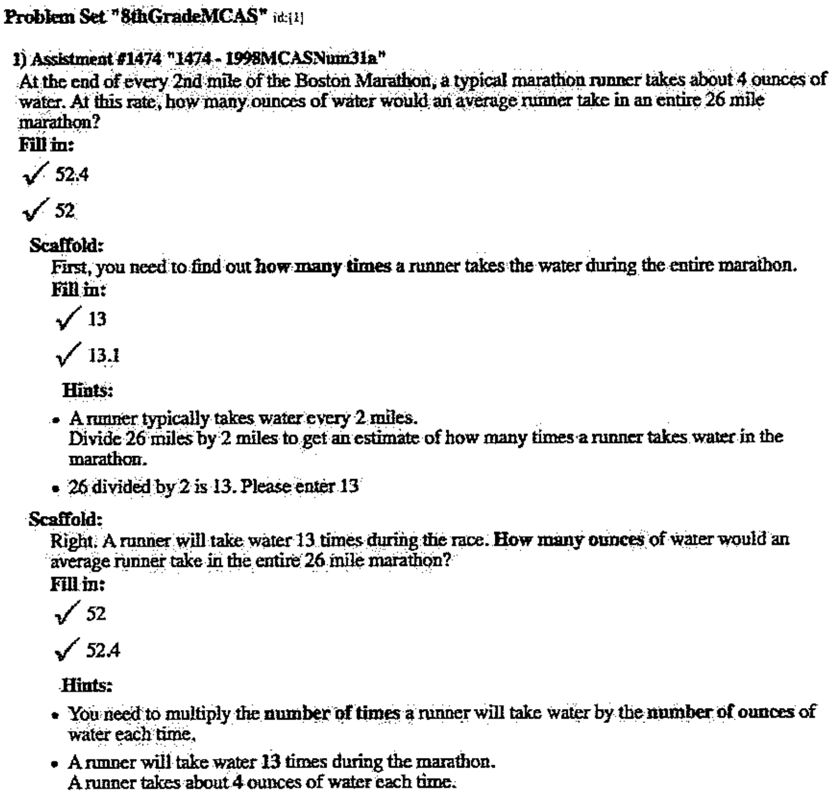
The student can

- 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?

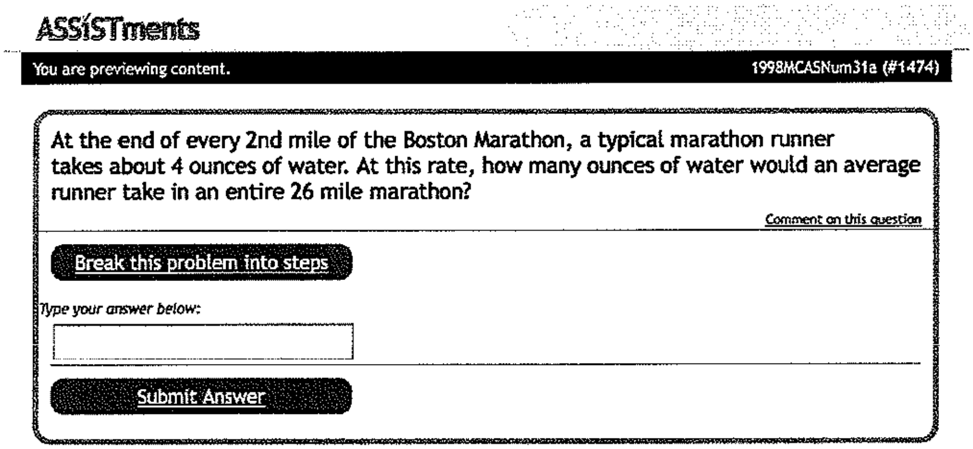
**Fill in:**

✓ 52

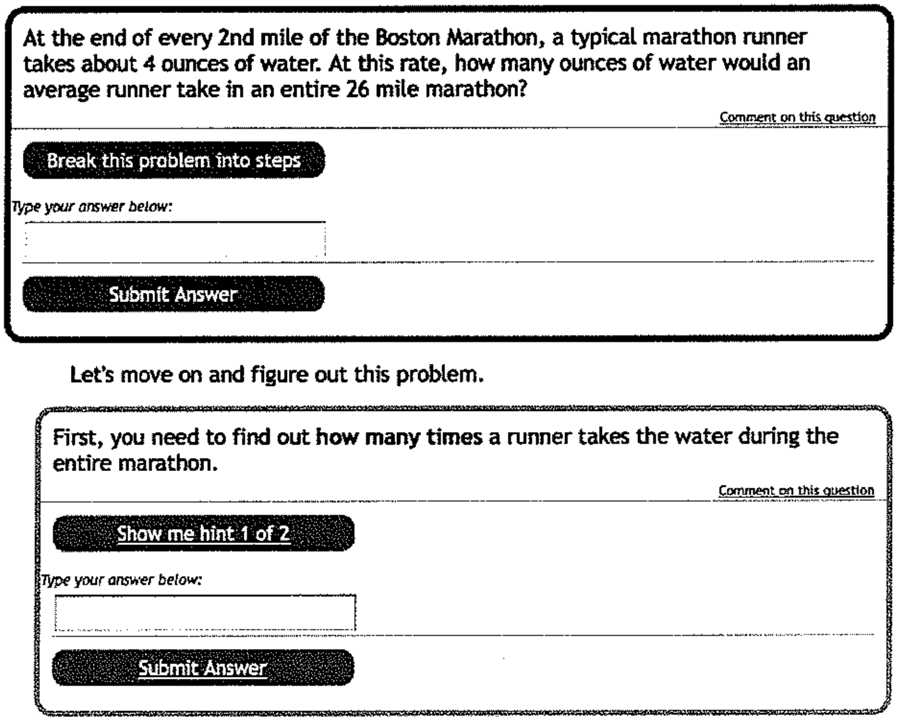
✓ 52.4

**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.



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?


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?

Let's move on and figure out this problem.

First, you need to find out **how many times** a runner takes the water during the entire marathon.

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

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

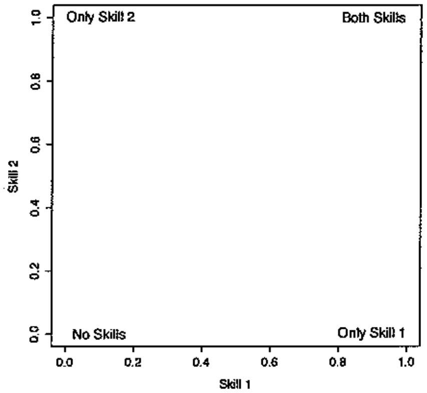
*Ex:* Dynamic Inputs, Noisy "and" Gate model (DINA):

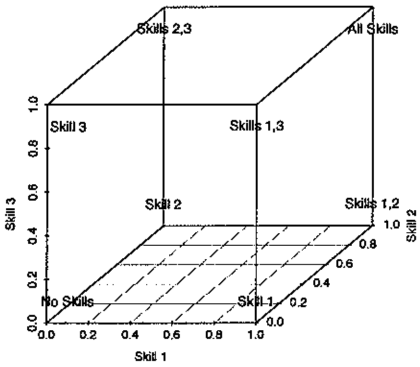
$$P(Y_{ij} = 1 | \eta_{ij}, s_j, g_j) = (1 - s_j)^{\eta_{ij}} g_j^{1-\eta_{ij}}$$

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

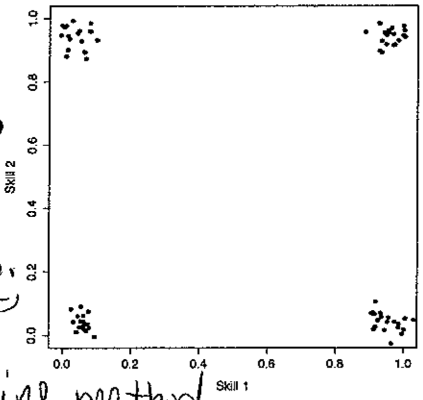
$2^K$ possible skill set profiles $\alpha_i \in \{0,1\}^K$ (e.g. $\alpha_1 = (0,1,0)$).

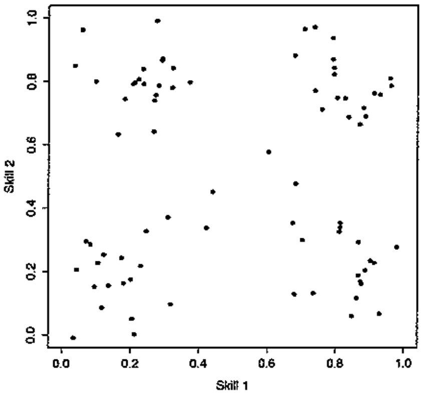
True skill set profiles are corners of a $K$-dim hypercube.





what we hope the data look like; any clustering method could find these groups →





← what actually happens

*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 $\alpha_{ik}$ 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 $k$.

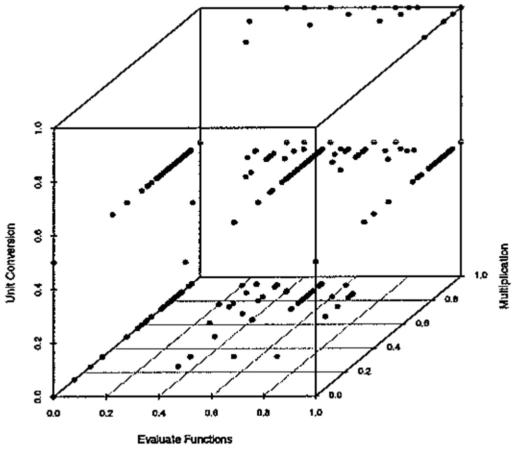
$B_{ik}$ scales for the number of items seen; reduces influence of over-represented skills; incorporates missingness

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

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**

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: mean, median, prototype/representative obs
- Spread: stdev, variance, range, quartile
- Shape: Gaussian, spherical, ellipse, curvilinear

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

$I_{x_i \in C_k} = 1$ if $x_i$ is assigned to cluster $C_k$

$\sum_{x_i \in C_k}$ shorthand for $\sum_{i=1}^{n} I_{x_i \in C_k}$ (only sum obs in cluster $C_k$)

### 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:* $X = \{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_n\} \in \mathbb{R}^p$ (just using reals; could have categorical data)

n obs in p-dim space $x_k$ refers to $k^{th}$ var
$x_i$ refers to $i^{th}$ obs

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

$d(i,j)$ $1 \le i, j \le n$ between pairs of obs $x_i, x_j$

*Properties of a Distance:*

Minimal set
1) symmetry $d(i,j) = d(j,i)$
2) non-negativity $d(i,j) \ge 0$
3) Identification $d(i,i) = 0$

can define a coefficient of dissimilarity using the first three

Gives us a metric
4) Definiteness $d(i,j) = 0$ iff $i = j$
5) Triangle Inequality $d(i,j) \le d(i,k) + d(j,k)$

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

$$n\times n \quad \begin{array}{cc} & \begin{array}{cccc} O_1 & O_2 & \cdots & O_n \end{array} \\ \begin{array}{c} O_1 \\ O_2 \\ \vdots \\ O_n \end{array} & \left[\begin{array}{cccc} & & & \\ & & & \\ & & & \\ & & & \end{array}\right] \end{array}$$

element i,j = $d(i,j)$

symmetric

what does the diagonal look like?

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

$$d(i,j) = \|\underline{x}_i - \underline{x}_j\| = \sqrt{\sum_{k=1}^{p} (x_{ik} - x_{jk})^2}$$

used for continuous var

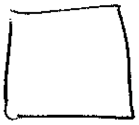
satisfies all five properties; large $d(i,j) \rightarrow$ very different/dissimilar obj

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)

Often used w/ density estimates

looking at grid of values



Why is the structure evident? What happens in practice?

observations are in order of cluster

what does diagonal look like?

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

will look all mixed up

*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

Flat partition: generates one set of K clusters; do not know relationships among clusters

*Hierarchical Partitioning:* Agglomerative vs Divisive

↳ generates several nested sets of clusters

each cluster in a given partition is the union of one or more clusters in the next or most recent partition: $P_K$ = clustering into K groups

Agglomerative: Start w/ $P_n$, merge two "closest" clusters to get $P_{n-1}$; repeat until have $P_1$

**(Agglomerative) Hierarchical Linkage Clustering:** an algorithm that links observations/groups in order of closeness in a hierarchically linked structure; generates $n$ possible partitions

Divisive; Start w/ $P_1$, split into two clusters, apply splitting recursively until have $P_n$

Define distance $d(C_1, C_2)$ between clusters $C_1, C_2$ as a function of a distance/dissimilarity between pts in those clusters $x \in C_1, y \in C_2$

1) Start w/ every obs in its own cluster

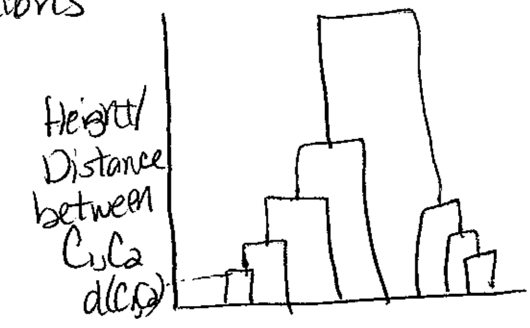
2) Find $\min_{C_1, C_2} d(C_1, C_2)$; merge $C_1, C_2$

search all pairs of clusters

3) repeat until have one cluster

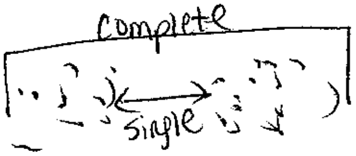
This hierarchical structure is stored in a *dendrogram.* (type of binary tree)

- root of the tree represents everything (top of the tree)
- terminal nodes → observations
- interior node → cluster
- subtree is a partition



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

$$d(C_1,C_2) = \min_{x\in C_1, y\in C_2} d(x,y)$$

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

*Complete Linkage:* intergroup distance: largest possible distance

$$d(C_1,C_2) = \max_{x\in C_1, y\in C_2} d(x,y)$$

Characterized by splitting the data up into more compact subsets

*Other types of Linkage:*

- Average: $d(C_1,C_2) = \operatorname{avg}_{x\in C_1, y\in C_2} d(x,y)$
- Centroid: $d(C_1,C_2) = \|\bar{X}_{C_1} - \bar{X}_{C_2}\|$ distance between the mean vectors
- Ward's $d(C_1,C_2) = \frac{2\,|C_1|\cdot|C_2|}{|C_1|+|C_2|}\|\bar{X}_{C_1}-\bar{X}_{C_2}\|^2$
$= 2\,\frac{n_{c_1}\cdot n_{c_2}}{n_{c_1}+n_{c_2}}\|\bar{X}_{C_1}-\bar{X}_{C_2}\|^2$

merging two clusters with smallest Ward's dist optimizes the minimization of within cluster distance

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

**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:* distance between all pairs of obs within a cluster
all distances should be small

$\|x_i - x_j\|$ $\quad x_i, x_j \in C_k$ $\quad$ Cluster $C_k$ is of size $n_k$, i.e. $|C_k| = n_k$
for cluster $C_k$, $\binom{n_k}{2}$ pairs

*Between-Cluster Distance:* distances between all pairs of observations in different clusters, should be larger

$\|x_i - x_j\|$ $\quad x_i \in C_k$, $x_j \in C_{k^*}$ $\quad |C_k| = n_k$, $|C_{k^*}| = n_{k^*}$ $\quad$ have $n_k \cdot n_{k^*}$ pairs

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

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

$$WC = \underbrace{\sum_{k=1}^{K}}_{\text{sum over clusters}} \underbrace{\sum_{x_i \in C_k}}_{\text{observations}} \|x_i - \bar{x}_k\|^2$$

using squared Eucl Dist

lower criterion → higher quality

<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: need to pick A to minimize

  for a given i $\sum_{k=1}^{K} \sum_{A} \|x_i - \bar{x}_k\|^2$ ↗ not really necessary; can just assign all obs in one pass

  smallest when $\|x_i - \bar{x}_k\|^2$ smallest

  $A \rightarrow$ closest center $\bar{x}_k$

- Given the current assignments: need to pick $m_k$ to minimize

  for a given cluster $C_k$, $\sum_{x_i \in C_k} \|x_i - m_k\|^2 = \sum_{x_i \in C_k} \sum_{l=1}^{p} (x_{il} - m_l)^2$

  $\sum_{x_i \in C_k} -2 \sum_{l=1}^{p} (x_{il} - m_l) = 0$

  $\sum_{l=1}^{p} (\sum_{x_i \in C_k} x_{il}) - n_k m_l = 0$

  each $m_l = \frac{\sum x_{il}}{n_k} = \bar{x}_k$

*In practice:*

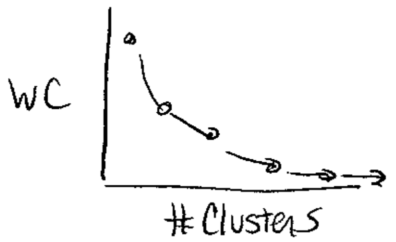
- First few steps correspond to large drops in the criterion; later steps correspond to negligible drops.
- Use $K$ randomly chosen observations as the starting centers (but don't have to; can choose specific centers)
- Have an idea of what $K$ should be in advance

What if we don't know $K$? How do we choose?

If we increase $K$, what happens to the within-cluster criterion?

$\sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \bar{x}_k\|^2$ increase K by 1; move one pt over to the new cluster WC criterion decreases (new cluster has WC=0)

We use an *elbow graph* to determine a "useful" $K$. In general, splitting moves obs closer to means, reduces criterion



What do we look for in the elbow graph? Where do the large drops stop? Where do negligible drops begin?

see K-Means Consensus for more (Steinley)

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

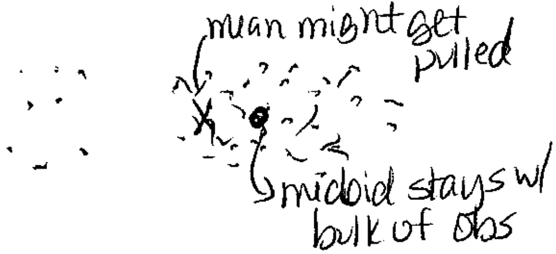
look for stable values of $K$; where do the negligible drops begin consistently?

Some research advocates Steinley
1) picking a stable $K$
2) run K-means 5000 times (say) with different starting sets of centers
pick solution w/ lowest criterion

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



Medoid is always an obs in the data set/cluster (mean does not have to be)

Given a starting set of $K$ observations (medoids), alternate between:
$X_1\ X_2, \dots, X_K$

- Assign each observation to the closest medoid.

  for $X_i$, $\arg\min_j \|X_i - X_j\|^2$ $j \in 1,2,\dots,K$

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

  $\min_k \sum_{X_i \in C_k} \|X_i - X_k\|^2$ $X_k \to$ medoid in cluster $C_k$

Check each obs as a potential medoid

(sometimes K-medoids uses the criterion $\sum \|X_i - X_k\|$ - sum of distances instead of squared distances)

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?)*
tends to be more stable though

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

Observations are considered a sample from unknown population density $p(x)$

Goal: estimate $p(x)$ and its properties $\left\{ \begin{array}{l} \text{mean} \\ \text{var} \\ \text{modes (location, size)} \\ \text{tails} \\ \text{skew, etc} \end{array} \right.$

Two parts: 1) density estimation
2) finding/estimating/characterizing the properties of the density (estimate)

There are two subfields:

- Parametric associates a specific model w/ the density
  model has a $\underbrace{\text{set of parameters}}_{\text{use these to characterize the clusters}}$ Gaussian, Beta, Unif, Exp, etc
- Nonparametric
  associates no model; looks at contours of the density to find cluster info
  modes ⟺ groups/clusters

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

$$p(x) = \sum_{g=1}^{G} \pi_g \cdot p_g(x; \theta_g) \qquad \sum_{g=1}^{G} \pi_g = 1 \qquad 0 \le \pi_g \le 1$$

weighted combination of individual densities

$\theta_g \rightarrow$ set of parameters specific to the density
$N \rightarrow \mu, \Sigma \qquad t \rightarrow df$
$\text{Unif} \rightarrow \underline{a}, \underline{b}$
Beta → shape, scale, etc

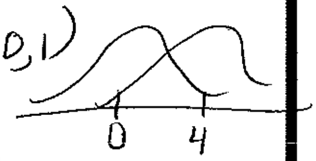
What type of densities do we fit?

Most often Gaussians

Assume $p(\underline{x}) = \sum_{g=1}^{G} \pi_g \cdot p_g(\underline{x}; \mu_g, \Sigma_g)$

e.g.

Two group mixture: $p(x) = 0.5 \times N(x; 4, 1) + 0.5 \times N(x; 0, 1)$

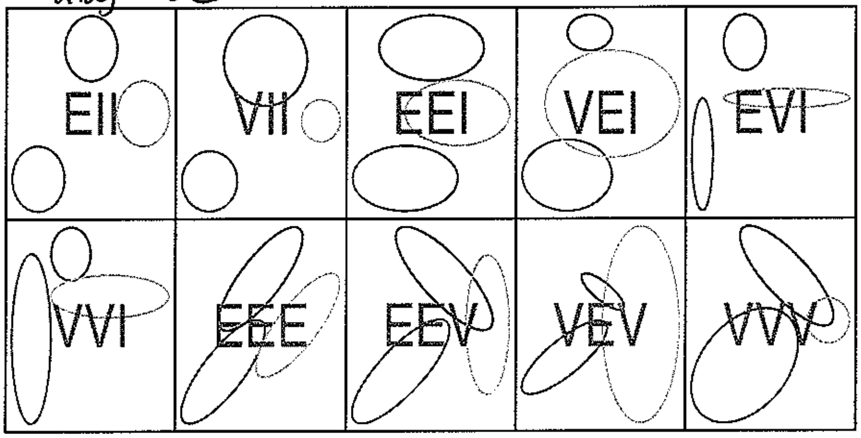


Gaussians can have wide variety of covariance structures

spheres, ellipses
diagonal $\Sigma$

(can sometimes fit a noise component)

In R, these 10 models are considered



Three Letters: Volume/Shape/Orientation

Equal/Variable: Volume
Equal/Variable: Shape
Equal/Variable/Axis Aligned: Orientation

Choosing the "Best" Model:

Pick the model that maximizes the Bayesian Information Criterion.

Model: $M_i \quad i = 1, \ldots, 10$

$BIC(M_i) = 2\log L(M_i) - p \cdot \log(n)$

$L(M_i)$
↳ likelihood of the data given $M_i$

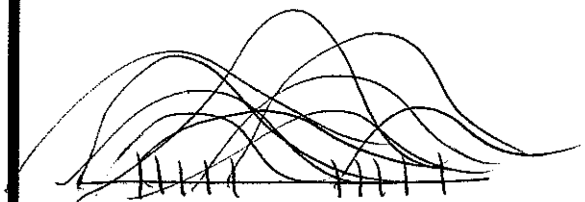
p = # independent parameters
for each g, have to estimate $\mu_g, \Sigma_g$ ($\theta_g$)
trying to keep you from overfitting with lots of $p_g$

n = # of obs; again, don't overfit large data sets

Looking at a Two Group Mixture:

possible # of groups

goes through, e.g., G=1, ..., 9



estimates the best model for each G then chooses the best overall

- can plot the BIC values as function of G

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

$\mu_g$ $\Sigma_g$ $\pi_g$

$$\hat{p}(\underset{\wedge}{x}) = \sum_{g=1}^{G} \hat{\pi}_g \hat{p}_g(x; \hat{\mu}_g, \hat{\Sigma}_g)$$

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

Decompose $\Sigma_g = \lambda_g D_g A_g D_g^T$

$\lambda_g \rightarrow$ largest eigenvalue; volume of the $g^{th}$ component

$A_g \rightarrow$ diagonal matrix of scaled eigenvalues; shape of $g^{th}$ component

$D_g \rightarrow$ matrix of eigenvectors; orientation of the $g^{th}$ component

The models are fit using the Expectation-Maximization Algorithm:

EM → used to estimate MLE in incomplete data (missing cluster labels)

E-step: compute conditional expectation of cluster labels M-step: update

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

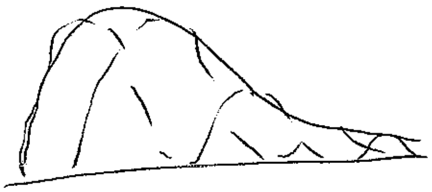
- the name of the model
- the estimated means and covariance
- the estimated membership probabilities
- the cluster assignments $\rightarrow \max_k Z_{ik}$ = cluster assn for obs i

$$Z_{ik} = \frac{\hat{\pi}_k p_k(x_i | \hat{\theta}_k)}{\sum_{g=1}^{G} \hat{\pi}_g p_g(x_i | \hat{\theta}_g)}$$

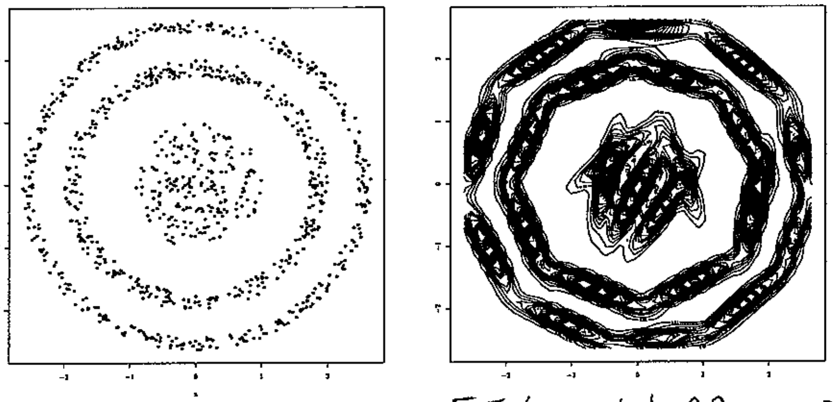
Common assumption: each component represents a population group

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

Goal: maximize the likelihood of the data coming from the model



Can fit any continuous density to a given degree of accuracy with a mixture of Gaussians



17% correctly Classified

EEV model, 23 components

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

measure how much they overlap?
density contours? entropy?

What about Gaussian clusters with noise?

mixture of true groups + some random uninformative noise

*Two options:*

1) Treat the noise as a Gaussian; what would it look like?

- center
- variance

2) Model it as a uniform noise component
- component that "soaks up" the extra obs/outliers/noise

$$\sum_{g=1}^{G} \pi_g \cdot p_g(x; \mu_g, \Sigma_g) + \pi_0 V$$

prior prob of being noise

volume of data range

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

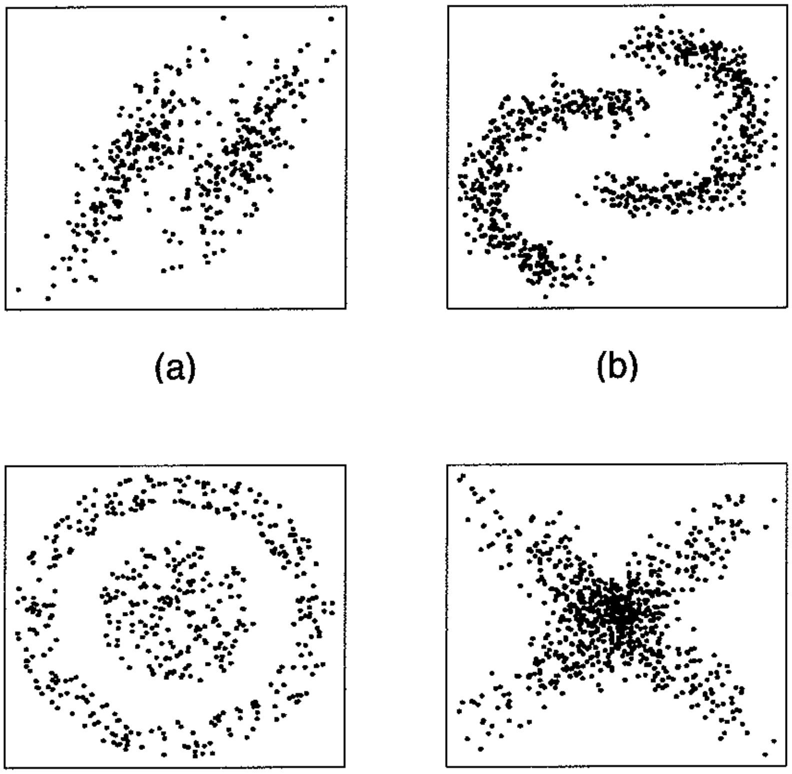
## Nonparametric Clustering:

Often we just associate groups with high frequency areas.

more common to see data

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). size, shape doesn't matter



two groups

(a)

(b)

two groups

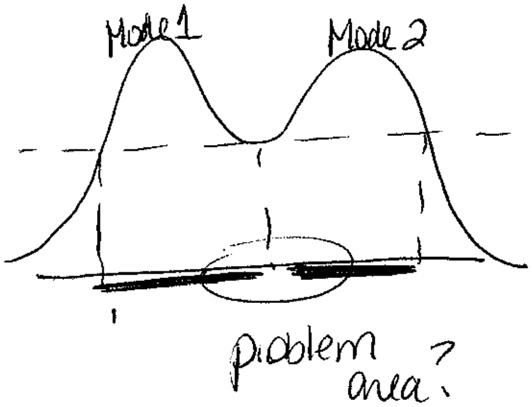
two groups

(c)

(d)

one group

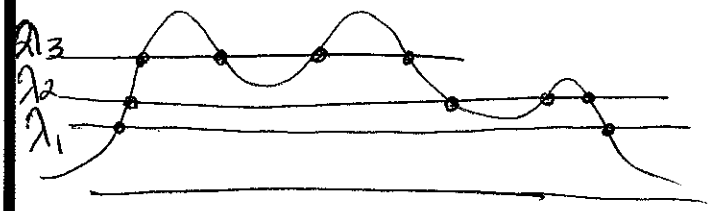
NP Goal: find the modes of a density $p(x)$ (or $\hat{p}(x)$); assign observations to the "domain of attraction" of a mode *(contrast with MBC)*



how do the "tails" get assigned?

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

Level Sets of a Density: cross-sections of a density (estimate)



$L(\lambda; p(x)) = \{x \mid p(x) > \lambda\}$

looking at connected components of level set
each conn comp represents a mode

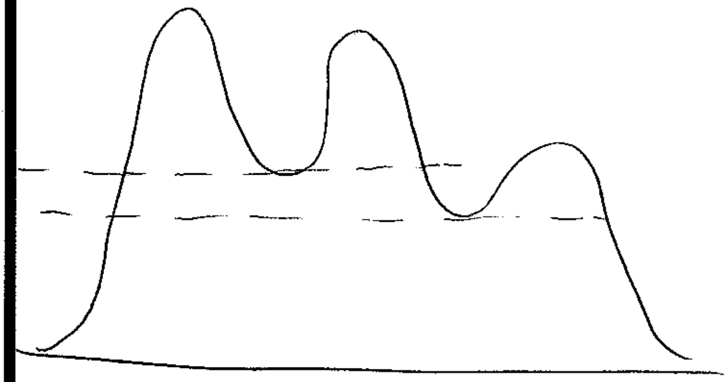
Level sets are "nested"; increase $\lambda$, get smaller level set contained in lower level sets

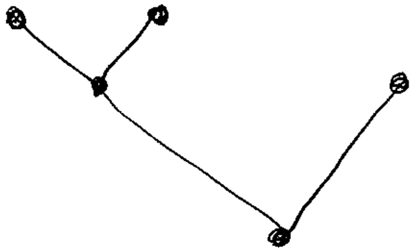
NP Goal: find the modes of a density $p(x)$ (or $\hat{p}(x)$); assign observations to the "domain of attraction" of a mode; build cluster tree of $p(x)$; *(NPEx.pdf)*

↳ see other handout

Cluster tree → binary, recursive

- Starts w/ root node, represents all obs/feature space
- increase $\lambda$ until level set splits into two conn comp
- create one daughter node for each conn comp
  split obs & feature space accordingly
- recurse for each daughter node
  e.g. start w/ "left" node, then "right" node





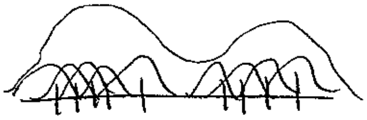
Unlike other clustering procedures, NP clustering is very dependent on the density estimate $\hat{p}(x)$.

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

*Kernel Density Estimate:* common nonparametric density estimate

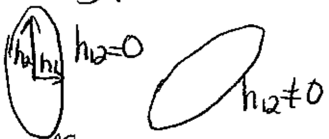
1-dim $\hat{p}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x-x_i}{h}\right)$ h → bandwidth of the Kernel

$K(t)$ = Kernel



p-dim $\hat{p}(x) = \frac{1}{nh^p}\sum_{i=1}^{n} K\left(\frac{\underline{x}-\underline{x}_i}{h}\right)$ ← assumption here is that have single smoothing parameter

$\hat{p}(x) = \frac{1}{n|H|}\sum_{i=1}^{n} K(H^{-1}(\underline{x}-\underline{x}_i))$ $H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}$



Choice of kernel: → usually symmetric shape, satisfies $\int_{-\infty}^{\infty} K(x)dx = \underline{1}$

- Gaussian $N(0,1)$

  $K(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2}$ infinite support, all obs contribute

- Epanechikov

  $K(t) = \frac{3}{4}(1-\frac{1}{5}t^2)/\sqrt{5}$ for $|t| < \sqrt{5}$ compact support

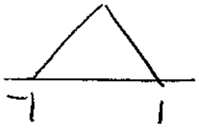
- Biweight/Triweight

  $K(t) = \frac{15}{16}(1-t^2)^2$ for $|t|<1$ Beta(3,3)

  $K(t) = \frac{35}{32}(1-t^2)^3$ for $|t|<1$ Beta(4,4)

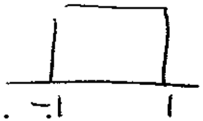
- Triangular

  $K(t) = 1-|t|$ for $|t|<1$



- Box

  $K(t) = \frac{1}{2}$ for $|t|<1$



*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:

- interpretation not dependent on sample size (n scaled out)
- can compare across clustering algorithms / standardized
- requires true labels
- can be hard to determine if # clusters(k) ≠ # of groups

  R function matchClasses

What if the clustering algorithm is not completely deterministic?

e.g. Kmeans - solution depends on starting centers

could find measure (e.g. % correct) several times (as many as feasible)

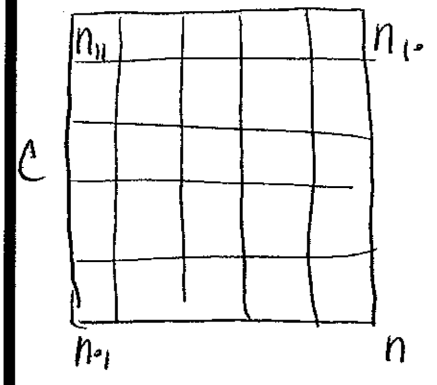
average the measures?

median?

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.

Two clusterings: $C_k$: $k=1,2,\ldots,K$ $\quad C'_{k'}$: $k'=1,2,\ldots,K'$ one of these could be the truth

$C'_i$



summarize this table w/ four #'s

$N_{11}$ = # of obs pairs that are in same $C_k$ and same $C'_{k'}$

$N_{10}$ = # of obs pairs in same $C_k$ but different $C'_{k'}$

$N_{01}$ = # of obs pairs in different $C'_{k'}$ but same $C_k$

$N_{00}$ = # of obs pairs in different $C_k$ and diff $C'_{k'}$

want $N_{11}, N_{00}$ big; $N_{10}, N_{01}$ small

$$N_{11} + N_{10} + N_{01} + N_{00} = \binom{n}{2}$$

$n_{ij}$ = # of obs in $\underbrace{C_i}_{\text{from partition1}}$ & $\underbrace{C'_j}_{\text{from partition 2}}$

Geometric Mean: $\sqrt[n]{\prod_{i=1}^{n} x_i}$ used to show "central tendency" of a group of #'s

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

$$FM = \sqrt{P(\text{pair of obs in } C'_{k'} \mid \text{in } C_k) \cdot P(\text{pair of obs in } C_k \mid \text{in } C'_{k'})}$$

$$= \sqrt{\frac{N_{11}}{N_{11}+N_{10}} \cdot \frac{N_{11}}{N_{11}+N_{01}}} = \frac{N_{11}}{\sqrt{(N_{11}+N_{10}) \cdot (N_{11}+N_{01})}} \quad 0 \le FM \le 1$$

*Rand Index:*

$$RI = \frac{N_{11}+N_{00}}{\binom{n}{2}}$$ ← # of pairs that are clustered similarly in $C, C'$

$0 \le RI \le 1$ ratio of the # of pairs that cluster together & # of pairs that fall into different clusters in both places over total # of pairs

→ probability that two obs are treated alike in both clusterings

*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

$$\text{Adjusted Index} = \frac{\text{Index} - E[\text{Index}]}{\max \text{Index} - E[\text{Index}]}$$ ← sets the Expected value to zero ← scales so max = 1

$$ARI = \frac{RI - E[RI]}{1 - E[RI]} = \frac{\sum_{k=1}^{K}\sum_{k'=1}^{K'}\binom{n_{kk'}}{2} - \sum_{k'=1}^{K'}\binom{n_{k'}}{2}\sum_{k=1}^{K}\binom{n_k}{2} \Big/ \binom{n}{2}}{\frac{\sum_{k=1}^{K}\binom{n_{k\cdot}}{2} + \sum_{k'=1}^{K'}\binom{n_{\cdot k'}}{2}}{2} - \sum_{k'=1}^{K'}\binom{n_{k'}}{2}\sum_{k=1}^{K}\binom{n_k}{2} \Big/ \binom{n}{2}}$$

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

represents the prob of the two cluster labels disagreeing for an obs

1) Find the "best" mapping $M$ of $C$ to $C'$, want diagonal to as large as possible

2) $\text{error} = 1 - \frac{1}{n} \max_{M} \sum n_{k, M(k)}$ "unmatched mass" left over in confusion matrix/contingency table

(*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. Or the more stable, consistent one. Depends on goals.
- 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:*

$$\hat{p}(x) = \sum_{g=1}^{G} \pi_g \cdot \underbrace{p_g(x; \mu_g, \Sigma_g)}_{\text{most often Gaussian density, different volumes, shapes, orientations}}$$

EM procedure chooses $G, \pi_g, \mu_g, \Sigma_g$

$0 \leq \pi_g \leq 1$ $\sum_g \pi_g = 1$

model has highest BIC

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

*Maximum Membership Probability*: $\tau_i = \max_k z_{ik}$ maximum over each row of the membership prob matrix

*Uncertainty Index:* $1 - \tau_i = UI_i$

$0 \leq UI_i \leq 1$

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

all zero (or close)

Spread out; might see modes/groups of poorly classified obs

Could visualize maximum probability/uncertainty using histograms, etc boxplots by clusters to see which clusters have higher membership probs see also Neighborhoods, Stripes, Stars by Leisch (handout posted)

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

- obs is "closer"/more similar to obs in that cluster than any others
- poorly assigned would mean that there are several clusters that are "close"

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

Define $\bar{d}(x_i, C_k)$ as the average distance from $x_i$ to $x_j \in C_k$

$$\bar{d}(x_i, C_k) = \frac{1}{n_k} \sum_{x_j \in C_k} \| x_i - x_j \| \qquad n_k = |C_k|$$

Let $C_0(i)$ be the cluster to which obs $i$ $(x_i)$ is assigned

Let $C_1(i)$ be the "next closest" cluster

i.e. the cluster that minimizes $\bar{d}(x_i, C_k)$ for $k \neq 0$

would expect $\bar{d}(x_i, C_0(i))$ to be small

$\bar{d}(x_i, C_1(i))$ to be bigger

**NOTE:** can use any distance, not just Euclidean

*Silhouette Measure:*

$$S_i = \frac{\bar{d}(x_i, C_1(i)) - \bar{d}(x_i, C_0(i))}{\max\{\bar{d}(x_i, C_1(i)), \bar{d}(x_i, C_0(i))\}} \leftarrow \text{scales s.t. } 0 \leq |S_i| \leq 1$$

$S_i$ near $\underline{1}$ (large)

$\bar{d}(x_i, C_0(i))$ very small, almost zero

avg distance from $x_i$ to its cluster small; $x_i$ to any other cluster large

$S_i$ near 0 (small)

$\bar{d}(x_i, C_0(i)) \approx \bar{d}(x_i, C_1(i))$

obs $x_i$ is between two clusters "posterior prob" near 0.5

$S_i$ negative → probably in wrong cluster

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

well-assigned, all near $\underline{1}$ ; some high, some low

near zero, or negative poorly assigned

have some well-assigned & some not

There is an analogous silhouette measure for density-based clustering

Rousseauw ← (Statistics & Computing journal) see Longitudinal Notes Handout posted

## 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

Clustering: partitioning observations into groups
↳ could be hard or soft assignment

Need to think about goals/application in advance

What shapes are you interested in?
Maybe it doesn't matter?

Do you need stable, consistent clusters?
Ones w/ statistical properties?
Maybe not?

Do you know the # of groups in advance?
Most likely not. Do you have a plausible range?

How much computational time can you afford?

How do you want to represent the clusters?
Means? Prototypes? Shapes?

Always think about the assumptions of your approach when interpreting your results. Value simplicity and stability.