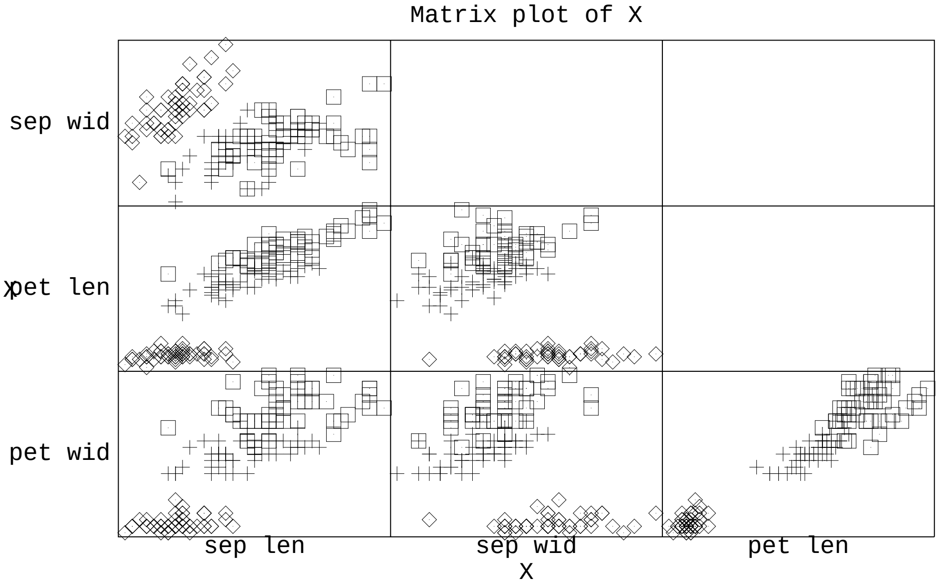
Statistics 5401
31. Clustering

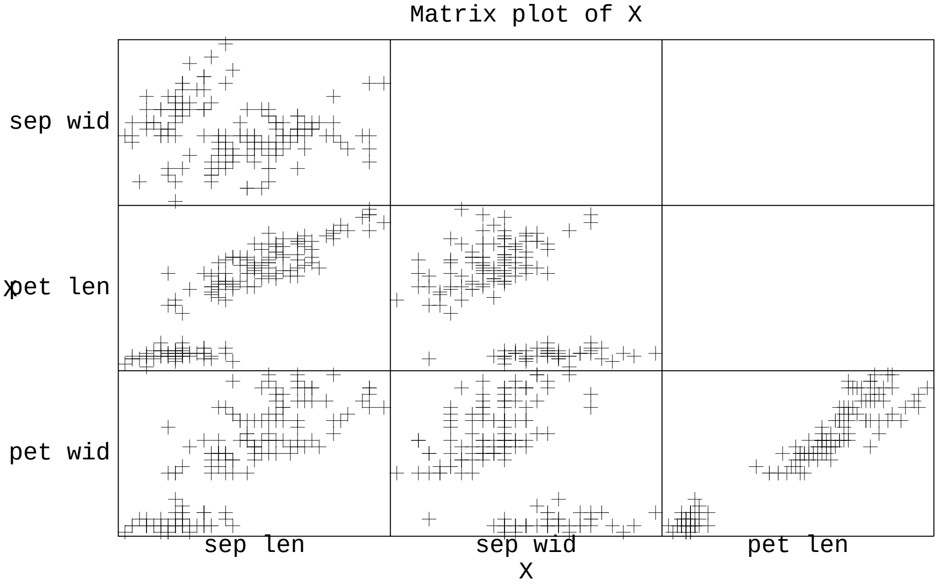
Gary W. Oehlert
School of Statistics
313B Ford Hall
612-625-1557
gary@stat.umn.edu

Clustering versus classification.
We have done some classification. In that problem, we have “sample” data from known groups, and we construct a method to allocate future data into the known groups. We know lots about the groups:
we know how many there are,
we know their prior probabilities,
we have some data from each group so that we can get a handle on the within group density.
Clustering is a completely different kettle of fish.
In clustering, we have a set of data; perhaps they are a sample. We are trying to find groupings, but ...
we don’t know how many groups there are,
we know nothing about the density or distribution of any groups that might be there,
we know nothing about how many data should be in each group.
None the less, we try to find groups.
For example, the iris data. Three species, with four measurements on each plant.

```
Cmd> X <- matrix(vecread(""),5)’
Read from file "˜/JW5data/T11-5.DAT"

Cmd> species <- X[,5]

Cmd> X <- X[,-5]

Cmd> X <- matrix(X,labels:structure("(",\
vector("sep len","sep wid","pet len","pet wid")))

Cmd> plotmatrix(X,lower:T,symbols:\
vector("\1","\2","\3")[species])
```



In classification, we have those classes, and from them make the rule. We'd like to be able to do something similar, but for situations where we didn't know the groups. Ie, work from a plot like

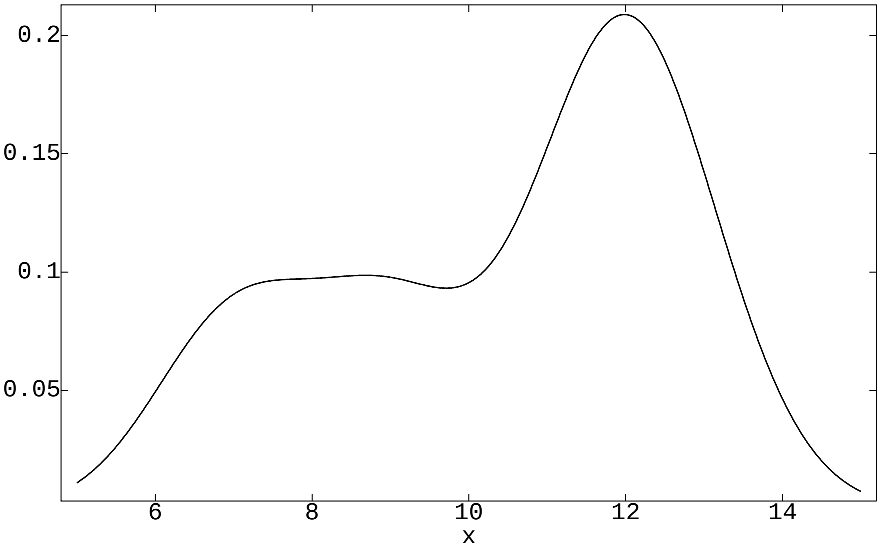


Clustering can be

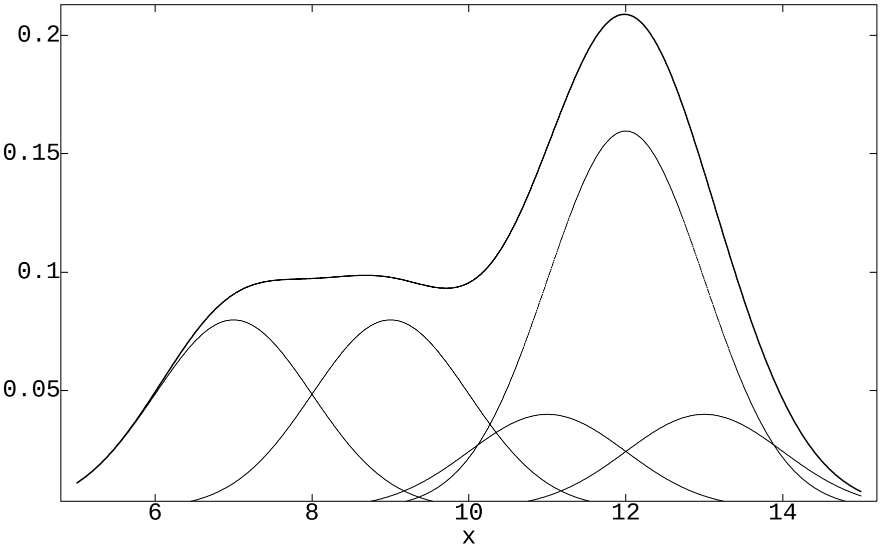
- purely descriptive
- model based
- starting point for more analysis
- a black art
- very frustrating and confusing

An example of model-based clustering.

We believe that there are g populations, each with proportion p_i. Each population has density $f_i()$.
We need to determine g, the proportions p_i, and $f_i()$. Note that we already know all these in a classification problem.
Even if you knew more, it can be a difficult problem. Suppose that you knew that each population was normal with equal variance? Does that make it easy.
How many groups?



Did you say five?



Input for clustering can be of many types. The basic type is the raw data x. That is, we have a vector of length p for each data case. Some clustering algorithms work directly with the data.
From the data we can compute *distances* or *dissimilarities* d_{ij} between every pair of objects

$$d_{ij} = d(x_i, x_j)$$

for some distance function.

We could instead compute *similarities* for all the pairs

$$s_{ij} = s(x_i, x_j)$$

for some similarity function.

Distances or dissimilarities are small when x_i and x_j are close together. Similarities are large when x_i and x_j are close together.

A real, mathematical distance must satisfy

- $d(x, x) = 0$
- $d(x, y) > 0$
- $d(x, y) = d(y, x)$
- $d(x, y) \leq d(x, z) + d(z, y)$

Some clustering methods work even with "pseudo" distances.

Here's an interesting twist: we can also compute similarities (or distances) between variables, not just between objects.

Correlation is an obvious similarity measure; sometimes we use absolute correlation if we don't care about the direction of the relationship.

We can cluster or group variables, as well as cases.

Sometimes we never see the original data, and all we see are distances or dissimilarities. That is, we just have the $n \times n$ matrix of the d_{ij}s. In some contexts, $d_{ij} \neq d_{ji}$ and $d_{ii} \neq 0$.

Other times, all we have are similarities, again in an $n \times n$ matrix, and they also may be nonsymmetric.

It's usually easier to work with symmetry, so you could construct $\tilde{\mathbf{D}} = (\mathbf{D} + \mathbf{D}')/2$ or $\tilde{\mathbf{S}} = (\mathbf{S} + \mathbf{S}')/2$

You can always change a dissimilarity into a similarity, and vice versa, but the transformation is not unique, and the obtained dissimilarity may not be a true distance.

For example, assume $\max s_{ij} = s_{ii} = 1$, then

$$d_{ij} = \frac{1 - s_{ij}}{s_{ij}}$$

and

$$d_{ij} = \sqrt{2(1 - s_{ij})}$$

both satisfy $d \geq 0$ and $d_{ii} = 0$. If the s_{ij} matrix is nonnegative definite, $d_{ij} = \sqrt{2(1 - s_{ij})}$ will also be a true distance.

Euclidean distance

$$d(x, y) = ||x - y|| = \sqrt{\sum_{i=1}^{p} (x_i - y_i)^2}$$

This is highly scale dependent.

Standardized Euclidean distance

$$d(x, y) = \sqrt{\sum_{i=1}^{p} \frac{(x_i - y_i)^2}{s_i}}$$

where s_i is some scale estimate for the ith component.

Manhattan distance

$$d(x,y) = \sum_{i=1}^{p} |x_i - y_i|$$

(Think about city blocks in a grid of streets.) This is also scale dependent.
Standardized Manhattan distance

$$d(x,y) = \sum_{i=1}^{p} \frac{|x_i - y_i|}{s_i}$$

Maximal distance

$$d(x,y) = \max_{i=1}^{p} |x_i - y_i|$$

Just the biggest coordinate difference. This is also scale dependent.
Standardized Maximal distance

$$d(x,y) = \max_{i=1}^{p} \frac{|x_i - y_i|}{s_i}$$

Minkowsky distance

$$d(x,y) = \left(\max_{i=1}^{p} |x_i - y_i|^q \right)^{1/q}$$

$q = 1$ gives Manhattan, $q = 2$ gives Euclidean, $q = \infty$ gives maximal. This can also be scaled.
Generalized distance

$$d(x,y) = \sqrt{(x-y)'\mathbf{A}(x-y)}$$

for some positive definite $\mathbf{A}$. Variance matrices of some sort or another are obvious choices for $\mathbf{A}$.
The distances above are all more or less standard mathematical distances.
Sometimes, we want something more specialized. For example,

$$\sqrt{(\bar{x} - \bar{y})^2 + \left(\log \frac{s_x^2}{s_y^2} \right)^2}$$

measures how far apart the means and variances of the components are.
A standard measure of similarity is the correlation coefficient, or more likely, the absolute value of r.
If you let x_s and y_s be standardized versions of x and y, then

$$r_{xy} = 1 - \frac{||x_s - y_s||^2}{2(n-1)}$$

So if you define

$$d(x,y) = \frac{||x_s - y_s||^2}{(n-1)}$$

then

$$d(x,y) = 2(1 - r_{xy})$$