ETC3250 Business Analytics: Unsupervised classification - k-means

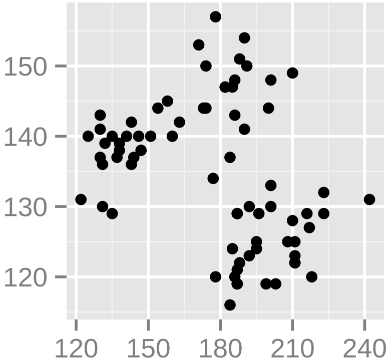
Souhaib Ben Taieb, Di Cook, Rob Hyndman

October 19, 2015

What is cluster analysis?

- The aim of cluster analysis is to group cases (objects) according to their similarity on the variables. It is also often called unsupervised classification, meaning that classification is the ultimate goal, but the classes (groups) are not known ahead of time.
- Hence the first task in cluster analysis is to construct the class information. To determine closeness we start with measuring the interpoint distances.

Cluster this



k-means algorithm

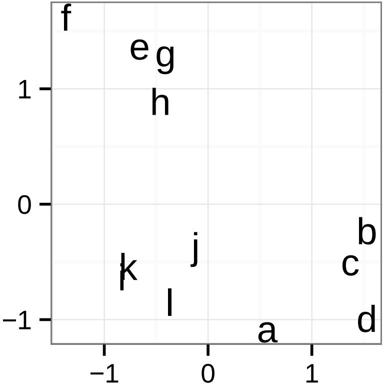
This is an iterative procedure. To use it the number of clusters, k, must be decided first. The stages of the iteration are:

- Initialize by either (a) partitioning the data into k groups, and compute the k group means or (b) an initial set of k points as the first estimate of the cluster means (seed points).
- Loop over all observations reassigning them to the group with the closest mean.
- Recompute group means.
- Iterate steps 2 and 3 until convergence.

Example

```
##     l    x1    x2
## 1   a  0.57 -1.08
## 2   b  1.53 -0.23
## 3   c  1.37 -0.50
## 4   d  1.53 -0.99
## 5   e -0.66  1.37
## 6   f -1.37  1.62
## 7   g -0.41  1.31
## 8   h -0.46  0.89
## 9   i -0.83 -0.63
## 10 j -0.12 -0.36
## 11 k -0.77 -0.54
## 12 l -0.37 -0.85
```

Example



Example

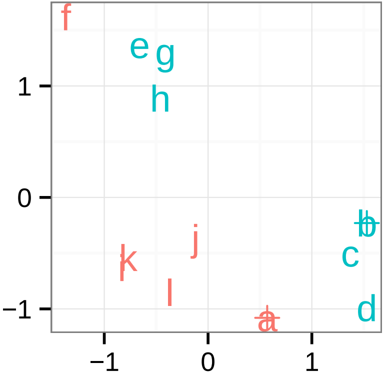
Select $k = 2$, and $\bar{x}_1 = 0.57, -1.08$, $\bar{x}_2 = 1.53, -0.23$

```
##    l    x1    x2   d1   d2
## 1  a  0.57 -1.08 0.00 1.81
## 2  b  1.53 -0.23 1.81 0.00
## 3  c  1.37 -0.50 1.38 0.43
## 4  d  1.53 -0.99 1.05 0.76
## 5  e -0.66  1.37 1.22 0.59
## 6  f -1.37  1.62 0.76 1.05
## 7  g -0.41  1.31 1.41 0.40
## 8  h -0.46  0.89 0.94 0.87
## 9  i -0.83 -0.63 0.95 2.76
## 10 j -0.12 -0.36 0.03 1.78
## 11 k -0.77 -0.54 0.80 2.61
## 12 l -0.37 -0.85 0.71 2.52
```

Example

```
##     l    x1    x2   d1   d2 cl
## 1  a  0.57 -1.08 0.00 1.81  1
## 2  b  1.53 -0.23 1.81 0.00  2
## 3  c  1.37 -0.50 1.38 0.43  2
## 4  d  1.53 -0.99 1.05 0.76  2
## 5  e -0.66  1.37 1.22 0.59  2
## 6  f -1.37  1.62 0.76 1.05  1
## 7  g -0.41  1.31 1.41 0.40  2
## 8  h -0.46  0.89 0.94 0.87  2
## 9  i -0.83 -0.63 0.95 2.76  1
## 10 j -0.12 -0.36 0.03 1.78  1
## 11 k -0.77 -0.54 0.80 2.61  1
## 12 l -0.37 -0.85 0.71 2.52  1
```

Example



Example

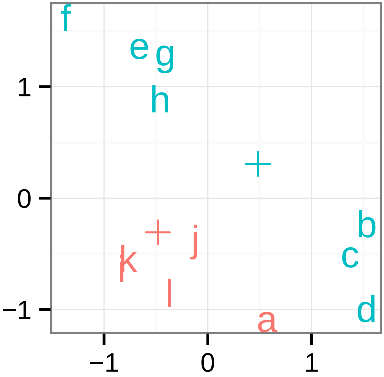
```
##      cl         x1         x2
## [1,]  1 -0.4816667 -0.3066667
## [2,]  2  0.4833333  0.3083333
```

```
##     l    x1    x2        d1         d2
## 1   a  0.57 -1.08 0.2783333 1.30166667
## 2   b  1.53 -0.23 2.0883333 0.50833333
## 3   c  1.37 -0.50 1.6583333 0.07833333
## 4   d  1.53 -0.99 1.3283333 0.25166667
## 5   e -0.66  1.37 1.4983333 0.08166667
## 6   f -1.37  1.62 1.0383333 0.54166667
## 7   g -0.41  1.31 1.6883333 0.10833333
## 8   h -0.46  0.89 1.2183333 0.36166667
## 9   i -0.83 -0.63 0.6716667 2.25166667
## 10  j -0.12 -0.36 0.3083333 1.27166667
## 11  k -0.77 -0.54 0.5216667 2.10166667
## 12  l -0.37 -0.85 0.4316667 2.01166667
```

Example

```
##     l    x1    x2        d1         d2 cl
## 1  a  0.57 -1.08 0.2783333 1.30166667  1
## 2  b  1.53 -0.23 2.0883333 0.50833333  2
## 3  c  1.37 -0.50 1.6583333 0.07833333  2
## 4  d  1.53 -0.99 1.3283333 0.25166667  2
## 5  e -0.66  1.37 1.4983333 0.08166667  2
## 6  f -1.37  1.62 1.0383333 0.54166667  2
## 7  g -0.41  1.31 1.6883333 0.10833333  2
## 8  h -0.46  0.89 1.2183333 0.36166667  2
## 9  i -0.83 -0.63 0.6716667 2.25166667  1
## 10 j -0.12 -0.36 0.3083333 1.27166667  1
## 11 k -0.77 -0.54 0.5216667 2.10166667  1
## 12 l -0.37 -0.85 0.4316667 2.01166667  1
```

Example



Example

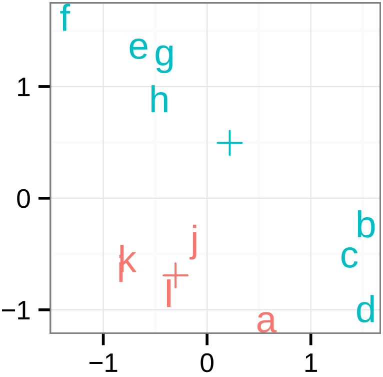
```
##      cl         x1         x2
## [1,]  1 -0.3040000 -0.6920000
## [2,]  2  0.2185714  0.4957143

##     l    x1    x2    d1          d2
## 1   a  0.57 -1.08 0.486 1.224285714
## 2   b  1.53 -0.23 2.296 0.585714286
## 3   c  1.37 -0.50 1.866 0.155714286
## 4   d  1.53 -0.99 1.536 0.174285714
## 5   e -0.66  1.37 1.706 0.004285714
## 6   f -1.37  1.62 1.246 0.464285714
## 7   g -0.41  1.31 1.896 0.185714286
## 8   h -0.46  0.89 1.426 0.284285714
## 9   i -0.83 -0.63 0.464 2.174285714
## 10 j -0.12 -0.36 0.516 1.194285714
## 11 k -0.77 -0.54 0.314 2.024285714
## 12 l -0.37 -0.85 0.224 1.934285714
```

Example

```
##    l    x1    x2    d1          d2 cl
## 1  a  0.57 -1.08 0.486 1.224285714  1
## 2  b  1.53 -0.23 2.296 0.585714286  2
## 3  c  1.37 -0.50 1.866 0.155714286  2
## 4  d  1.53 -0.99 1.536 0.174285714  2
## 5  e -0.66  1.37 1.706 0.004285714  2
## 6  f -1.37  1.62 1.246 0.464285714  2
## 7  g -0.41  1.31 1.896 0.185714286  2
## 8  h -0.46  0.89 1.426 0.284285714  2
## 9  i -0.83 -0.63 0.464 2.174285714  1
## 10 j -0.12 -0.36 0.516 1.194285714  1
## 11 k -0.77 -0.54 0.314 2.024285714  1
## 12 l -0.37 -0.85 0.224 1.934285714  1
```

Example



Example

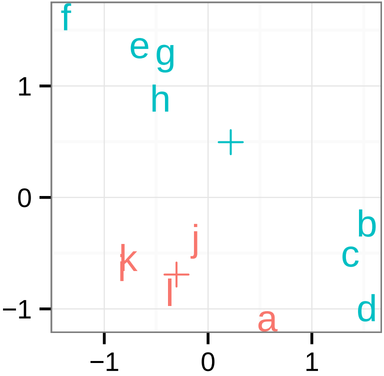
```
##      cl         x1         x2
## [1,]  1 -0.3040000 -0.6920000
## [2,]  2  0.2185714  0.4957143

##     l    x1    x2    d1          d2
## 1   a  0.57 -1.08 0.486 1.224285714
## 2   b  1.53 -0.23 2.296 0.585714286
## 3   c  1.37 -0.50 1.866 0.155714286
## 4   d  1.53 -0.99 1.536 0.174285714
## 5   e -0.66  1.37 1.706 0.004285714
## 6   f -1.37  1.62 1.246 0.464285714
## 7   g -0.41  1.31 1.896 0.185714286
## 8   h -0.46  0.89 1.426 0.284285714
## 9   i -0.83 -0.63 0.464 2.174285714
## 10  j -0.12 -0.36 0.516 1.194285714
## 11  k -0.77 -0.54 0.314 2.024285714
## 12  l -0.37 -0.85 0.224 1.934285714
```

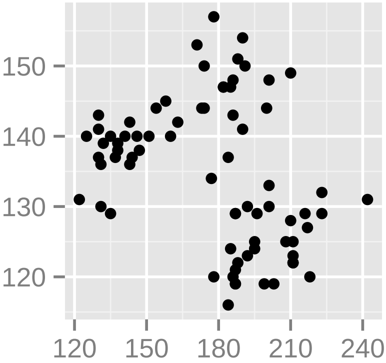
Example

```
##     l    x1    x2    d1          d2 cl
## 1   a  0.57 -1.08 0.486 1.224285714  1
## 2   b  1.53 -0.23 2.296 0.585714286  2
## 3   c  1.37 -0.50 1.866 0.155714286  2
## 4   d  1.53 -0.99 1.536 0.174285714  2
## 5   e -0.66  1.37 1.706 0.004285714  2
## 6   f -1.37  1.62 1.246 0.464285714  2
## 7   g -0.41  1.31 1.896 0.185714286  2
## 8   h -0.46  0.89 1.426 0.284285714  2
## 9   i -0.83 -0.63 0.464 2.174285714  1
## 10 j -0.12 -0.36 0.516 1.194285714  1
## 11 k -0.77 -0.54 0.314 2.024285714  1
## 12 l -0.37 -0.85 0.224 1.934285714  1
```

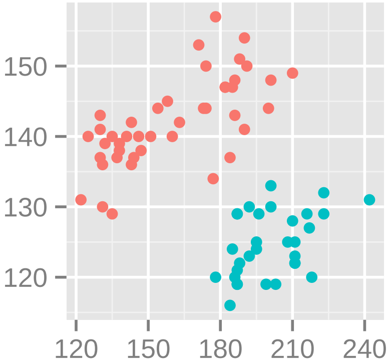
Example



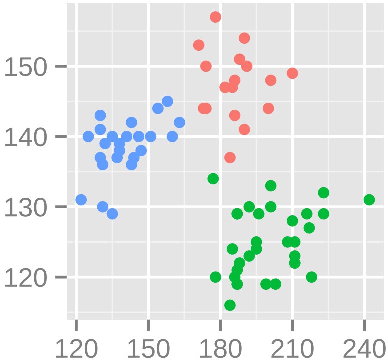
Cluster this



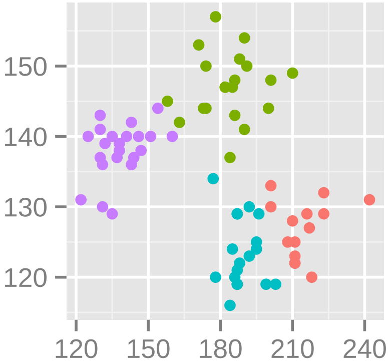
Two clusters



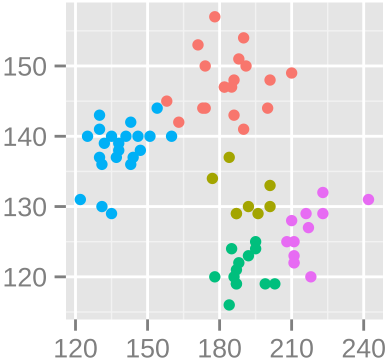
Three clusters



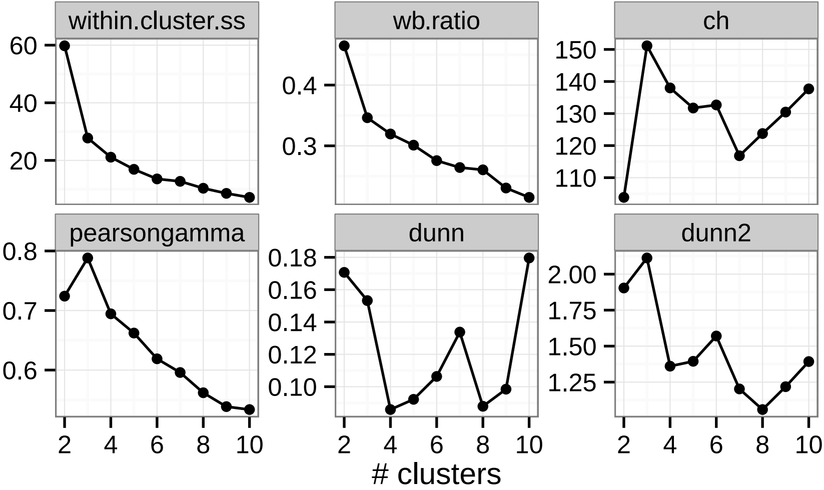
Four clusters



Five clusters



Compare results



Cluster stats

- WBRatio: average within/average between want it to be low, but always drops for each additional cluster so look for large drops
- Hubert Gamma: (s+ - s-)/(s+ + s-) where s+=sum of number of within < between, s- = sum of number within > between, want this to be high
- Dunn: smallest distance between points from different clusters/maximum distance of points within any cluster, want this to be high
- Calinski-Harabasz Index, $\frac{\sum_{i=1}^{p} B_{ii}/(k-1)}{\sum_{i=1}^{p} W_{ii}/(n-k)}$ want this to be high

Effect of seed

- The k-means algorithm can yield quite different results depending on the initial seed.
- We ran all the previous runs using 5 random starts, and using the within.cluster.ss metric to decide on the best solution

Distance metrics

- Cluster analysis depends on the interpoint distances, points close together should be grouped together
- Euclidean distance was used for the example. Let $A = (x_{a1}, x_{a2}, ..., x_{ap}), B = (x_{b1}, x_{b2}, ..., x_{bp})$

$$d_{EUC}(A, B) = \sqrt{\sum_{j=1}^{p}(x_{aj} - x_{bj})^2} = ((X_A - X_B)^T (X_A - X_B))$$

Other distance metrics

- Mahalanobis (or statistical) distance

$$d_{MAH}(A, B) = sqrt((X_A - X_B)^T S^{-1}(X_A - X_B))$$

- Manhattan:

$$d_{MAN}(A, B) = \sum_{j=1}^{p} |(X_{aj} - X_{bj})|$$

- Minkowski:

$$d_{MIN}(A, B) = (\sum_{j=1}^{p} |(X_{aj} - X_{bj})|^m)^{1/m}$$

Distances for count data

- Canberra:

$$d_{CAN}(A,B) = \frac{1}{n_z}\sum_{j=1}^{p}(X_{aj} - X_{bj})/(X_{aj} + X_{bj})$$

- Bray-Curtis:

$$d_{BRA}(A,B) = \sum_{j=1}^{p}|X_{aj} - X_{bj}|/\sum_{j=1}^{p}(X_{aj} + X_{bj})$$

Rules for metric to be a distance

1. $d(A, B) \geq 0$
2. $d(A, A) = 0$
3. $d(A, B) = d(B, A)$
4. Metric dissimilarity satisfies $d(A, B) \geq d(A, C) + d(C, B)$, and an ultrametric dissimilarity satisfies $d(A, B) \geq max\{d(A, C), d(C, B)\}$