Multivariate Analysis (slides 9)

- Today we consider $k$-means clustering.
- We will address the question of selecting the appropriate number of clusters.
- Properties and limitations of the algorithm will be explored.
$k$-means clustering

- The aim is to divide the data into $k$ distinct groups so that observations within a group are similar, whilst observations between groups are different.

- $k$-means clustering is an iterative, rather than a hierarchical, clustering algorithm.

- This means that at each stage of the algorithm data points will be assigned to a fixed number of clusters (contrast with hierarchical clustering where the number of clusters ranges from the number of data points down to a single cluster).

- We will discuss ways of selecting an appropriate $k$ from a statistical viewpoint, but there may be expert knowledge as to the appropriate number of clusters.

- Alternatively, there may be previous results from preliminary data exploration, i.e., we could start the $k$-means algorithm at the result of a hierarchical clustering.
$k$-means clustering

- It is simple and computationally efficient, but can sometimes be sensitive to the selection of starting points.
- Running the $k$-means algorithm several times for different starting values can help check whether results are robust.
- We will see an example of the problems this can cause.
Pseudo code

1. Choose the number of clusters $k$ and designate initial cluster centers
   \[ \mu_1, \ldots, \mu_k. \]

2. Assign each data point to the cluster whose center it is closest to: for each observation $i$, set
   \[
   r_{il} = \begin{cases} 
   1 & \text{if } l = \operatorname{arg\ min}_p \left\{ \sum_{j=1}^m (x_{ij} - \mu_{pj})^2 \right\}, \\
   0 & \text{otherwise}
   \end{cases}
   \]

3. For each cluster $p$, calculate the new centroid $\mu^T_p = (\mu_{p1}, \ldots, \mu_{pm})$, as a weighted average:
   \[
   \mu_{pj} = \frac{\sum_{i=1}^n x_{ij} r_{ip}}{\sum_{i=1}^n r_{ip}}.
   \]
Pseudo code

4. Calculate the **sum of squared distances of each object to its cluster centroid**:

\[
SS = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{p=1}^{k} (x_{ij} r_{ip} - \mu_{pj})^2
\]

Here we assume a total of \( n \) observations. We want the SS value to be as small as possible.

5. Re-assign each observation to the cluster whose centroid is closest.

6. Repeat (3)-(5) until convergence.
Simulated Data

- Consider the following simulated data.

- We want to cluster the data into three groups.
$k$-Means Clustering: Iteration 0a

- We start by randomly generating three centers (prototypes).

Iteration 0
The initial partition can be constructed in several ways, \textit{e.g.,}

1. A random selection of $k$ observations.

2. Specify selection based on prior knowledge.

3. By using results from an exploratory hierarchical clustering algorithm.
$k$-Means Clustering: Iteration 0b

- Label points according to which center is closest.
$k$-Means Clustering: Iteration 1a

- Update the values for the three centers (prototypes).
$k$-Means Clustering: Iteration 1b

- Label points according to which center is closest.
$k$-Means Clustering: Iteration 2a

- Update the values for the three centers (prototypes).
**k-Means Clustering: Iteration 2b**

- We label points according to which center is closest.
Convergence

- The $k$-Means algorithm has converged when no points are moved between groups on an iteration.

- Once this happens, the estimates of the centers will no longer change, nor will the allocation of points to groups thereafter.

- This convergence criteria might not be suitable in some cases, e.g., if $n$ is very large, and alternatives are possible, e.g., within cluster sum of squares changes by a very small amount over successive iterations.

- See https://www.naftaliharris.com/blog/visualizing-k-means-clustering for some nice interactive visualisations of the method.
Didn’t It Do Well?

- The data from the last example had been simulated, so that there were actually three groups in the data.
- How well did $k$-means perform at finding these groups?
Any errors?

- The coloured in points were misclassified.
- Only 8/300 were misclassified.
Choosing $k$

- This is not an exact science, but there are guidelines.
- Generally we run the $k$-means algorithm for a differing number of values for $k$, e.g., $k = 1, \ldots, 10$.
- When running $k$-means the aim is to minimize the SS, so why not choose $k$ to minimize the $SS$?
- However, the more clusters that are fitted, the smaller the $SS$ (think of what would happen if we selected $k = n$).
- A general rule is to plot $k$ against $SS$ and look for a ‘kink’ in the curve. If there is no kink then there is a trade-off between additional complexity by increasing $k$ and better fit by reducing the $SS$. 
Choosing $k$

- If we plot the total of the within sum of squares values versus $k$, then we get the following:

- Notice that the graph flattens very quickly. What $k$ would you use?
Old Faithful Data

- Running $k$-means on the standardized Old Faithful data allows a plot of the within sum of squares values versus $k$:

- What value of $k$ looks good?
Old Faithful Data

- This provides the following clustering of the data:
Consider the $k=2$ solution for the Faithful data:

Number of clusters: 2

<table>
<thead>
<tr>
<th></th>
<th>Number of Obs.</th>
<th>WSS</th>
<th>Avg. Dist. to Centroid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 1</td>
<td>100</td>
<td>3456.2</td>
<td>4.9</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>172</td>
<td>5445.6</td>
<td>4.6</td>
</tr>
<tr>
<td>Sum</td>
<td>272</td>
<td>8901.8</td>
<td></td>
</tr>
</tbody>
</table>

Cluster Centroids:

<table>
<thead>
<tr>
<th></th>
<th>Cluster 1</th>
<th>Cluster 2</th>
<th>Total Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>eruptions</td>
<td>2.1</td>
<td>4.3</td>
<td>3.5</td>
</tr>
<tr>
<td>waiting</td>
<td>54.8</td>
<td>80.3</td>
<td>70.9</td>
</tr>
</tbody>
</table>

Distance Between Cluster Centroids:

<table>
<thead>
<tr>
<th></th>
<th>Cluster 1</th>
<th>Cluster 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster 1</td>
<td>0.0</td>
<td>25.6</td>
</tr>
<tr>
<td>Cluster 2</td>
<td>25.6</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Olive Oil Data

- Running $k$-means on the standardized olive oil data allows a plot of the within sum of squares values versus $k$:

- What value of $k$ looks good? Let’s look at $k = 9$. 
Olive Oil Data

- This provides the following clustering of the data:
Cross Tabulation

- A cross tabulation of the olive oil regions (rows) and the clusters (columns) shows some agreement:

```
4  9  6  7  1  8  3  5  2
1 22  2  0  0  0  0  0  0  1
2  0 32  0 23  0  0  1  0  0
3  0 12 144 1  0 49  0  0  0
4  6 16  0 12  0  2  0  0  0
5  0  0  0  0 65  0  0  0  0
6  0  0  0  0 33  0  0  0  0
7  0  0  0  0  0 33 10  7
8  0  0  0  0  0 50  0
9  0  0  0  0  0  1  0 50
```
Silhouette Width

- Another way to assess cluster performance is to compute the silhouette width for observations \( x = x_1, \ldots, x_n \).

- For each \( i = 1, \ldots, n \), compute:

\[
a(i) = \frac{1}{n_l - 1} \sum_{j=1}^{n} r_{jl} d(x_i, x_j),
\]

where \( i \in \text{Cluster } l \), i.e., compute the average distance of \( x_i \) from the other data points in its cluster. (We set \( a(i) = 0 \) if \( x_i \) is in a singleton cluster.)

- Also compute

\[
b(i) = \min_{p \neq l} \frac{1}{n_p} \sum_{j=1}^{n} r_{jp} d(x_i, x_j),
\]

i.e., the average distance of \( x_i \) from the data points in the nearest cluster that it is not itself assigned to.
Silhouette Width

- Then compute
  \[ s_i = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \]
- We call \( s_i \) the **silhouette width** of observation \( x_i \).
- Note that \(-1 \leq s_i \leq 1\). (Why?)
- A large value of \( s_i \) indicates that \( b(i) \) is much larger than \( a(i) \), i.e. that \( x_i \) is much closer to the data points in its own cluster than points in its most neighbouring cluster.
- Hence, high silhouette width indicates good cluster separation.
- Values close to zero suggest that clusters are poorly separated or overlapping.
- Negative values indicate that a data point has probably been assigned to the wrong cluster.
Silhouette Width

- Here is a Silhouette plot for the faithful data, with \( k = 2 \).
- What does this tell us?
- Note that we can also compute the mean silhouette width for each cluster, and for the data overall.
Silhouette Width

The Silhouette plot for the olive oil data with $k = 9$ shows the average silhouette width for each cluster. The average silhouette width is 0.58. The plot indicates how well each sample fits into its own cluster compared to other clusters.

What does the Silhouette plot for the olive oil data with $k = 9$ tell us?
Tricky Data

• What if we run $k$-means on the following, more tricky, standardized data:

-15 −10 −5 0 5 10 15
−6 −2 0 2 4 6

• It looks like there should be two groups.
• Plotting the within sum of squares values versus $k$ gives:

• What value of $k$ looks good?
Tricky Data

- This provides the following clustering of the data:

- The elliptical group is broken into subgroups. This is because $k$-means clustering looks for circular clusters.
Distance From Means

- Return to the first data set.

- Consider the line separating the points that are closest to the mean of the triangles (△) and the points closest to the mean of the circles (○).
Distance From Means

- Including such lines for each pair of means:
The plane is partitioned into three polygonal regions depending on which mean is closest.
• Recall the tricky data from earlier.

• If we cluster it into two groups, we get:
• But if we ran $k$-means from a different and specific starting point:
The \( k \)-means algorithm can give different answers when initiated at different starting values.

This means that the algorithm does not always find the minimum value for the Total Within Sum of Squares.

The Total Within Sum of Squares for the first clustering is \( 82.4 + 36.3 = 118.7 \).

The Total Within Sum of Squares for the second clustering is \( 11.0 + 98.1 = 109.1 \).

Therefore, the second set of results is better.