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 cluster centers.

2. Assign each data point to the cluster whose center is closest.

3. For cluster $i$, calculate its centroid $C_i^T = (C(i)_1, C(i)_2, \ldots, C(i)_m)$, where $m$ denotes the number of variables in an observation (these are found by averaging variables scores for data points within the cluster).

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

\[
SS = \sum_{i=1}^{n} \sum_{j=1}^{m} (x_{ij} - C(i)_j)^2
\]

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, say, three groups.
$k$-Means Clustering: Iteration 0a

- We start by randomly generating three centers (prototypes).
$k$-Means Clustering: Iteration 0a

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.

![Iteration 0 Diagram]
$k$-Means Clustering: Iteration 1

- 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 does not change over 3 iterations etc.
How Did It Do?

- 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 The Value For $k$

- This is not an exact science, but there are guidelines.
- Generally we should 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 The Value For $k$

- 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 $k$:

![Plot of total within sum of squares versus k]

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

- This provides the following clustering of the data:
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:
A cross tabulation of the olive oil regions (rows) and the clusters (columns) shows some agreement:

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>9</th>
<th>6</th>
<th>7</th>
<th>1</th>
<th>8</th>
<th>3</th>
<th>5</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>22</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>32</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>12</td>
<td>144</td>
<td>1</td>
<td>0</td>
<td>49</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>16</td>
<td>0</td>
<td>12</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>65</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>33</td>
<td>10</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>50</td>
<td></td>
</tr>
</tbody>
</table>
Faithful Data $k=2$

- Consider the $k = 2$ solution for the Faithful data:

  Number of clusters: 2

<table>
<thead>
<tr>
<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>
</tr>
<tr>
<td>Cluster 2</td>
<td>172</td>
<td>5445.6</td>
</tr>
<tr>
<td>Sum</td>
<td>272</td>
<td>8901.8</td>
</tr>
</tbody>
</table>

  Cluster Centroids:

<table>
<thead>
<tr>
<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>
</tr>
<tr>
<td>waiting</td>
<td>54.8</td>
<td>80.3</td>
</tr>
</tbody>
</table>

  Distance Between Cluster Centroids:

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

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

\[
\begin{array}{ccccccc}
-15 & -10 & -5 & 0 & 5 & 10 & 15 \\
-6 & -2 & 0 & 2 & 4 & 6 & \\
\end{array}
\]

- It looks like there should be two groups.
Tricky Data

- 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:
Partitioning Of The Plane

- 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:
Tricky Data

- But if we ran \( k \)-means from a different and specific starting point:
Local Minima

• 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.