MONASH University

MONASH BUSINESS SCHOOL

Multidimensional Scaling

High Dimensional Data Analysis

Anastasios Panagiotelis

Lecture 5

Motivation

Motivation

- Last week we looked at the concept of *distance* between observations.
- We looked at our usual understanding of distance known as *Euclidean* distance.
- We also looked at higher dimensional versions of Euclidean distance.
- Other distance metrics including *Jaccard* distance can be used for categorical data.

Can we see distance?

- Suppose we have n observations and the distance between each possible pair of observations.
- A scatterplot shows whether observations are close together or far apart.
- This works nicely when there are 2 variables.
- Can we see higher dimensional distances or non Euclidean distances in 2-dimensions?
- Unfortunately the answer is no...
- ... but we can get a good approximation

Multidimensional Scaling

- Multidimensional scaling (MDS) finds a low (usually 2) dimensional representation.
- The pairwise 2D Euclidean distances in this representation should be as *close* as possible to the original distances.
- The meaning of *close* can vary since there are different ways to do MDS.
- However MDS always begins with a matrix of distances and ends with a low dimensional representation that can be plotted.

An optical illusion with Beyonce

Why does the illusion work?

- Difference between the 3D distance of reality, and the 2D distance in the photo.
- In reality the distance between Beyonce's hand and the Eiffel Tower is large.
- In the 2D photo, this distance is small.
- This is a misleading representation to understand the distance between Beyonce's hand and the Eiffel Tower.
- A much more informative representation could be found by *rotation*.

Why do we care?

- An important issue in business is to profile the market. For example
  - Which products do customers perceive to be similar to one another?
  - Who is my closest competitor?
  - Are there ‘gaps’ in the market, where a new product can be introduced?
- Multidimensional Scaling can help us to produce a simple visualisation that can address these questions.

Beer Example

Beer Example

- The plot on the previous slide is an MDS solution for the beer dataset.
- The data are 5-dimensional so we cannot use a scatter plot.
- MDS shows Olympia Gold Light and Pabst Extra Light are similar (both light beers).
- This also suggest that there is a low number of competitors with St Pauli Girl.
- This may also reflect that the attributes of St Pauli Girl are not desired by customers.
- How did we get the plot?

Beer Data

rating	beer	origin	avail	price	cost	cal
Fair	Olympia Gold Light	USA	Regional	2.75	0.46	
Fair	Pabst Extra Light	USA	National	2.29	0.38	
Fair	Schlitz Light	USA	National	2.79	0.47	

Details

- To keep the example simple only the beers rated *fair* are used
- In general, all the beers can be used.
- Also to keep things simple we only consider the metric variables so that we can use Euclidean distance.
- In general, we can use distance metrics that work for categorical data.

Metric Variables

- After standardising, Euclidean distances are formed between every possible pair of beers.
- For example, the distance between Blatz and Tuborg is given by

$$\delta\left(\mathrm{Blatz}, \mathrm{Tbrg}\right) = \sqrt{\sum_{h=1}^{5}\left(\mathrm{Blatz}_h - \mathrm{Tbrg}_h\right)^2}$$

Both the notation δ_{ij} and $\delta(i,j)$ will be used interchangeably

Doing it in R

To obtain the distance matrix in R

```
filter(Beer,rating=='Fair')%>% #Only fair
  select_if(is.numeric)%>% #Only metric c
  scale%>% #Standardise
  dist->delta #Distance

filter(Beer,rating=='Fair')%>% #Only fair
  use_series(beer)%>% #Get beer names
  abbreviate(6)-> #Abbreviate
  attributes(delta)$Labels #Assign to d
```

MDS in R

We can do what is known as **classical** MDS in R using the `cmdscale` function

```
mdsout<-cmdscale(delta)
```

OlymGL	-1.9758212	-1.6276821
PbstEL	-2.1860282	-1.1600914
SchltL	-0.7420968	-0.7994497
[illegible]	[illegible]	[illegible]

Two new variables

- We have just created two new variables for visualising the distances.
- The distances that we visualise will be 2-dimensional distances. For example

$$d(\mathrm{Blatz}, \mathrm{Tbrg}) = \sqrt{(-0.339 - 0.808)^2 + (1.493 - 0.703)^2}$$

Not exact

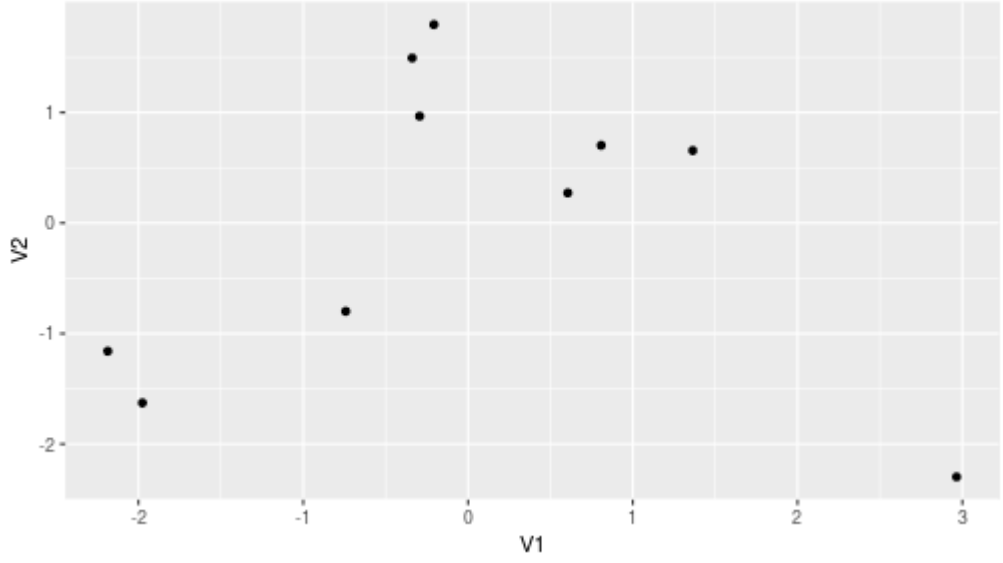
- In this example $d(\mathrm{Blatz}, \mathrm{Tuborg}) = 1.3927$ while $\delta(\mathrm{Blatz}, \mathrm{Tuborg}) = 1.4762$. Notice that

$$d(\mathrm{Blatz}, \mathrm{Tuborg}) \neq \delta(\mathrm{Blatz}, \mathrm{Tuborg})$$

- But they are close.

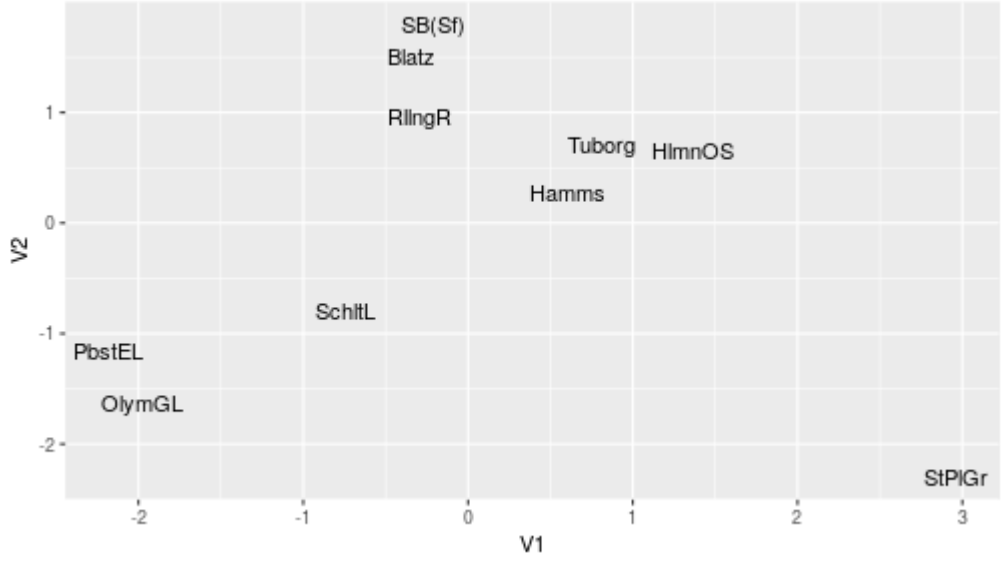
Getting the plot

```
mdsout%>%
  as_data_frame%>%
  ggplot(aes(x=V1,y=V2))+geom_point()
```



Getting the plot with names

```
mdsout%>%
   as_data_frame(rownames='BeerName')%>%
   ggplot(aes(x=V1,y=V2,label=BeerName))+g
```



The math behind classical MDS

- In *classical* MDS the objective is to minimise strain

$$\text{Strain} = \sum_{i=1}^{n-1} \sum_{j>i} (\delta_{ij}^2 - d_{ij}^2)$$

- This is a similar procedure (and can even be equivalent) to PCA.

MDS and PCA

- There are some similarities with PCA
  - We represent information in a small number of dimensions
  - This representation involves some sort of rotation in the high dimensional space.
- In the case where the δ are Euclidean distances MDS will be equivalent to PCA.
- For this reason, the coordinates that come out of classical MDS are often called **principal coordinates**.

MDS and PCA

- This also means that when we look at principal components we know two things
  - The principal components maximise the explained variance
  - The principal components accurately represent the distances between our observations!

How good is this representation?

- To evaluate the quality of a MDS representation there are two goodness of fit measures.
- As a rule of thumb, values greater than 0.8 are desirable.
- These can be obtained using the option `eig=TRUE` in the `cmdscale` function

```
## [1] 0.8543224 0.8543224
```

Is the solution any good?

- These values depend on the eigenvalues

$$\mathrm{GF}_1 = \frac{\sum_{i=1}^{2} |\lambda_i|}{\sum_{i=1}^{n} |\lambda_i|} \quad \mathrm{GF}_2 = \frac{\sum_{i=1}^{2} max(0, \lambda_i)}{\sum_{i=1}^{n} max(0, \lambda_i)}$$

- For Euclidean distances δ_{ij} eigenvalues are always positive and $\mathrm{GF}_1 = \mathrm{GF}_2$.

Non-Euclidean distances

- In practice non-Euclidean distances can also be used in classical multidimensional scaling.
- However, using non-Euclidean distances can lead to negative eigenvalues. In this case:
  - Classical MDS may not minimise Strain.
  - Two fit measures will differ.
- MDS is not equivalent to PCA when δ is non-Euclidean.
- Overall, we can use classical MDS for non-Euclidean distance but must be more careful.

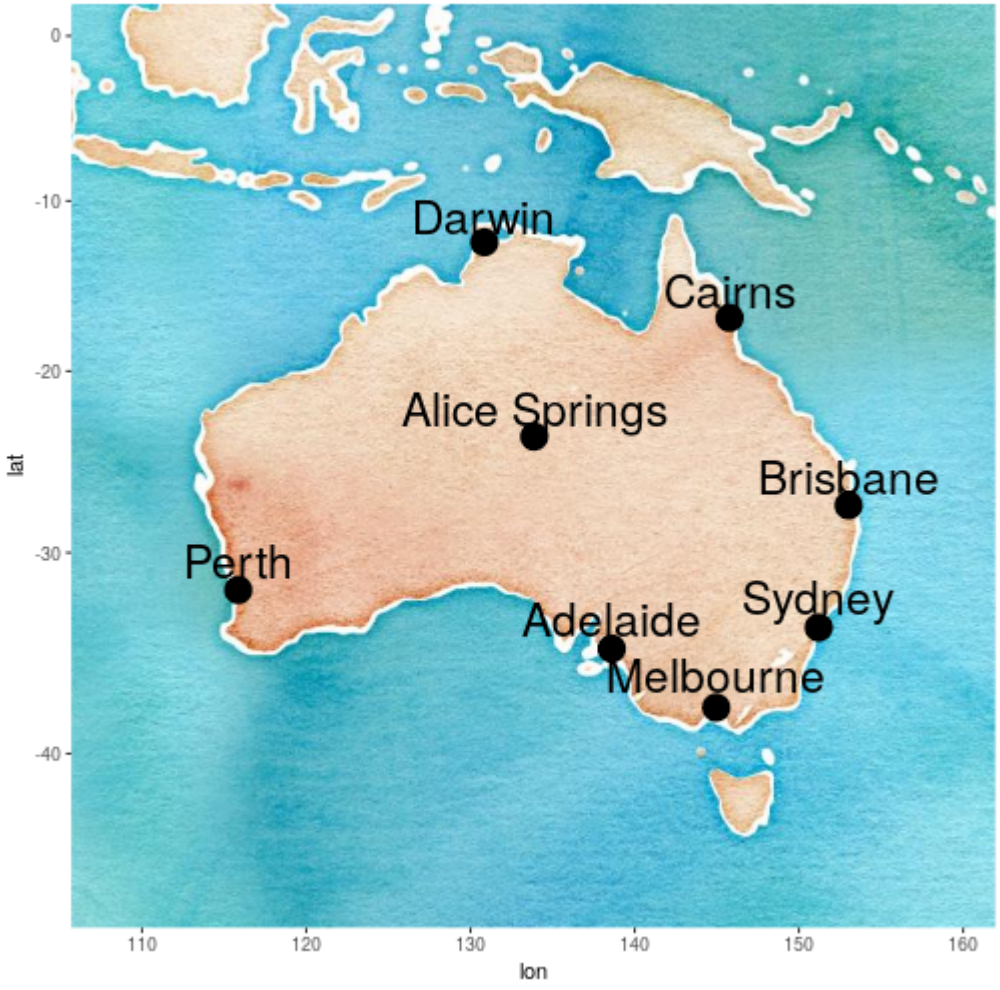
An example: Road distances

- Suppose that we have the road distances between different cities in Australia.
- We want to create a 2-dimensional map with the locations of the cities using only these road distances. However
  - The earth is 3D
  - The roads are not straight lines, in particular some costal roads may be very wiggly. Road distances are non-Euclidean.
- However MDS gives an approximation that is quite close to a real map.

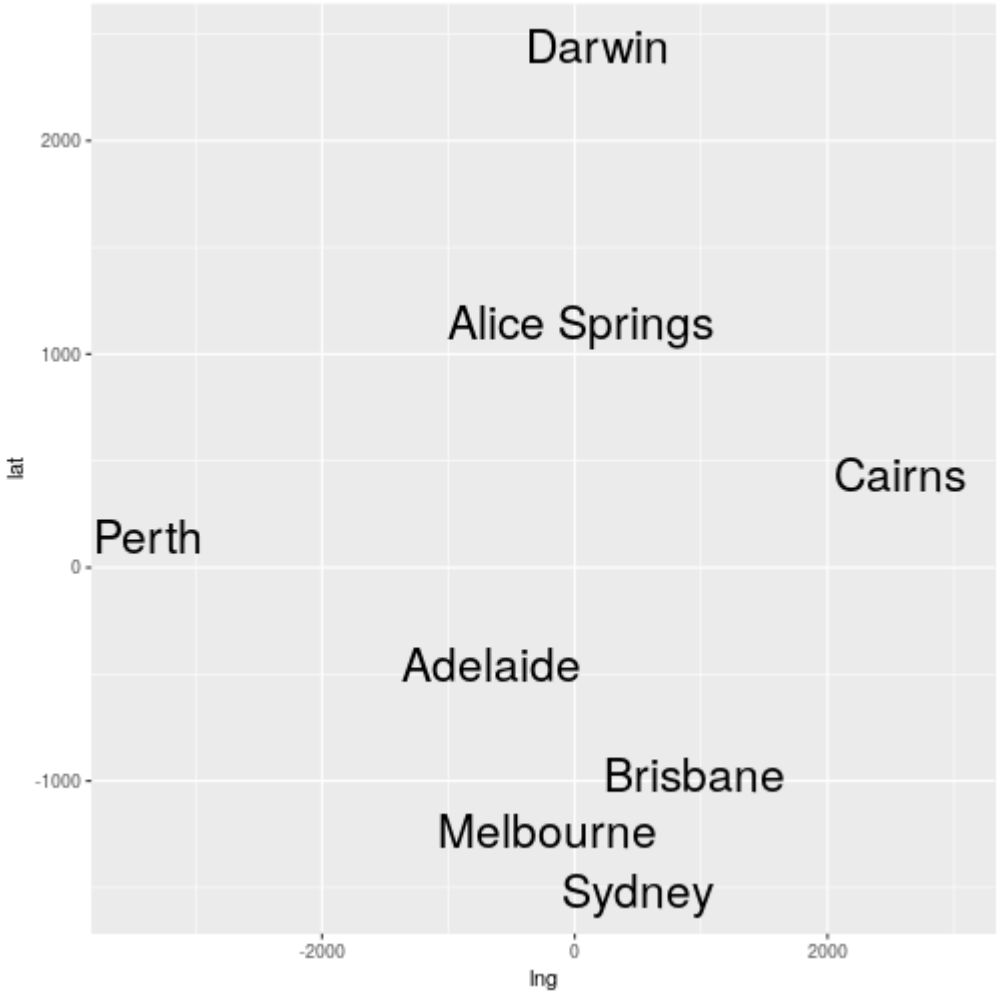
Road Distances

	Cairns	Brisbane	Sydney	Melbourne	A
Cairns	0	1717	2546	3054	
Brisbane	1717	0	996	1674	
Sydney	2546	996	0	868	
Melbourne	3054	1674	868	0	
Adelaide	3143	2063	1420	728	
Perth	5954	4348	4144	3452	

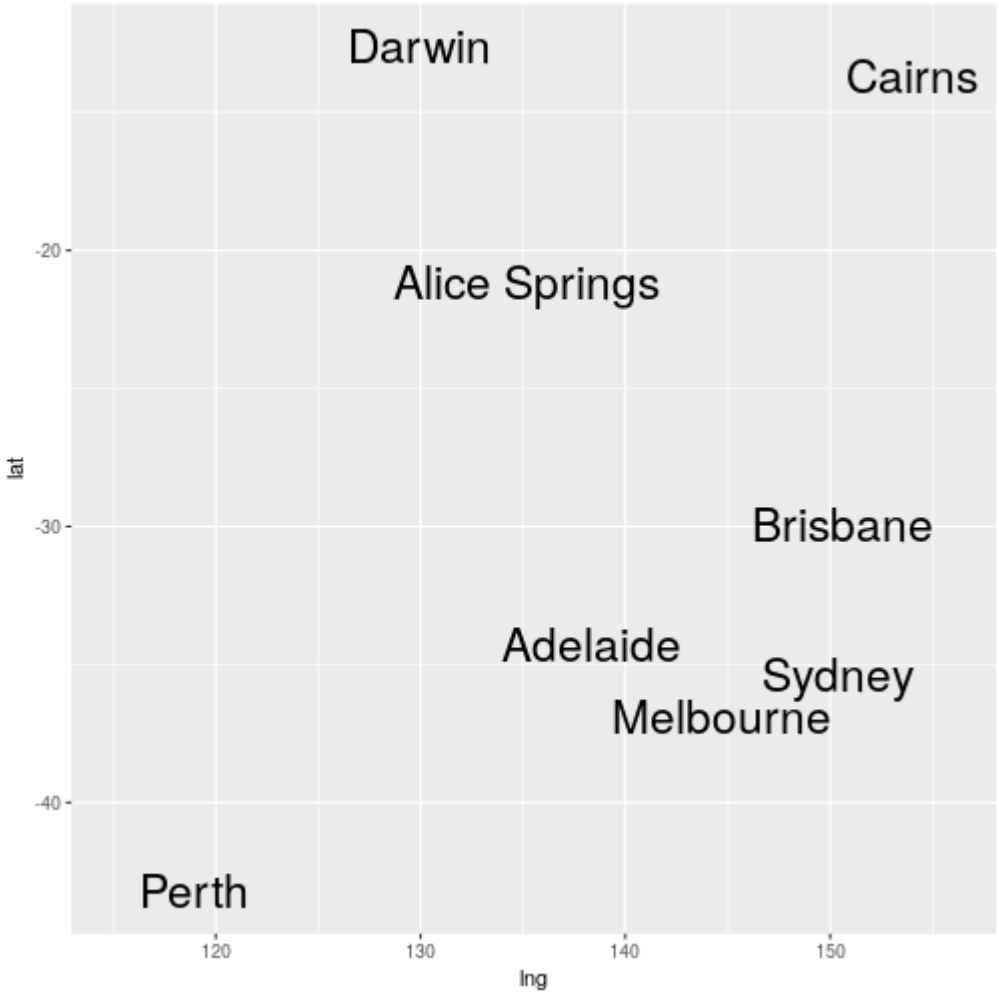
Australia


Darwin
Cairns
Alice Springs
Brisbane
Perth
Adelaide
Sydney
Melbourne
0
-10
-20
-30
-40
lat
110
120
130
140
150
160
lon

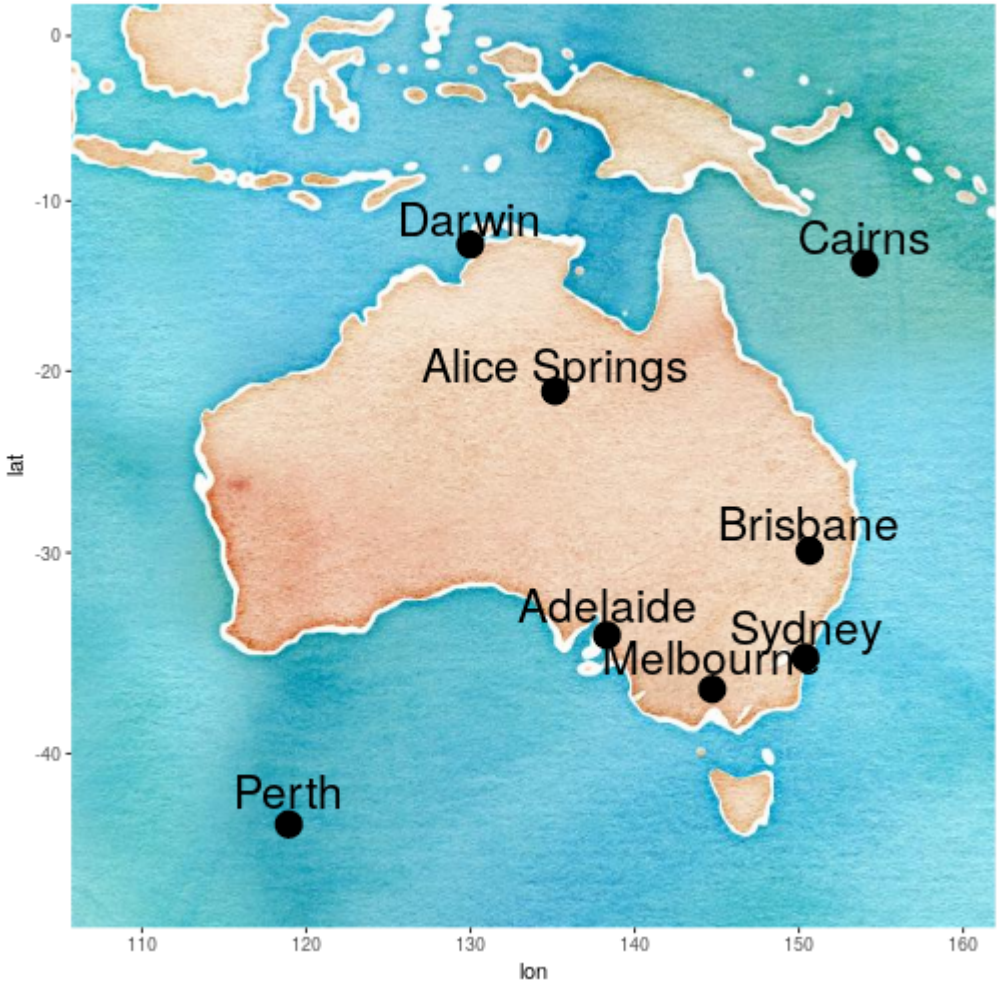
MDS Solution



Rotate


Darwin
Cairns
Alice Springs
Brisbane
Adelaide
Sydney
Melbourne
Perth
-20
-30
-40
lat
120
130
140
150
lng

Back with Map



Rotating

- If we rotate the points then the distances between the points do not change.
- Once we have an MDS solution any 2D rotated solution will be equally good.
- Sometimes these rotations help us to interpret the axes.
- In the previous example I rotated so the the x-axis represented East West direction and the y-axis represented North South.

Evaluating this result

```
cmdscale(doz,eig=TRUE)->dozout
print(dozout$eig)
```

```
## [1]  1.970494e+07  1.248534e+07  2.622827e
## [6] -3.117856e+05 -1.083294e+06 -2.179888e
```

```
print(dozout$GOF)
```

```
## [1] 0.8372483 0.9230786
```

Evaluating the Result

- There are negative eigenvalues.
  - This occurs since road distances are not Euclidean
  - This also implies that classical MDS may not minimise strain.
- Both goodness of fit measures are quite high.
  - The solution is an accurate representation.

Another example: Cheese

On moodle you will find the paper ‘Multidimensional Scaling of Sorting Data Applied to Cheese Perception’, Food Quality and Preference,6, pp.91-98. The purpose of this study was to visualise the difference between types of cheese.

Another example: Cheese

- The motivation is to investigate the similarities and differences between types of cheese.
- In principle one could measure attributes of the cheese.
- However the purpose of this study was to ask customers about their perceptions.
- How do we ask customers about distances?
- Could you walk out on to the street and ask someone about the Euclidean distance between Brie and Camembert?

Constructing the Survey

- Customers can be asked:
- On a scale of 1 to 10 with 1 being the most similar and 10 being the most different, how similar are the following cheeses
  - Brie and Camembert
  - Brie and Roquefort
  - Camembert and Roquefort
- The dissimilarity scores can be averaged over all customers and used in an MDS
- This is not a good method when there is a large number of products.

A more feasible approach

- In the study there are 16 cheeses therefore 120 possible pairwise comparisons.
- It is not practical to ask survey participants to make 120 comparisons!
- Instead of being asked to make so many comparisons, customers were asked to put similar cheeses into groups.
- Proportion of customers with two cheeses in same group is a similarity score.
- Proportion of customers with two cheeses in different groups is a dissimilarity score.

Consider four customers

- Suppose there are four customers sorting cheeses
  - Customer A: Brie and Camembert together, Roquefort and Blue Vein together
  - Customer B: Roquefort and Blue Vein together, all others separate
  - Customer C: All cheeses in their own category
  - Customer D: All cheeses in one category

Comparisons

- Customer A and D have Brie and Camembert in the same group, customers B and C have them in different groups.
  - The distance between Brie and Camembert is 0.5.
- Customer A, B and D have Roquefort and Blue Vein in the same group, customer C has them in different groups.
  - The distance between Roquefort and Blue Vein is 0.25.

Beyond Classical MDS

- If you continue to read the paper you will notice that there are references to some new terms.
  - SYSTAT (a software program)
  - Non-metric MDS
  - Kruskal's algorithm
- So far we have only talked about one type of MDS, namely classical MDS.
- There are many alternatives.

Beyond Classical MDS

- Classical MDS is designed to minimise Strain.
- An alternative objective function called Stress can be minimised instead

$$\text{Stress} = \sum_{i=1}^{n-1} \sum_{j>i} \frac{(\delta_{ij} - d_{ij})^2}{\delta_{ij}}$$

- This penalises errors of observations that are close to one another more.
- Implemented in the `sammon` function in the R package `MASS`.

Non-metric MDS

- In some cases, the distance themselves are not metric but ordinal.
- Suppose we only know

$$\delta_{\text{Bri.,Cam.}} < \delta_{\text{Roq.,Cam.}} < \delta_{\text{Roq.,Bri.}}$$

- Brie and Roquefort are *more different* compared to Brie and Camembert.
- We do not know *how big* the distance between Brie and Roquefort is compared to the distance between Brie and Camembert.

Non-metric MDS

- In this case we solve a different optimisation problem.

$$\text{Stress} = \sum_{i=1}^{n-1} \sum_{j>i} \frac{(\hat{\delta}_{ij}^2 - d_{ij}^2)}{\hat{\delta}_{ij}}$$

- Subject to constraints, e.g.

$$\hat{\delta}_{\text{Bri.,Cam.}} < \hat{\delta}_{\text{Roq.,Cam.}} < \hat{\delta}_{\text{Roq.,Bri.}}$$

Non-metric MDS

- Our solution requires us to find $\hat{\delta}$'s and d's
- This is a very useful algorithm for marketing since survey participants cannot easily and reliable assign numbers to the difference between products.
- The `isoMDS` function in the R package `MASS` is one implementation of non-metric MDS.

Summary

- Go from a general distance matrix to a 2D representation that preserves distances as best as possible
- Focus on classical MDS with Euclidean distances
  - Minimises strain
  - Is equivalent to PCA
  - Can work well non-Euclidean distances
- Be aware of other approaches
  - Sammon mapping
  - Non-metric MDS