Multidimensional Scaling

UBCO MDS — DATA 573

Multidimensional Scaling

- (Classical, metric) multidimensional scaling takes pairwise distances and attempts to recreate the original data (sort of)
- Let X be a matrix of n observations on p continuous predictors. For ease, assume the predictors are centred at 0 (mean subtracted), or scaled.
- We are used to taking

$$\mathrm{X} \rightarrow \text{euclidean distance computations} \rightarrow \mathrm{D}$$

where D is an $n \times n$ symmetric matrix of pairwise distances

Multidimensional Scaling

- Now, our goal with MDS instead is to reverse the process to find

$$\mathsf{D} \rightarrow ??? \rightarrow \mathsf{X}$$

- Since pairwise euclidean distances are not one-to-one functions, this is not a straightforward inversion.
- First, let's return to the original, known direction

$$\mathsf{X} \rightarrow \text{euclidean distance computations} \rightarrow \mathsf{D}$$

  we can add the following intermediate step

$$\mathsf{X} \rightarrow \mathsf{X}\mathsf{X}^T \rightarrow \mathsf{D}$$

Multidimensional Scaling

- For ease of notation, this can be summarized as

$$X \to XX^T = Q \to D$$

- Let's walk through the process on the board for

$$X \to Q \to D$$

Multidimensional Scaling

- Okay, now it's time to figure out the first of the reverse steps

$$\mathrm{D} \rightarrow \mathrm{Q}$$

- It can be shown

$$q_{rs} = -\frac{1}{2}(d_{rs}^2 - \frac{1}{n}\sum_r d_{rs}^2 - \frac{1}{n}\sum_s d_{rs}^2 + \frac{1}{n^2}\sum_s\sum_r d_{rs}^2)$$

- But what about

$$\mathrm{Q} \rightarrow \mathrm{X}$$

Multidimensional Scaling

- Well, Q is $n \times n$ and symmetric. We would like to map it to X which is $n \times p$ non-symmetric.
- Of course, in any realistic situation where X is unknown, its unlikely that we know what p is either.
- So this is not a trivial problem. In fact, it is unsolvable...

Multidimensional Scaling

- BUT, consider our previous discussions on dimensionality reduction.
- We can decompose Q using the spectral decomposition, say

$$\mathrm{Q} = \mathrm{P}\Lambda\mathrm{P}^T$$

- We are looking for something of the form XX^T, so note

$$\mathrm{P}\Lambda\mathrm{P}^T = \mathrm{P}\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}\mathrm{P}^T = (\mathrm{P}\Lambda^{\frac{1}{2}})(\mathrm{P}\Lambda^{\frac{1}{2}})^T$$

which suggests that $\mathrm{Y} = \mathrm{P}\Lambda^{\frac{1}{2}}$ provides points satisfying distances from D

Multidimensional Scaling

- However, even though $Q = XX^T = YY^T$, in fact $X \neq Y$
- Y as we've developed it would be $n \times n$, though the n^{th} eigenvalue (and possibly earlier ones) would be 0, so really $n \times (n-1)$ at most.
- MDS is usually used to find simple representations of pairwise distances. So fairly common choices are to take the first 2 or 3 columns of Y and ignore the rest.
- If this sounds a bit like PCA, it should...with a little more effort, one can show that the j^{th} column of Y is equivalent to the scores on the j^{th} component from a PCA on X (under some assumptions)

Okanagan Valley Example

```
> library(gmapsdistance) #will autocompute distances from google maps for us
> lcity <- c("Kelowna+BC", "Peachland+BC", "Vernon+BC", "West+Kelowna+BC",
             "Vancouver+BC", "Lake+Country+BC", "UBC+Okanagan",
             "Penticton+BC", "Kamloops+BC")
> matrix(lcity, ncol=1)
      [,1]
 [1,] "Kelowna+BC"
 [2,] "Peachland+BC"
 [3,] "Vernon+BC"
 [4,] "West+Kelowna+BC"
 [5,] "Vancouver+BC"
 [6,] "Lake+Country+BC"
 [7,] "UBC+Okanagan"
 [8,] "Penticton+BC"
 [9,] "Kamloops+BC"
```

Okanagan Valley Example

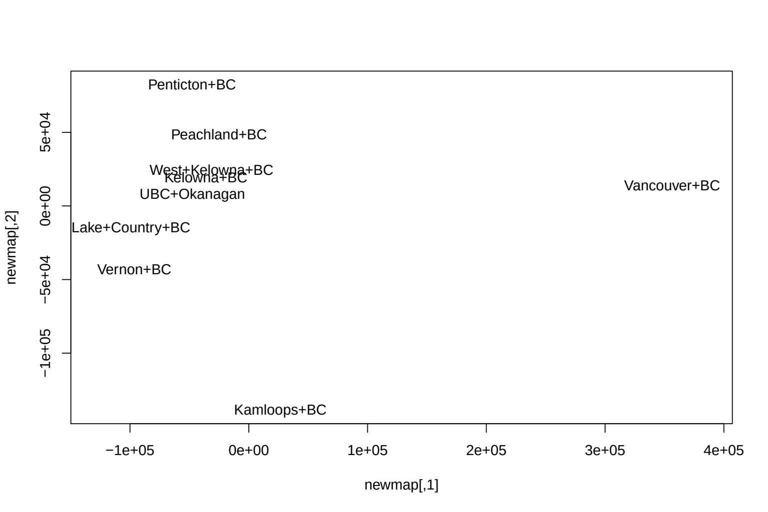
```
> pdis <- gmapsdistance(lcity, lcity, combinations="all", mode="driving")
> dismat <- pdis$Distance[,-1]
> colnames(dismat) <- rownames(dismat) <- lcity
> dismat[1:4, 1:4]
                Kelowna+BC Peachland+BC Vernon+BC West+Kelowna+BC
Kelowna+BC               0        25046     50550            7816
Peachland+BC         25132            0     75616           17975
Vernon+BC            50374        75345         0           58115
West+Kelowna+BC       7157        22045     57641               0
```

Okanagan Valley Example

```
> avgs <- (dismat[lower.tri(dismat)] + t(dismat)[lower.tri(dismat)] )/ 2
> dismat[lower.tri(dismat)] <- avgs
> dismat <- t(dismat)
> dismat[lower.tri(dismat)] <- avgs
> dismat[1:4, 1:4]
                Kelowna+BC Peachland+BC Vernon+BC West+Kelowna+BC
Kelowna+BC             0.0      25089.0   50462.0          7486.5
Peachland+BC       25089.0          0.0   75480.5         20010.0
Vernon+BC          50462.0      75480.5       0.0         57878.0
West+Kelowna+BC     7486.5      20010.0   57878.0             0.0
```

Okanagan Valley Example

```
> plot(newmap, type="n", xlim=c(min(newmap[,1])-30000, max(newmap[,1])+30000 ))
> text(newmap, labels=lcity)
```

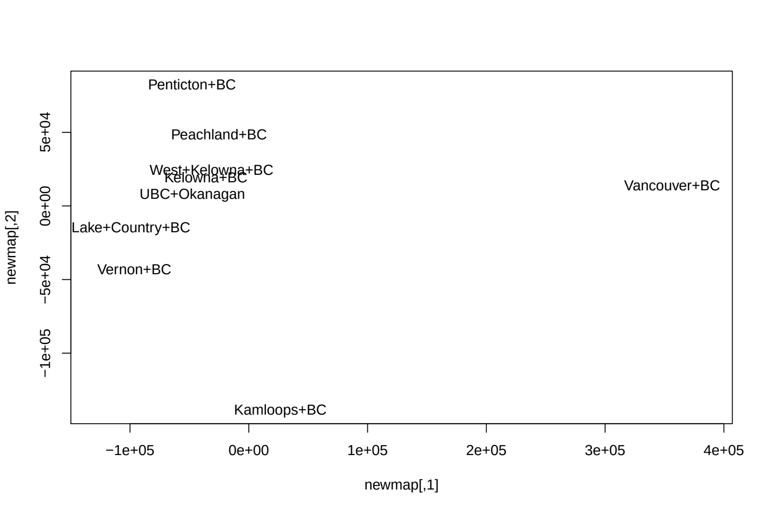


Okanagan Valley Example

- Recall with PCA, we could reflect (multiply by -1) on any component as the total sign is arbitrary.
- Same thing here. MDS doesn't "know" north/south/east/west in the same way that PCA didn't "know" how we'd prefer heptathlon scores to be high for good athletes.
- In this case, we clearly need to flip both axes...

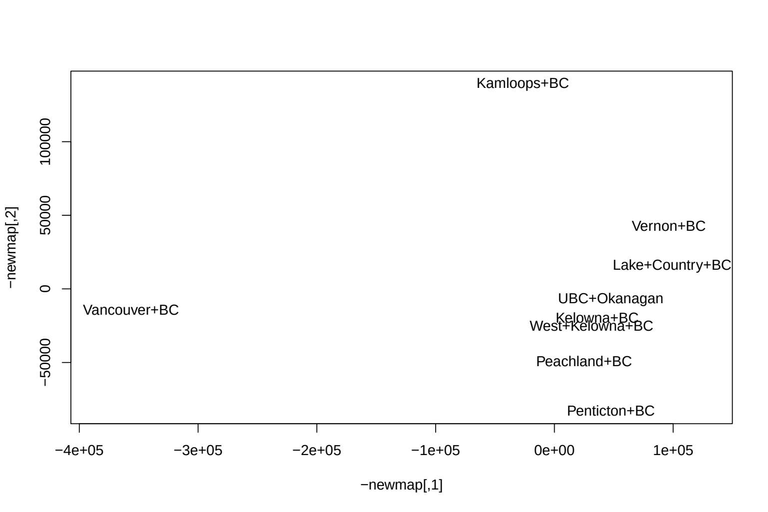
Okanagan Valley Example

```
> plot(newmap, type="n", xlim=c(min(newmap[,1])-30000, max(newmap[,1])+30000 ))
> text(newmap, labels=lcity)
```

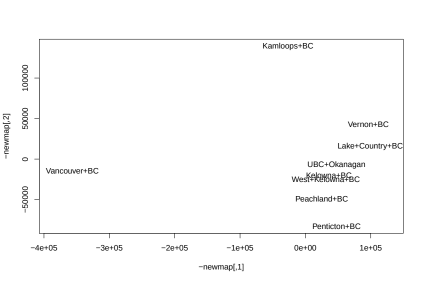


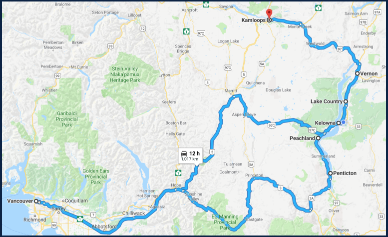
Okanagan Valley Example

```
> plot(-newmap, type="n", xlim=c(min(-newmap[,1])-30000, max(-newmap[,1])+30000 ))
> text(-newmap, labels=lcity)
```



Okanagan Valley Example



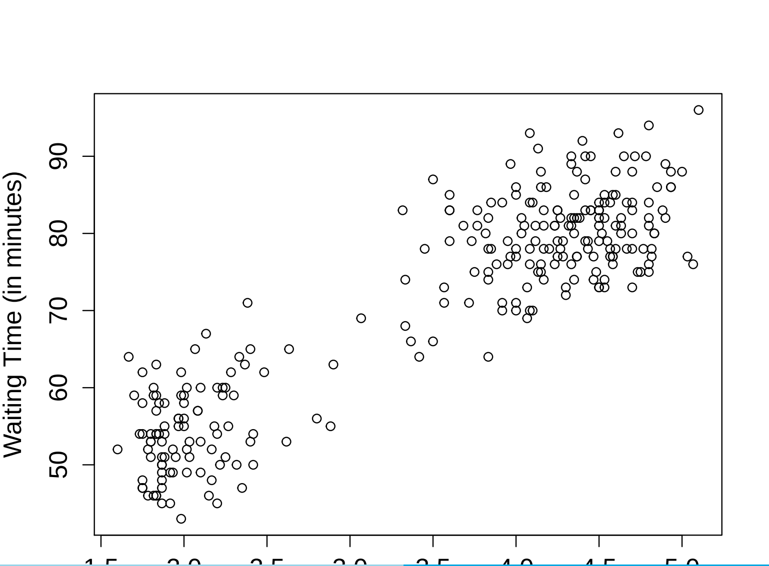


Connection to PCA

- While we didn't walk through the mathematics that connect PCA and MDS, we can do a quick example.
- Recall the faithful data set...

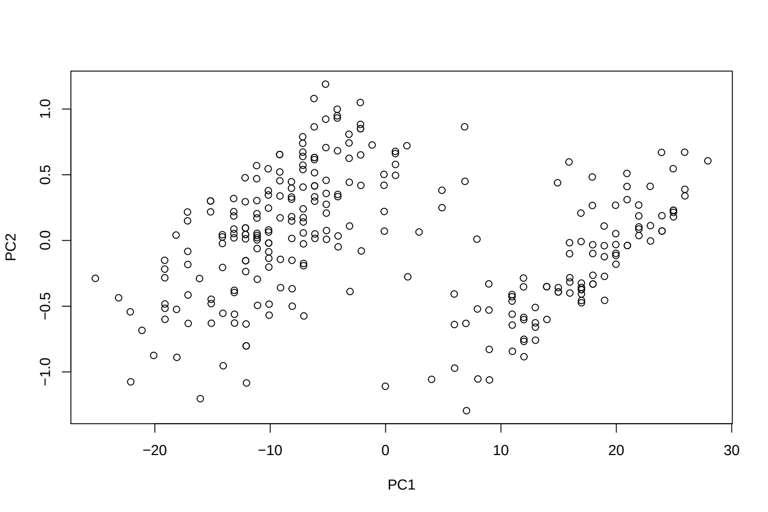
Faithful

```
> plot(faithful)
```



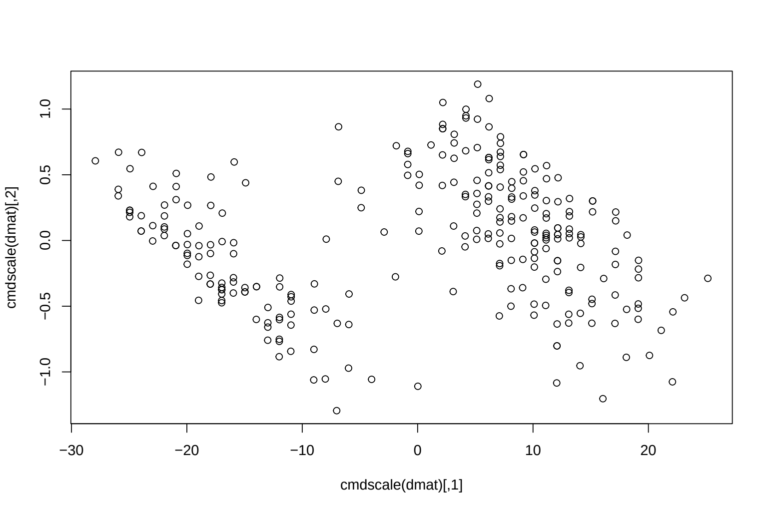
Faithful PCA

```
> pcaf <- prcomp(faithful)
> plot(pcaf$x)
```

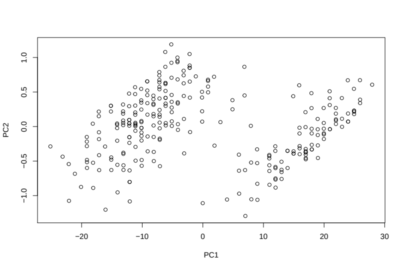


Faithful

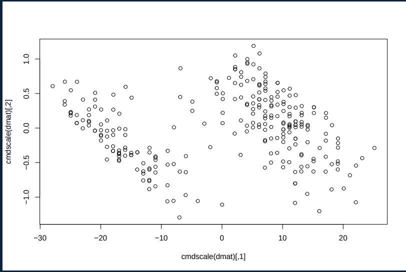
```
> dmat <- dist(faithful)
> plot(cmdscale(dmat))
```



MDS Example



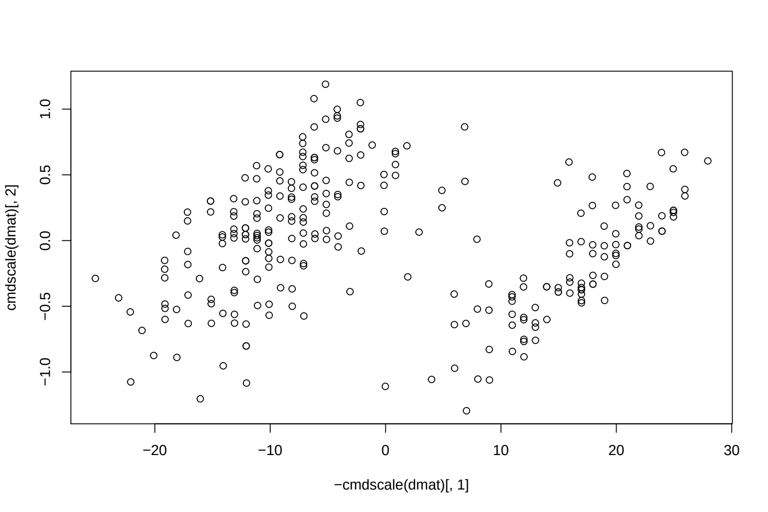
PC2
PC1
1.0
0.5
0.0
−0.5
−1.0
−20
−10
0
10
20
30



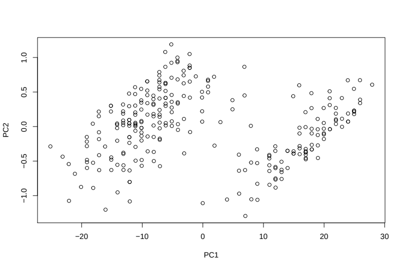
cmdscale(dmat)[,2]
cmdscale(dmat)[,1]
1.0
0.5
0.0
−0.5
−1.0
−30
−20
−10
0
10
20

Faithful

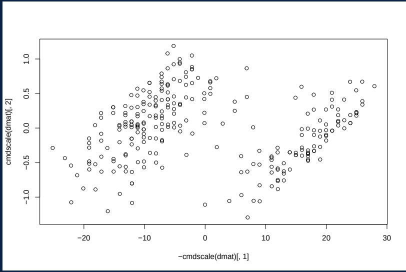
```
> plot(-cmdscale(dmat)[,1], cmdscale(dmat)[,2])
```



MDS Example



PC2
PC1
1.0
0.5
0.0
−0.5
−1.0
−20
−10
0
10
20
30



cmdscale(dmat)[, 2]
−cmdscale(dmat)[, 1]
1.0
0.5
0.0
−0.5
−1.0
−20
−10
0
10
20
30

Parting MDS Thoughts

- While we don't have time in this module to properly address textual data analyses (natural language processing, etc), it's worth noting that MDS is one way to try to spatially visualize non-numeric data in general.
- If you can define pairwise distances, regardless of the data type, then using MDS you can map the observations into a continuous space to visualize, or feed into other supervised algorithms.
- Also note: MDS is quite old. There are more popular ways to achieve similar visualization tasks. t-distributed stochastic neighbour embedding (tSNE) would be one.


UBC
THE UNIVERSITY OF BRITISH COLUMBIA