ETC3250/5250 Introduction to Machine Learning

Week 10: Model-based clustering and self-organising maps

Professor Di Cook

etc3250.clayton-x@monash.edu

Department of Econometrics and Business Statistics

Overview

We will cover:

- Models of multimodality using Gaussian mixtures
- Fitting model-based clustering
- Diagnostics for the model fit
- Self-organising maps and dimension reduction

Model-based clustering

Overview

Model-based clustering makes an assumption about the distribution of the data, primarily

- Assumes the data is a sample from a Gaussian mixture model
- Requires the assumption that clusters have an elliptical shape
- The shape is determined by the variance-covariance of the clusters
- A variety of models is available by using different constraints on the variance-covariance

Model is

$$f(x_i) = \sum_{k=1}^G\pi_kf_k(x_i; \mu_k, \Sigma_k)$$

where f_k is usually a multivariate normal distribution. The parameters are estimated by maximum likelihood, and choice between models is made using BIC.

Parametrisation of the var-cov matrices (1/2)

Constraints applied on cluster variance-covariance:

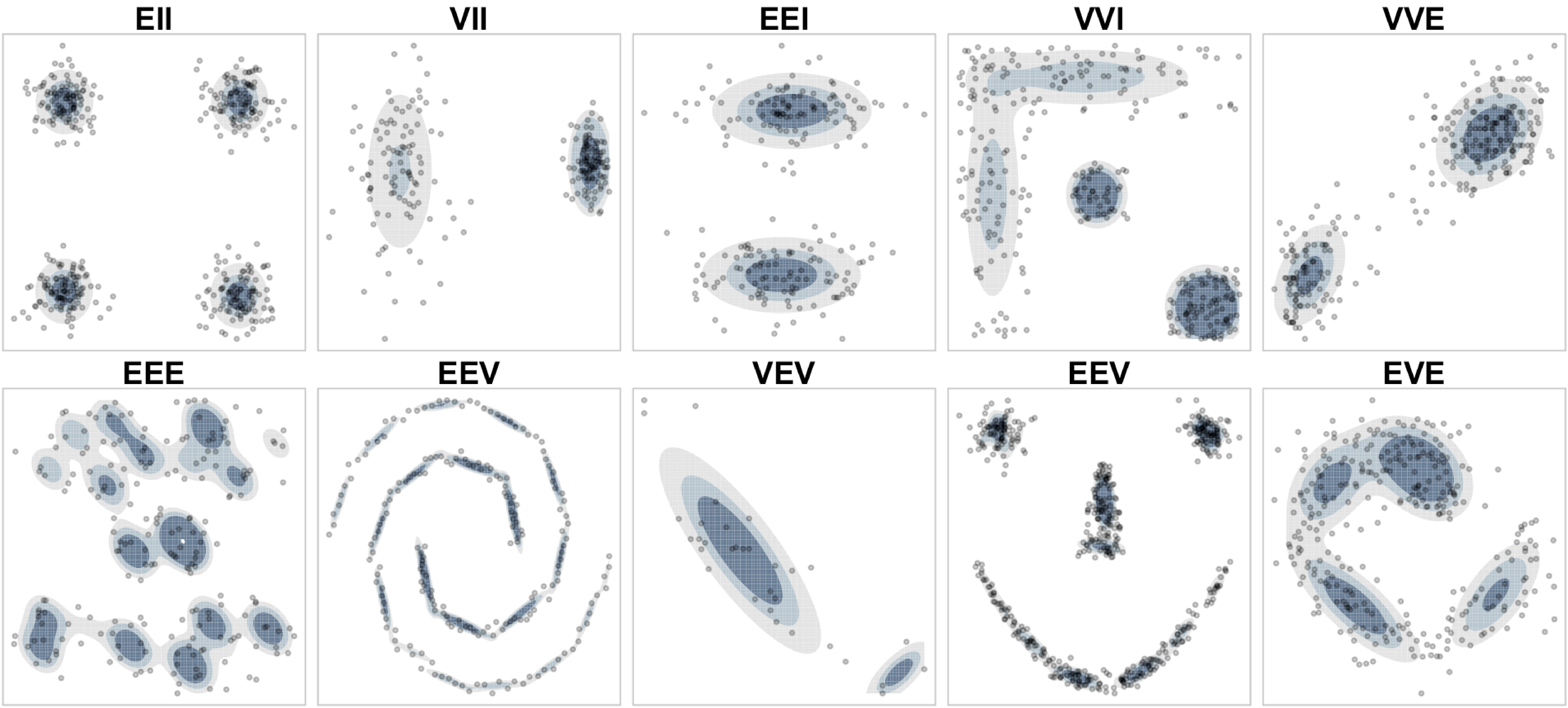
$$\Sigma_k = \lambda_kD_kA_kD_k^\top$$

- volume (λ_k): size of the cluster, ie number of observations
- shape (A_k): difference variances
- orientation (D_k): aligned with axes (low covariance) or not (high covariance)

Model	Family	Volume	Shape	Orientation	Identifier
1	Spherical	Equal	Equal	NA	EII
2	Spherical	Variable	Equal	NA	VII
3	Diagonal	Equal	Equal	Axes	EEI
4	Diagonal	Variable	Equal	Axes	VEI
5	Diagonal	Equal	Variable	Axes	EVI
6	Diagonal	Variable	Variable	Axes	VVI
7	General	Equal	Equal	Equal	EEE
8	General	Equal	Variable	Equal	EVE
9	General	Variable	Equal	Equal	VEE
10	General	Variable	Variable	Equal	VVE
11	General	Equal	Equal	Variable	EEV
12	General	Variable	Equal	Variable	VEV
13	General	Equal	Variable	Variable	EVV
14	General	Variable	Variable	Variable	VVV

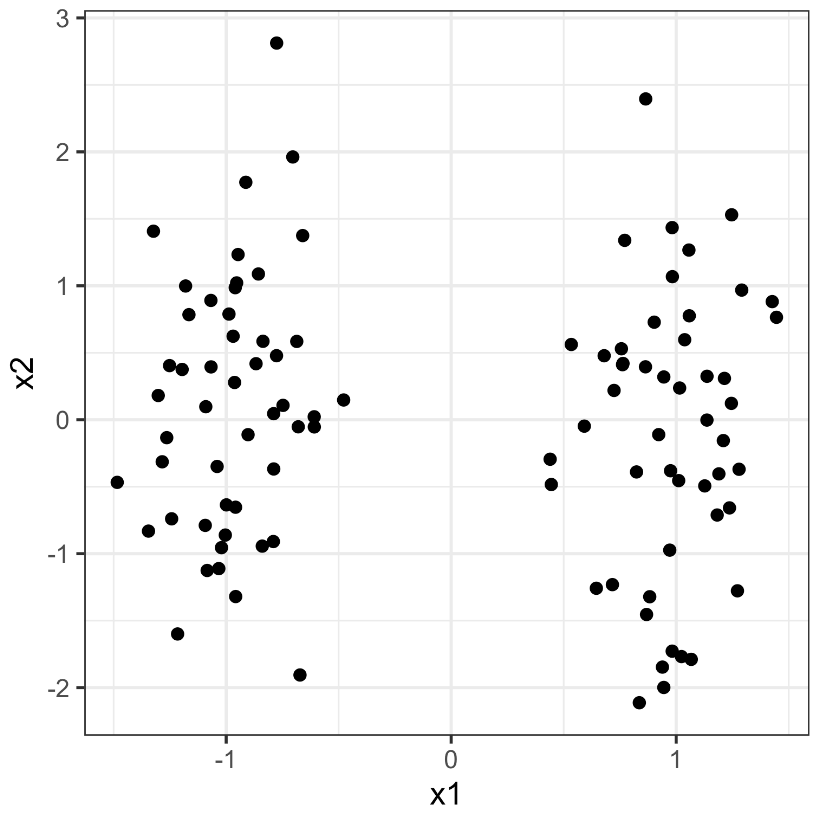
- λI is model 1, EI
- $\lambda DAD^\top$ is model 7, EEE
- $\lambda D_kAD_k^\top$ is model 11, EEV

Parametrisation of the var-cov matrices (2/2)



Source: Boehmke (2020) Hands-on machine learning

Example: nuisance variable (1/3)



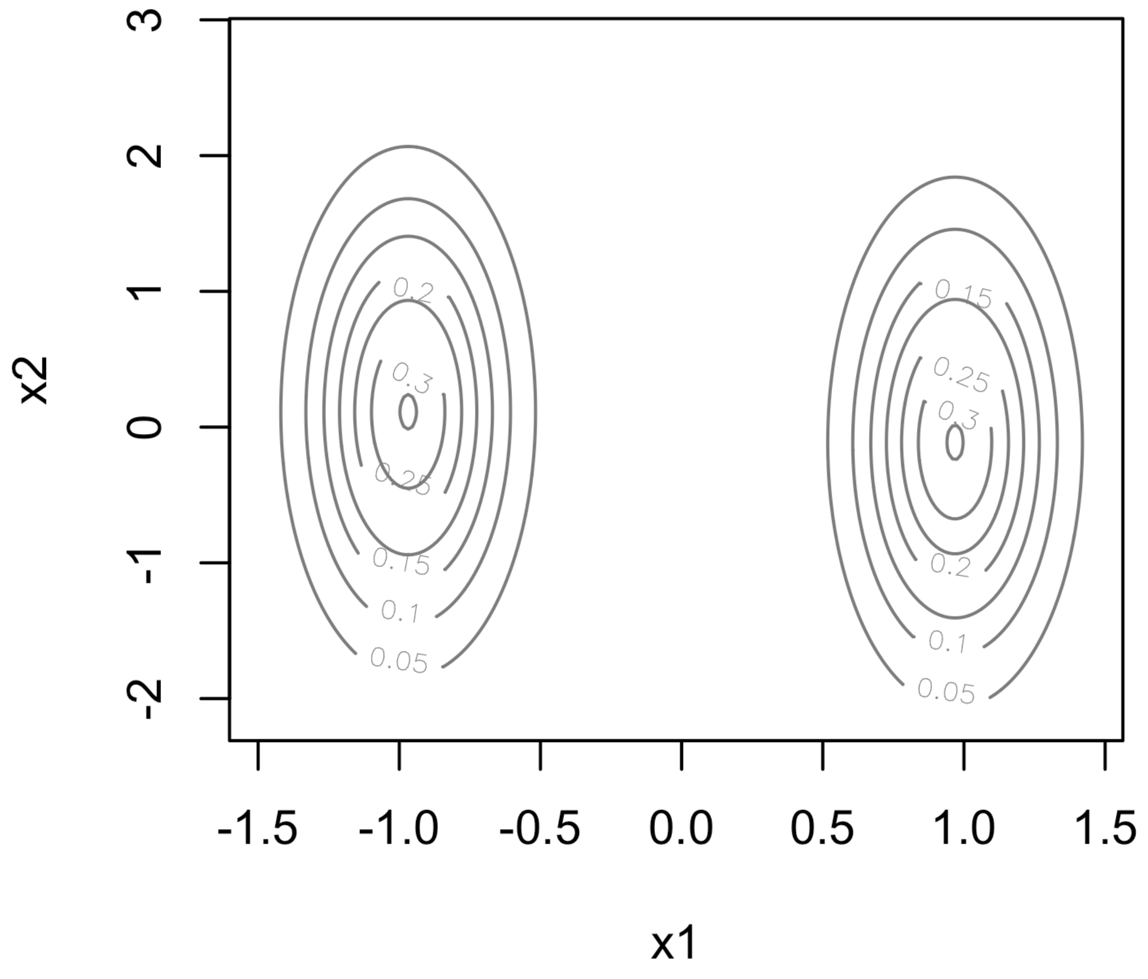

```r
df_mc <- Mclust(df, G = 2)
summary(df_mc)
```

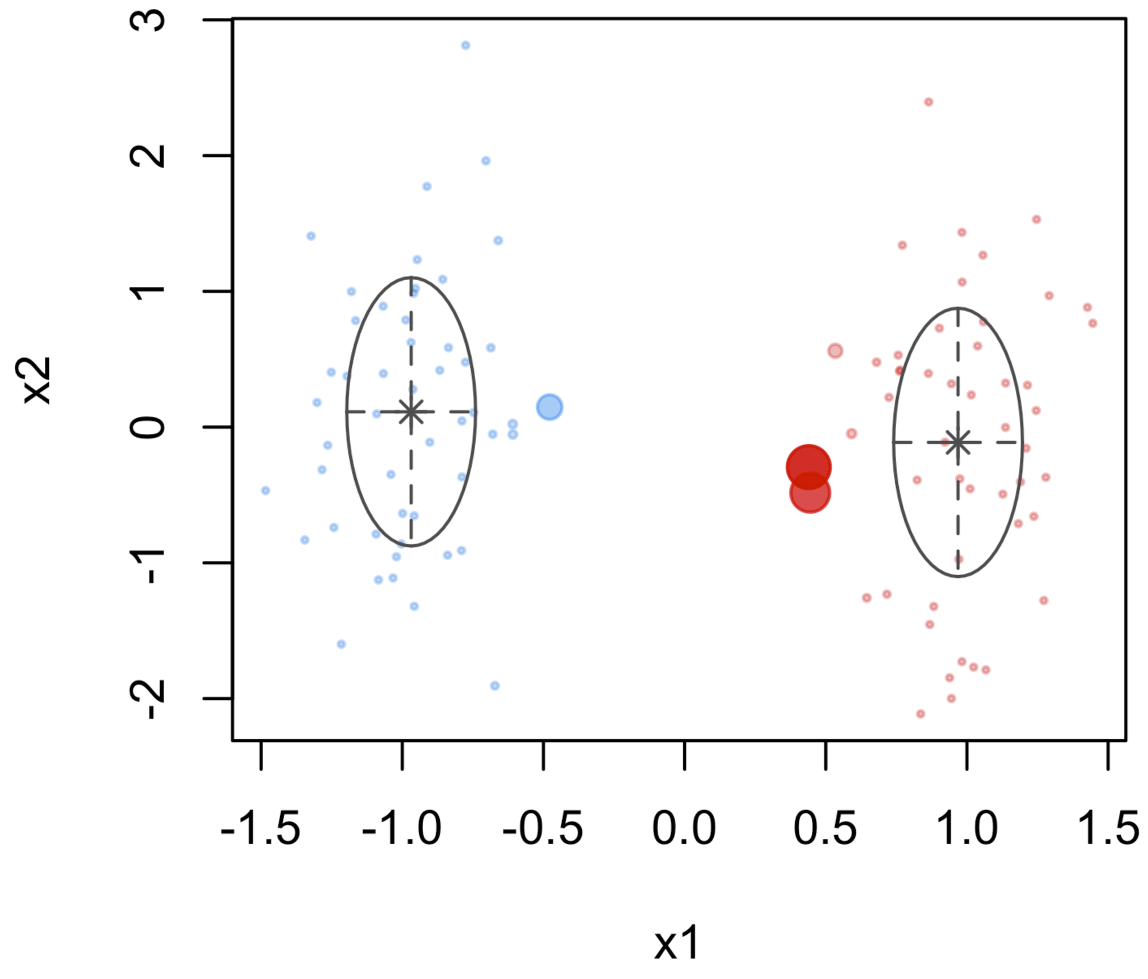
```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust EEI (diagonal, equal volume and shape) model with 2
components:

 log-likelihood   n df  BIC  ICL
           -204 100  7 -441 -441

Clustering table:
 1  2
50 50
```

Example: nuisance variable (2/3)

```
plot(df_mc, what = "density")
```

```
plot(df_mc, what = "uncertainty")
```





Example: nuisance variable (3/3)

Cluster means

```r
options(digits=2)
df_mc$parameters$mean
```

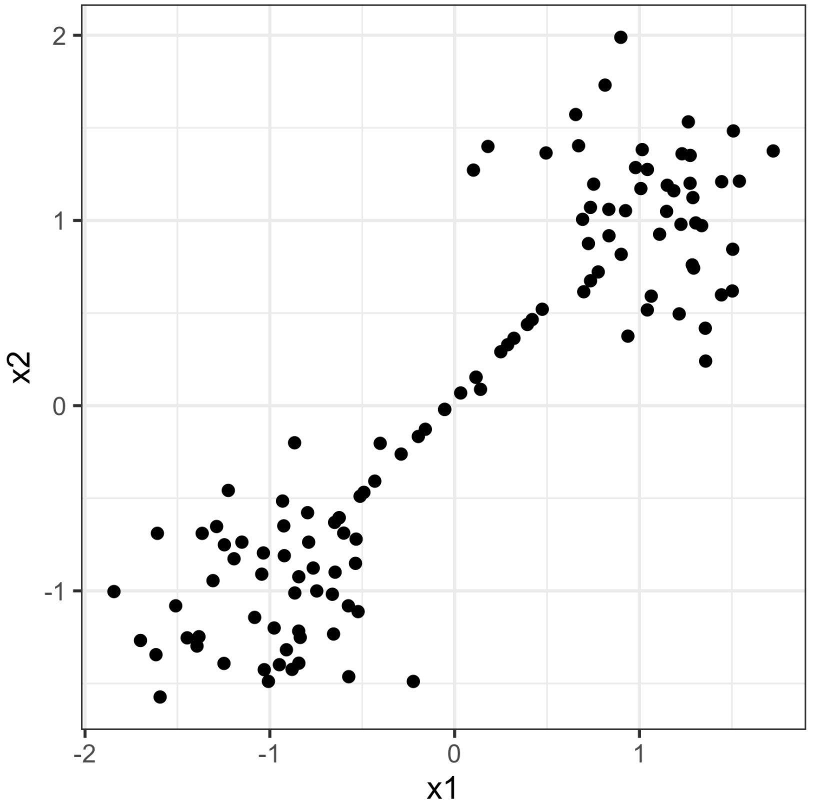
```
    [,1]  [,2]
x1 -0.97  0.97
x2  0.11 -0.11
```

Cluster variance-covariances

```r
df_mc$parameters$variance$sigma
```

```
, , 1

      x1   x2
x1 0.052 0.00
x2 0.000 0.98

, , 2

      x1   x2
x1 0.052 0.00
x2 0.000 0.98
```

Example: nuisance cases (1/3)



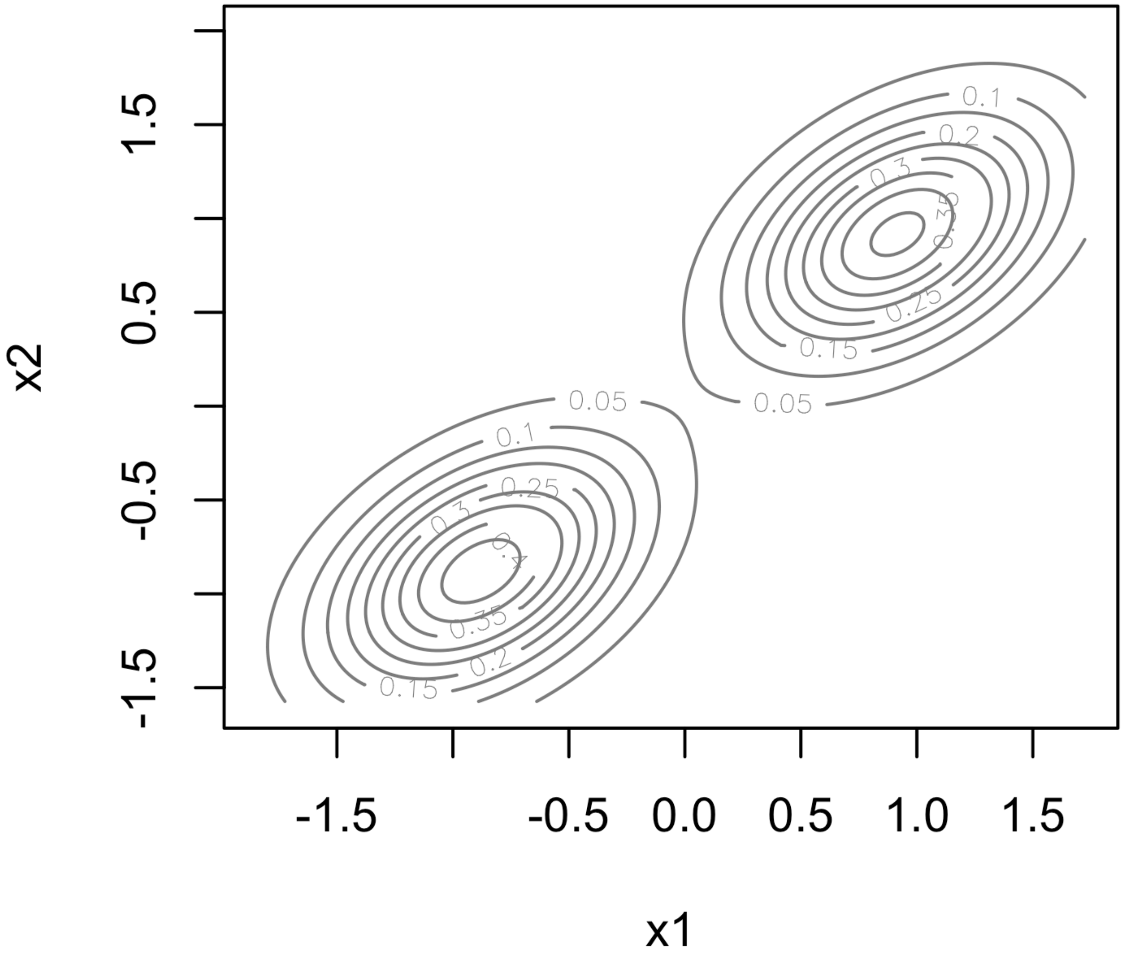

```r
df_mc <- Mclust(df, G = 2)
summary(df_mc)
```

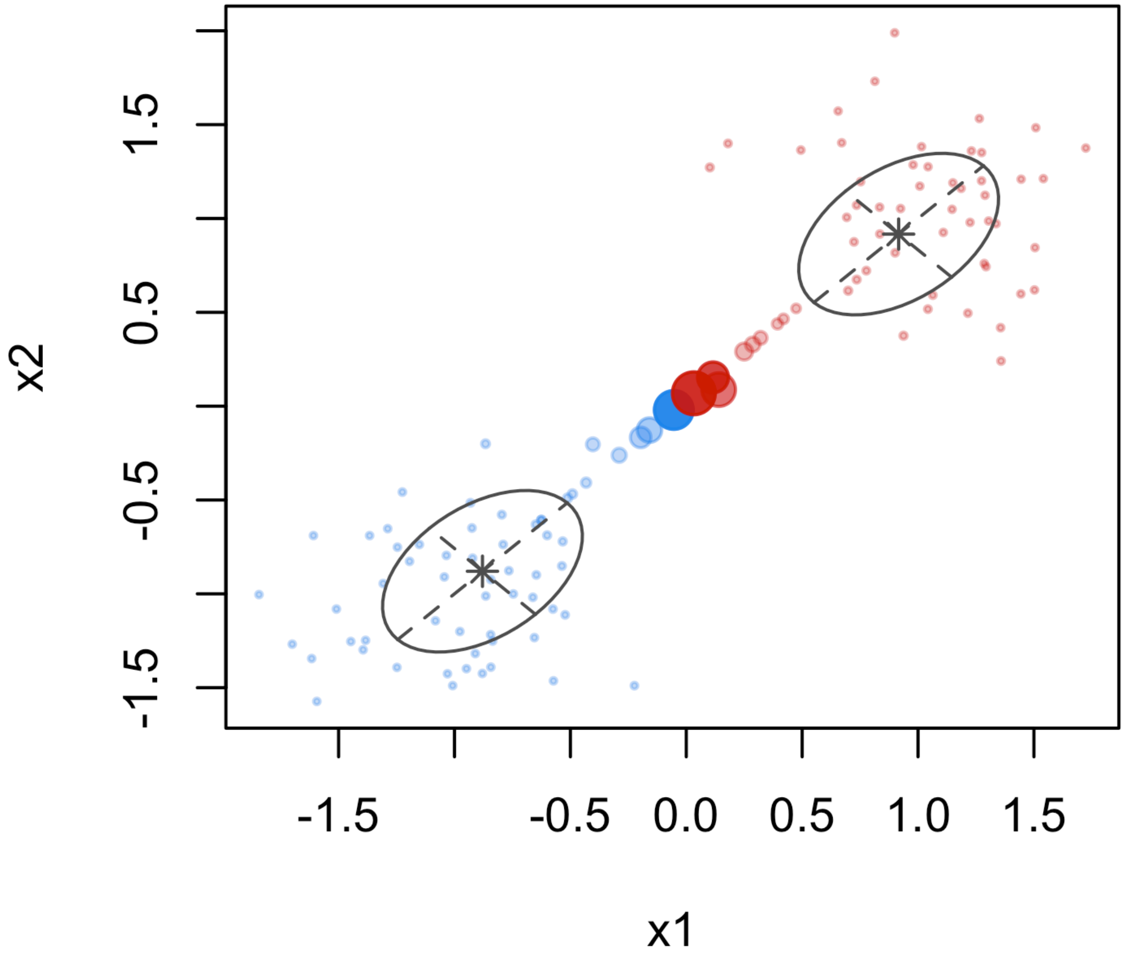
```
----------------------------------------------------
Gaussian finite mixture model fitted by EM algorithm
----------------------------------------------------

Mclust EEE (ellipsoidal, equal volume, shape and
orientation) model with 2 components:

 log-likelihood   n df  BIC  ICL
           -205 120  8 -447 -452

Clustering table:
 1  2
61 59
```

Example: nuisance cases (2/3)

```r
plot(df_mc, what = "density")
```

```r
plot(df_mc, what = "uncertainty")
```





Example: nuisance cases (3/3)

Cluster means

```r
df_mc$parameters$mean
```

```
    [,1] [,2]
x1 -0.88 0.92
x2 -0.88 0.92
```

Cluster variance-covariances

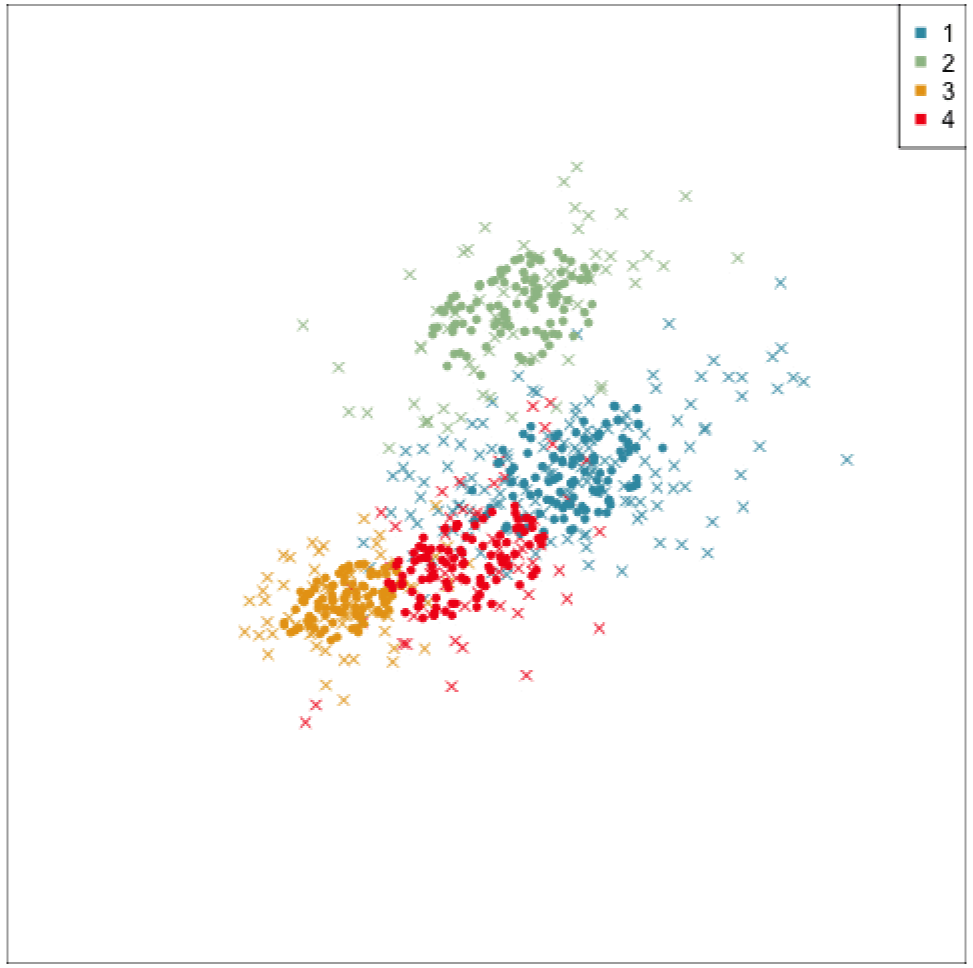
```r
df_mc$parameters$variance$sigma
```

```
, , 1

      x1    x2
x1 0.186 0.081
x2 0.081 0.185

, , 2

      x1    x2
x1 0.186 0.081
x2 0.081 0.185
```

Example: penguins (1/2)

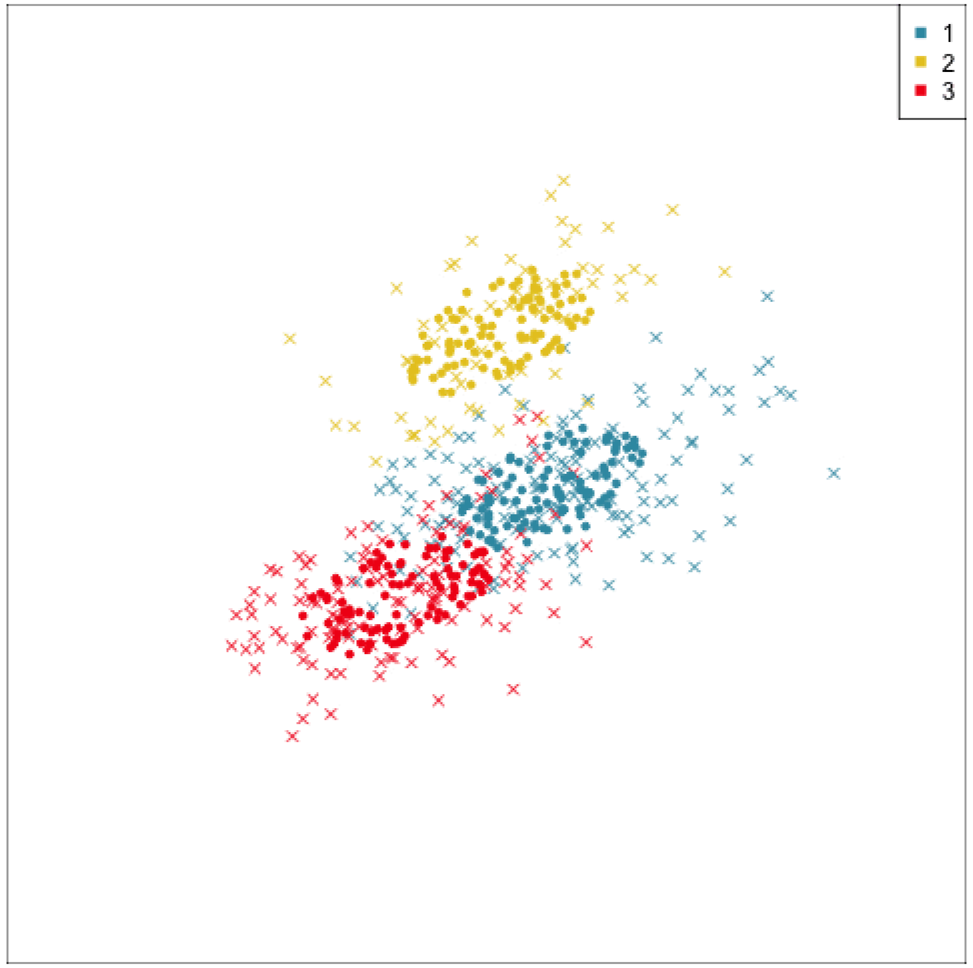
Example: penguins (2/2)

- Code



Best model: VEE, 4

What's wrong with this fit?



model: VEE, 3

Which is the better model? 4 or 3 clusters?

Purely based on how well it fits the data?

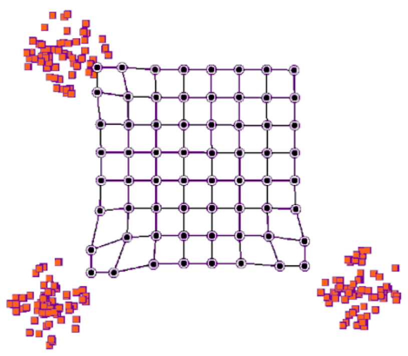
Summary

- Model-based clustering provides a nice automated clustering, if the data has neatly separated clusters, even in the presence of nuisance variables.
- Non-elliptical clusters could be modeled by combining multiple ellipses.
- It is affected by nuisance observations, and has a parameter `noise` to attempt to filter these.
- It may not function so well if the data hasn't got separated clusters.
- \(k\)-means and Wards linkage hierarchical would yield similar results to constraining the variance-covariance model to EEI (or VII, EEE).
- Having a functional model for the clusters is useful.

Self-organising maps

Overview

- A self-organizing map is a constrained k-means algorithm
- A 1D or 2D net is stretched through the data. The knots in the net form the cluster means, and points closest to the knot are considered to belong to that cluster.
- The net provides a low-dimensional summary of the clustering, nodes (and their corresponding clusters) that are close to each other being more similar than those that are further apart.



Source: wikipedia

Algorithm

1. Scale your data
2. Initialize the net defined by the knots (nodes $m_k, k=1, ..., K$): choose the number of nodes in the horizontal (and vertical directions for 2D), and set initial positions of these K nodes (eg first two PCs) in the data space.
3. Loop over data points, $x_i, i=1, ..., n$
   i. find the closest node, m_{k^*}
   ii. for each node, m_k in the neighborhood of m_{k^*} and update it by: $m_k = m_k + \alpha h_k(x_i, m_{k^*}) (x_i-m_{k^*})$, pulling it closer to x_i,

where α is a learning rate function that linearly shrinks from 1 to 0, or function with decreasing value as the number of iterations increases, and h_k is a neighbourhood function, e.g. $h_k(x_i, m_{k^*})=exp(\frac{||x_i-m_{k^*}||}{2\alpha})^2)$ which is a bubble function (within a distance or not).

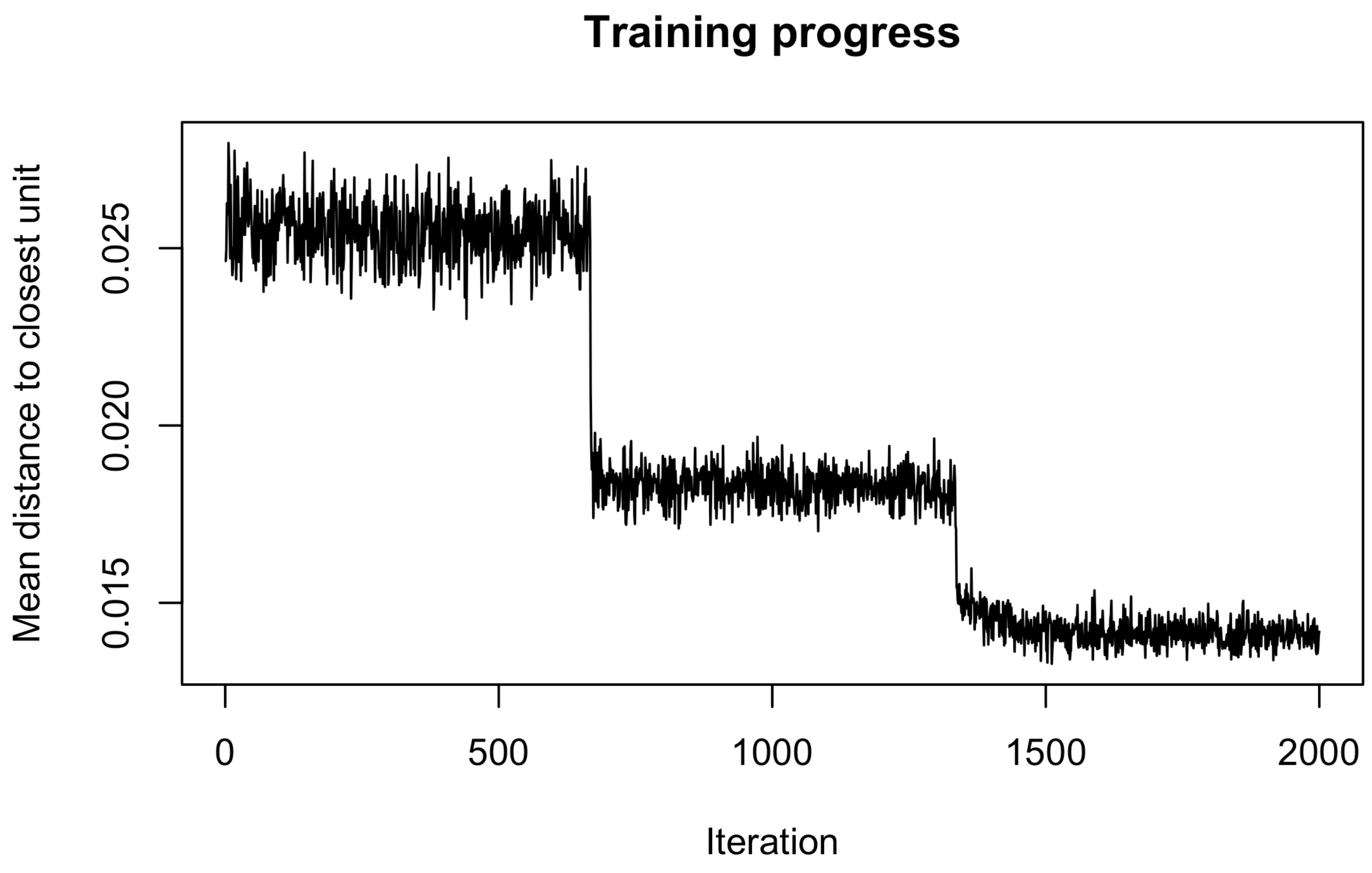
Step 3 is iterated until nodes stop changing position or a stopping rule is satisfied.

Example: penguins (1/2)

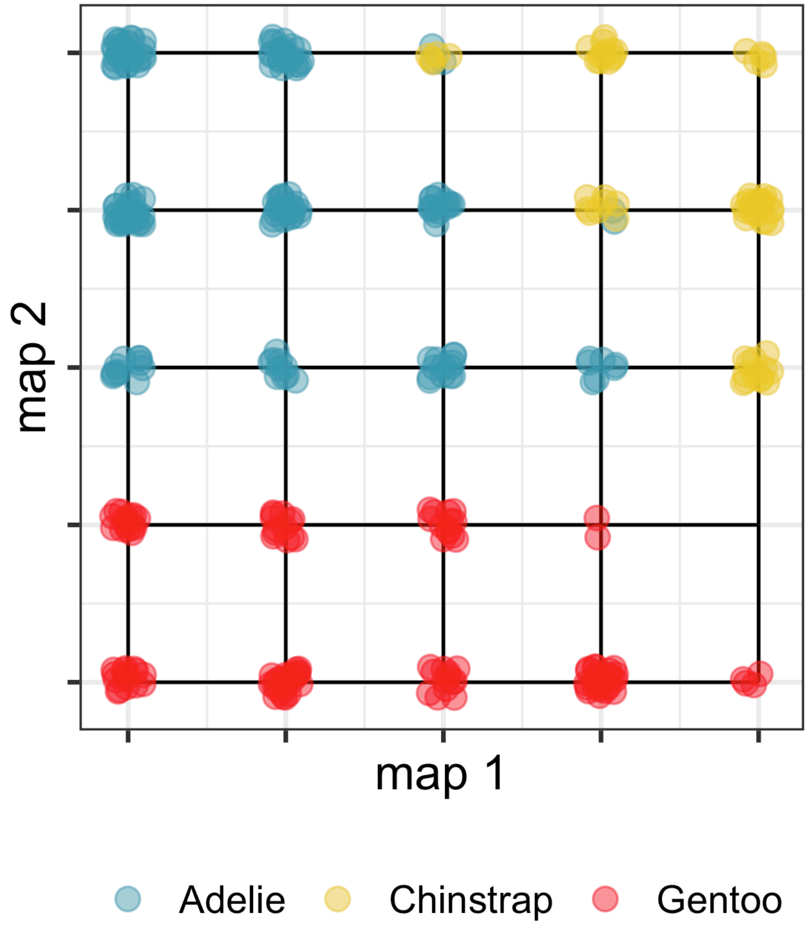
```r
set.seed(947)
p_grid <- kohonen::somgrid(xdim = 5, ydim = 5,
                           topo = 'rectangular
p_init <- somInit(as.matrix(p_std[,2:5]), 5, 5
p_som <- som(as.matrix(p_std[,2:5]),
             rlen=2000,
             grid = p_grid,
             init = p_init)
```

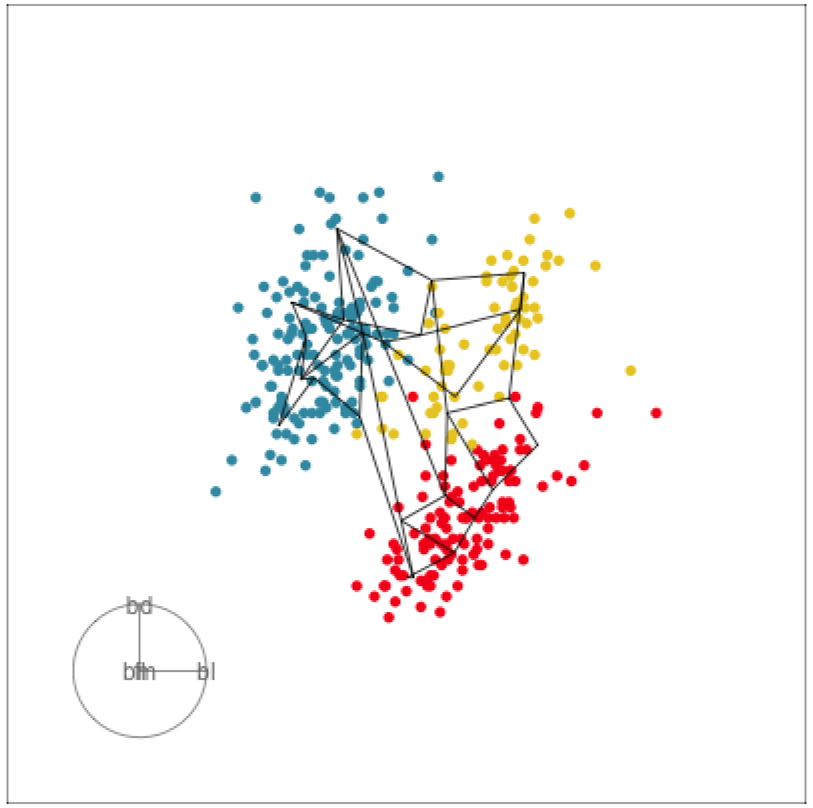
`rlen` controls the length of the optimisation. Tend to need to run it for longer than default.

```r
plot(p_som, type="changes")
```



Example: penguins (2/2)





The net is stretched and clumped into the three clusters in 4D.

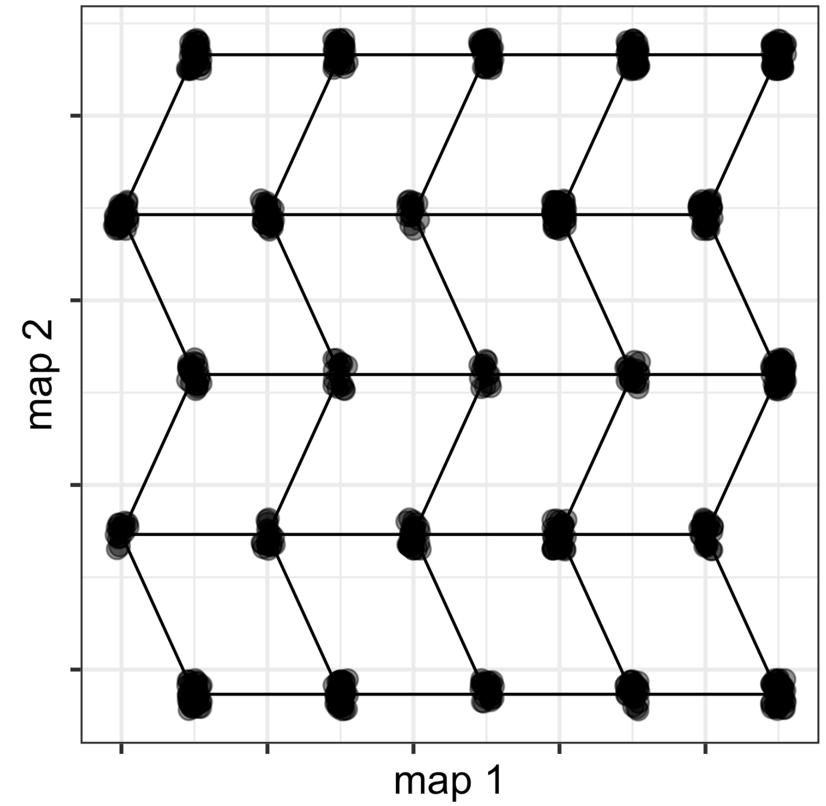
From the map we can see that the clustering has effectively distinguished the species, with some confusion between Chinstrap and Adelie.

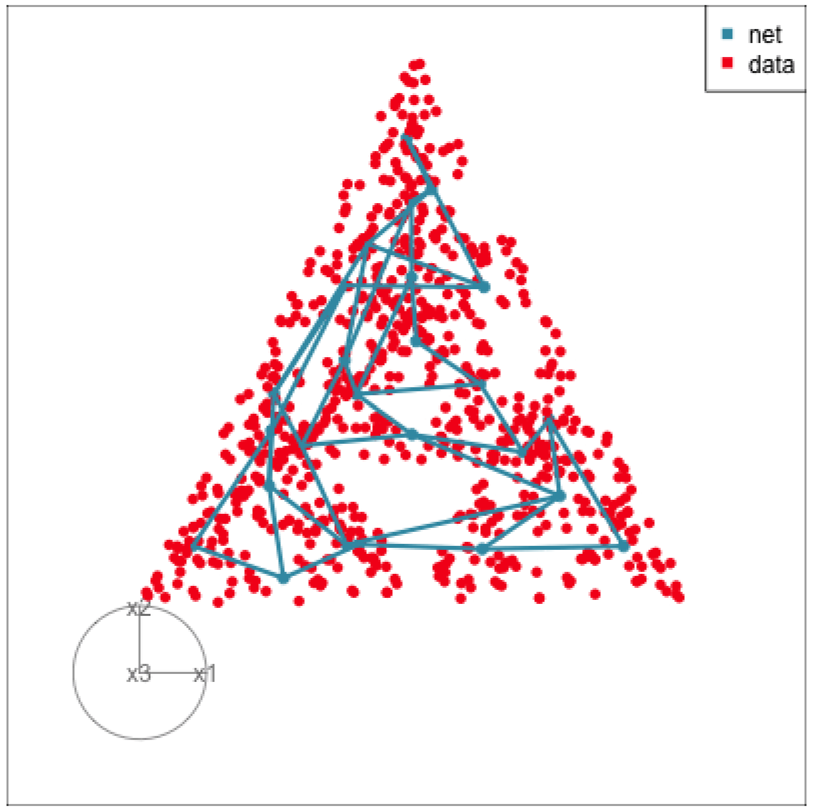
Example: surface (1/2)

```r
c3_som <- som(as.matrix(c3),
  grid = kohonen::somgrid(5, 5, "hexagonal"))
```

```r
plot(c3_som, type="changes")
```

Example: surface (2/2)





- the net stretches into the vertices of the tetrahedron, filling the smaller tetrahedrons
- see the break so that a 2D net fits the 3D object?
- the 7 noise dimensions were ignored

Next: Evaluating your clustering model