STAT40810 — Stochastic Models

Brendan Murphy

Week 10

Finite Mixture Models

Gender Data

- We have data from a study in Saxony, Germany, which seeks to identify the number of male children in 53680 families of size 8.

x	frequency
0	215
1	1485
2	5331
3	10649
4	14959
5	11929
6	6678
7	2092
8	342

- We could model the number of males as binomial with probability of a male, p, and number of trials $n = 8$.
- This yields, $\hat{p} = 0.515$, as a maximum likelihood estimate.

Gender Data

- We can see that the model fit is inadequate.

x	Frequency	Expected
0	215.00	165.22
1	1485.00	1401.69
2	5331.00	5202.65
3	10649.00	11034.65
4	14959.00	14627.60
5	11929.00	12409.87
6	6678.00	6580.24
7	2092.00	1993.78
8	342.00	264.30

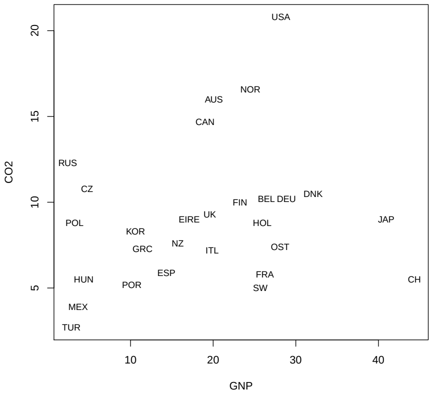
- Also, the sample variance is bigger than expected.
- So, we need an alternative model.

Alzheimer Dataset

- Data were collected on early onset Alzheimer patient symptoms in St. James' Hospital, Dublin.
- Two hundred and forty patients had six behavioural and psychological symptoms (Hallucination, Activity, Aggression, Agitation, Diurnal and Affective) recorded.
- It is believed that the patients cannot be modeled using a single model because there are different subtypes within the disease.
- The number of distinct groups of patients gives an idea of the number of subclasses or syndromes.
- It is believed that two or three groups are more suitable to describe data.

CO_2 Emmissions

- Data were collected on CO_2 emmissions and per capita GNP for a number of countries.



- A close inspection of the scatter plot suggests that there could be two different linear relationships at play here.

Model-Based Clustering/Mixture Models

- Suppose we have data $x = (x_1, x_2, \ldots, x_n)$ sampled from some population.
- We want to build a model that can account for substructure in the population.
- A finite mixture model has the following structure.
  - Assume there are G groups (classes, components, subpopulations, clusters).
  - The probability of an observation coming from group g is τ_g.
  - Each observation within group is modeled using a standard statistical model $p(x|\theta_g)$
- This gives,

$$p(x_n|\tau, \theta, G) = \sum_{g=1}^{G} \tau_g p(x_n|\theta_g).$$

Mixture Models and Clustering

- Finite mixture models can be seen as a way of developing clustering methods which are based on statistical models.
- This is why many mixture model papers refer to *model-based clustering*.
- However, finite mixture models can also be used as a way of extending the flexibility of a standard model, without focussing on clustering.

Model-Based Clustering Approach

...when clustering samples from a population, no cluster method is, a priori believable without a statistical model.

(Aitkin et al, 1981)

...With the underlying probability model, the problems of determining the number of clusters and of choosing an appropriate clustering method become statistical model choice problems.

(Yeung, et al, 2001)

Gender Data

- Let's think about the gender data.
- Suppose we use a finite mixture model (with G components) for modeling this data.
- Each component will have probability τ_g and data within each component will be modeled as binomial with probability p_g.
- An interpretation of this is that:
  - We have G different family types.
  - Each family type has its own propensity to have male children.
  - Each family type occurs with some probability.

Gender Data: Code

- Let's fit some mixture models to the data and see what happens.

```
# Saxony data
n <- 0:8
f <- c(215,1485,5331,10649,14959,11929,6678,2092,342)
x <- rep(n,f)
y<-8-x
dat<-cbind(x,y)
colnames(dat)<-c("M","F")

# Load relevant package (flexmix)
library(mixtools)

# Fit a G component mixture model
G<-1
fit<-multmixEM(dat,k=G)

# Examine the fit
summary(fit)
```

Gender Data: Code

- Let's fit some mixture models to the data and see what happens.

```
# Saxony data
n <- 0:8
f <- c(215,1485,5331,10649,14959,11929,6678,2092,342)
x <- rep(n,f)
y<-8-x
dat<-cbind(x,y)
colnames(dat)<-c("M","F")

# Load relevant package (mixtools)
library(mixtools)

# Fit a G component mixture model
G<-1
fit<-multmixEM(dat,k=G)

# Examine the fit
summary(fit)
```

Gender Data: Model Fit (1)

- The model fit for $G = 1$ is:

```
summary of multmixEM object:
         comp 1
lambda 1.000000
theta1 0.514677
theta2 0.485323
loglik at estimate:  -95587.84
```

Gender Data: Model Fit (2)

- The model fit for $G = 2$ is:

```
summary of multmixEM object:
         comp 1   comp 2
lambda 0.438987 0.561013
theta1 0.475432 0.545386
theta2 0.524568 0.454614
loglik at estimate:  -95570.28
```

Gender Data: Model Fit (3)

- The model fit for $G = 3$ is:

```
summary of multmixEM object:
         comp 1   comp 2   comp 3
lambda 0.147087 0.153361 0.699552
theta1 0.460877 0.463140 0.537288
theta2 0.539123 0.536860 0.462712
loglik at estimate:  -95570.5
```

Model Choice

- The model with the highest BIC is the $G = 2$ model.
- It has been shown that BIC is consistent when choosing G in finite mixture models.
- AIC is not consistent and it tends to overestimate G.
- What does this mean?
- Suppose we have data from a G component mixture model.
- The estimate $\hat{G}$ is consistent if $\hat{G} \to G$ as $n \to \infty$.

Latent Class Analysis

- Latent Class Analysis (LCA) is a model for clustering categorical data.
- Let $x_n = (x_{n1}, x_{n2}, \ldots, x_{nM})$ where x_{nm} takes a value from $\{1, 2, \ldots, C_m\}$.
- In LCA we assume that there is local independence between variables, so that if we knew x_n was in class g we could write it's probability as

$$p(x_n|\theta_g) = \prod_{m=1}^{M} p(x_{nm}|\theta_{gm}) = \prod_{m=1}^{M} \prod_{c=1}^{C_m} \theta_{gmc}^{\mathcal{I}(x_{nm}=c)},$$

  where $\{\theta_{gm1}, \ldots, \theta_{gmC_m}\}$ give the probabilities of observing the categories $\{1, \ldots, C_m\}$ in variable m.
- The θ_g values will characterize and embody the differences between groups.

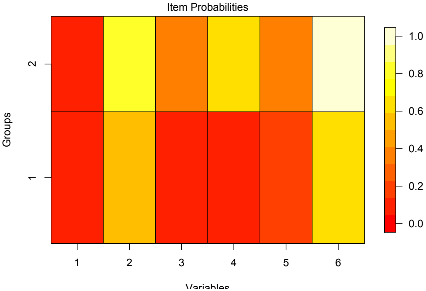
Alzheimer Code

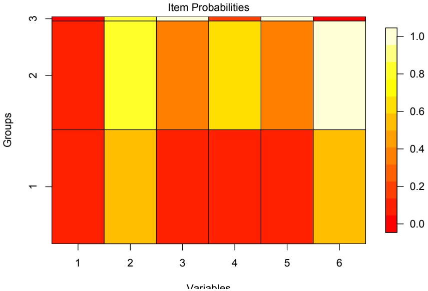
```
#Load the BayesLCA package
library(BayesLCA)

# Load the data
data(Alzheimer)

#Fit the G=2 model
fit2 <- blca.em(Alzheimer, 2)
fit2

#Fit the G=3 model
fit3<- blca.em(Alzheimer, 3, restarts=25)
fit3
```

Alzheimer Results

```
Item Probabilities:

        Hallucination Activity Aggression Agitation Diurnal Affective
Group 1         0.069    0.540      0.108     0.126   0.140     0.598
Group 2         0.093    0.811      0.396     0.669   0.381     0.970

Membership Probabilities:

Group 1 Group 2
   0.58    0.42


Item Probabilities:

        Hallucination Activity Aggression Agitation Diurnal Affective
Group 1         0.062    0.518      0.063     0.132   0.096     0.549
Group 2         0.100    0.790      0.372     0.594   0.364     1.000
Group 3         0.000    0.821      0.998     0.208   1.000     0.000

Membership Probabilities:

Group 1 Group 2 Group 3
  0.502   0.479   0.020
```

Alzhemier Results (2)





CO$_2$ Code: flexmix

```
# Load the flexmix package
library(flexmix)

# Load the CO_2 data
data(CO2data)

# Fit a mixture of experts model with 50 random starting values for the EM algorithm.
# The highest BIC value is stored as bicval and the best fitting model as bestfit
bicval <- Inf
itermax <- 50
for (iter in 1:itermax)
{
  fit<-flexmix(CO2~GNP,data=CO2data,k=2)

   if (bicval>BIC(fit))
  {
    bicval<-BIC(fit)
    bestfit<-fit
    print(c(iter,bicval))
  }
}

# Explore the fitted model
summary(bestfit)
parameters(bestfit)
```

CO_2 Output: flexmix

```
Call:
flexmix(formula = CO2 ~ GNP, data = CO2data, k = 2)

       prior size post>0 ratio
Comp.1 0.244    6     10 0.600
Comp.2 0.756   22     27 0.815

'log Lik.' -66.98375 (df=7)
AIC: 147.9675   BIC: 157.2929


                    Comp.1      Comp.2
coef.(Intercept) 1.4047073  8.65077326
coef.GNP         0.6768985 -0.02224341
sigma            0.8456090  2.13942674
```

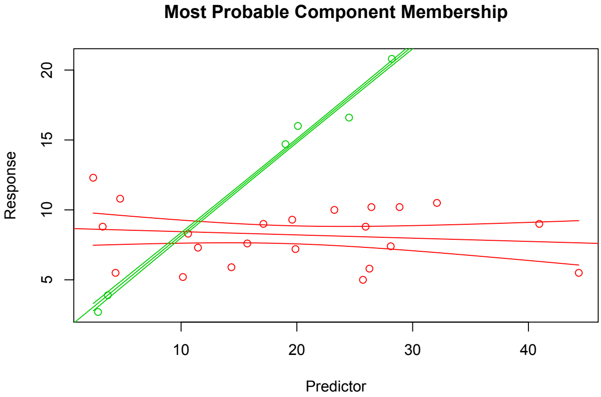
CO_2 Code: mixtools

```
set.seed(1)
# Load the CO_2 data
library(flexmix)
data(CO2data)

# Load the mixtools package
library(mixtools)

# Fit a mixture of experts model with 50 random starting values for the EM algorithm.
# The highest BIC value is stored as bicval and the best fitting model as bestfit
bicval <- -Inf
itermax <- 50
for (iter in 1:itermax)
{
  G<-2
  fit<-regmixEM(CO2data$CO2,CO2data$GNP,k=G)
  n<-nrow(CO2data)
  p<-nrow(fit$beta)*G+G+(G-1)
  fitbic <- 2*fit$loglik - log(n)*p
   if (bicval<fitbic)
  {

    bicval<-fitbic
    bestfit<-fit
    print(c(iter,bicval))
  }
}

# Explore the fitted model
summary(bestfit)
plot(bestfit,which=2)
```

CO$_2$ Output: mixtools

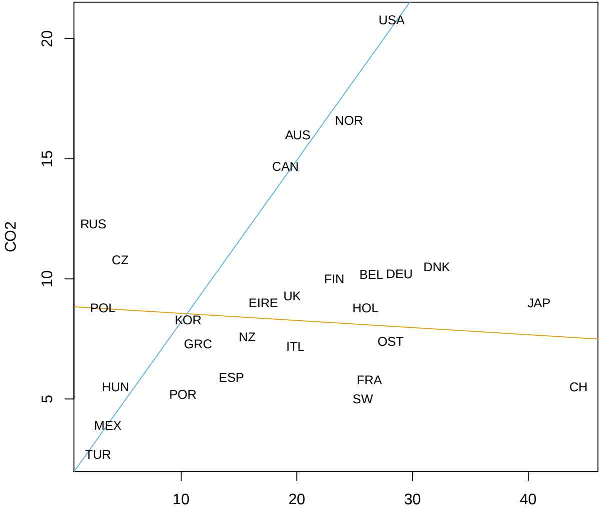
```
summary of regmixEM object:
          comp 1   comp 2
lambda  0.7549234 0.245077
sigma   2.0493214 0.809387
beta1   8.6789541 1.415133
beta2  -0.0233429 0.676597
loglik at estimate:  -66.93977
```

CO_2 Output: mixtools



CO_2 Output: mixtools

With country names we get:


CO2
USA
NOR
AUS
CAN
RUS
CZ
DNK
BEL
DEU
FIN
UK
EIRE
JAP
POL
HOL
KOR
NZ
GRC
ITL
OST
ESP
FRA
HUN
POR
SW
CH
MEX
TUR