Ve406 Lecture 22

Jing Liu

UM-SJTU Joint Institute

August 1, 2018

- In factor analysis, we have implicitly assumed the latent factor is continuous

$$\mathbf{X}_{n\times k} = \mathbf{F}_{n\times 1}\mathbf{W}_{k\times 1}^{\mathrm{T}} + \boldsymbol{\epsilon}_{n\times k}$$

- What if the latent variables are not continuous but categorical? e.g.

$$\begin{aligned} Z &\sim \mathrm{Binomial}\,(1, 0.2) \\ X \mid z = 0 &\sim \mathrm{Normal}\,(0, 9) \\ X \mid z = 1 &\sim \mathrm{Normal}\,(1, 9) \end{aligned}$$

which corresponds to

$$\mathbf{X}_{n\times 1} = \mathbf{Z}_{n\times 1}\mathbf{W}_{1\times 1}^{\mathrm{T}} + \boldsymbol{\epsilon}_{n\times 1}$$

- More generally, the following is known as a Gaussian mixture model.

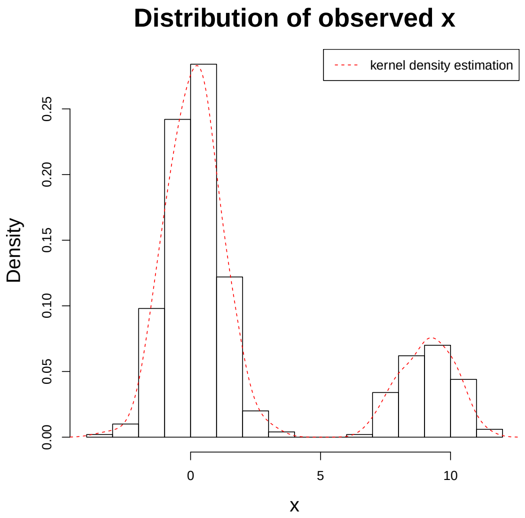
$$\begin{aligned} Z &\sim \mathrm{Multinomial}\,(1, \mathbf{p}) \\ X \mid z = k &\sim \mathrm{Normal}\left(\mu_k, \sigma_k^2\right) \end{aligned}$$

where $\mathbf{p}$ is a vector probabilities, $\mathbf{1}^{\mathrm{T}}\mathbf{p} = 1$, known as the mixing proportions.

- In general, a mixture model assumes x_i are generated in the following fashion

```
> set.seed(2)
> n = 500
> z.vec = rbinom(n, size = 1, prob = 0.2)
> # z is latent, thus not observed in practice

> # x is observed data
> x.vec = double(n)
>
> for (i in 1:n){
+   if (z.vec[i] == 0) {
+     x.vec[i] = rnorm(1, mean = 0, sd = 1)
+   } else {
+     x.vec[i] = rnorm(1, mean = 9, sd = 1)
+   }
+ }
```


Distribution of observed x
kernel density estimation
Density
0.00
0.05
0.10
0.15
0.20
0.25
0
5
10
x

- In terms of probability density function, it means

$$f_{X,Z}(x,z) = f_Z(z) f_{X|Z}(x \mid z)$$

- Consider the marginal distributions,

$$f_X(x) = \sum_z f_Z(z) f_{X|Z}(x \mid z) = \sum_z \Pr(Z = z) f_{X|Z}(x \mid z)$$

- Since the following formula is also true for the joint density

$$f_{X,Z}(x,z) = f_Z(z) f_{X|Z}(x \mid z) = f_X(x) f_{Z|X}(z \mid x)$$

we have

$$f_{Z|X}(z \mid x) = \frac{f_Z(z) f_{X|Z}(x \mid z)}{f_X(x)} = \frac{f_Z(z) f_{X|Z}(x \mid z)}{\sum_z \Pr(Z = z) f_{X|Z}(x \mid z)}$$

Q: What is the significance of this formula?

- Of course, the parameters are unknown and need to be estimated in practice,

$$
\begin{aligned}
Z &\sim \text{Binomial}\left(1, p\right) \\
X \mid z = 0 &\sim \text{Normal}\left(\mu_0, \sigma_0^2\right) \\
X \mid z = 1 &\sim \text{Normal}\left(\mu_1, \sigma_1^2\right)
\end{aligned}
$$

- The MLE of $\boldsymbol{\theta}^{\mathrm{T}} = [p, \mu_0, \mu_1, \sigma_0, \sigma_1]$ cannot be computed in a closed form.

$$
\begin{aligned}
f_X(x; p, \mu_0, \mu_1, \sigma_0, \sigma_1) &= \sum_z \Pr(Z = z) f_{X|Z}(x \mid z) \\
&= (1-p) \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left(-\frac{(x_i - \mu_0)^2}{2\sigma_0^2}\right) \\
&\quad + p \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{(x_i - \mu_1)^2}{2\sigma_1^2}\right)
\end{aligned}
$$

Q: What is the likelihood function if we assume observations are independent?

- We can numerically find MLE,

```
> obj_func = function(theta, x) {
+   p      = theta[1]
+   p0     = 1 - p
+   mu0    = theta[2]
+   mu1    = theta[3]
+   sigma0 = theta[4]
+   sigma1 = theta[5]
+
+   pdf    = p0 * dnorm(x, mu0, sigma0) +
+     p * dnorm(x, mu1, sigma1)
+
+   res    = -sum(log(pdf))
+   return(res)
+
+ }

> res.nlm = nlm( # Newton based numerical methods
+   obj_func, c(.25, 10, 10, 10, 10), x.vec)
```

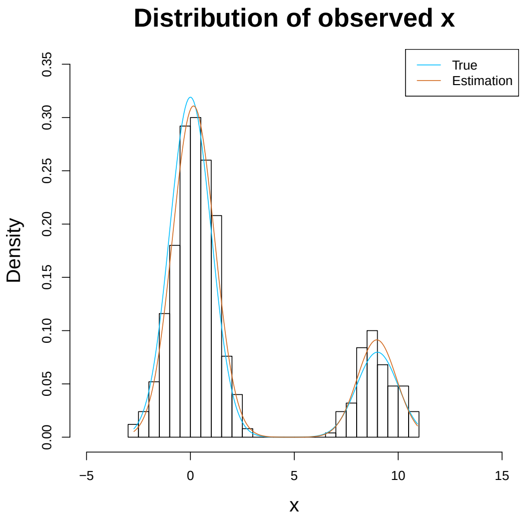
```
> res.nlm # True parameters [0.2, 0, 9, 1, 1]
```

```
$estimate
[1] 0.2159996 0.1487675 8.9709822 1.0059598 0.9431263
```

```
> p      = res.nlm$estimate[1]; p0 = 1 - p
> mu0    = res.nlm$estimate[2]
> mu1    = res.nlm$estimate[3]
> sigma0 = res.nlm$estimate[4]
> sigma1 = res.nlm$estimate[5]

> x.p = seq(min(x.vec), max(x.vec), length = 200)
>
> est.pdf = p0 * dnorm(x.p, mu0, sigma0) +
+   p * dnorm(x.p, mu1, sigma1)

> true.pdf = 0.8 * dnorm(x.p, 0, 1) +
+   0.2 * dnorm(x.p, 9, 1)
```


Distribution of observed x
True
Estimation
Density
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
−5
0
5
10
15
X

- Of course, R has a package and a function for mixture models

```
> library(mixtools)
> res = normalmixEM(x.vec)
> res
```

```
$lambda
[1] 0.784 0.216

$mu
[1] 0.148769 8.970986

$sigma
[1] 1.0059601 0.9431269
```

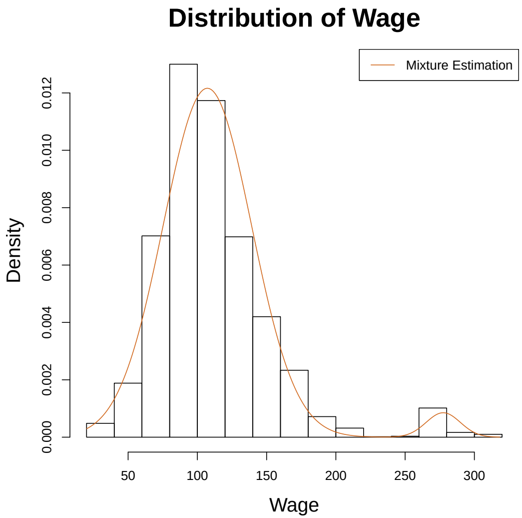
which is more stable and faster than general purpose minimisation routines.

Q: Now we know how to estimate parameters in a mixture model, what is the significance of it in terms of regression?

- Let us finish lectures for this semester by looking it my archenemy again!

wage	Raw wage in the Mid-Atlantic region	
age	Age of the worker	
year	The year that wage information was recorded	
education	A factor with levels:	1. $<$ HS Grad 2. HS Grad 3. Some College 4. College Grad 5. Advanced Degree

```
> library(ISLR);
> wage.df =
+   Wage[, c("year", "age", "education", "wage")]
>
> res = normalmixEM(wage.df$wage)
```


Distribution of Wage
Mixture Estimation
Density
0.000
0.002
0.004
0.006
0.008
0.010
0.012
50
100
150
200
250
300
Wage

```
> str(wage.df)
```

```
'data.frame':   3000 obs. of  4 variables:
 $ year     : int  2006 2004 2003 2003 2005 2008 2009 2008 2006 2004 ...
 $ age      : int  18 24 45 43 50 54 44 30 41 52 ...
 $ education: Factor w/ 5 levels "1. < HS Grad",..: 1 4 3 4 2 4 3 3 3 2 ...
 $ wage     : num  75 70.5 131 154.7 75 ...
```

```
> X = as.matrix(wage.df[, 1:2])
>
> X = cbind(X, as.integer(wage.df$education))
>
> wage.gauss.MM = regmixEM(wage.df[,4], X, k=2)
```

```
> summary(wage.gauss.MM)
```

```
summary of regmixEM object:
             comp 1       comp 2
lambda  7.48706e-02     0.925129
sigma   5.42237e+01    25.622410
beta1  -2.99259e+03 -2120.478107
beta2   1.48038e+00     1.081827
beta3   1.72743e+00     0.484611
beta4   3.92691e+01    12.019221
loglik at estimate:  -14510.38
```

The end of Ve406!