Stat 8003, Homework 5

Group G: `sample( c( "David" , "Andrew", "Salam" ))`

October 2, 2014

Question 5.1. Consider a simulated dataset. Assume that the data $x_1, x_2, \cdots, x_n$ follows the following distribution:

$$x_i \sim f(x_i) = \pi_0 f_0(x_i) + \pi_1 f_1(x_i)$$

where $f_0(x_i) = 1(0 \le x_i \le 1)$ is the density function of the uniform and $f_1(x_i) = \beta(1 - x)^{\beta-1}$ is the density function of $Beta(1, \beta)$. The group information can be treated as a missing value and is denoted as z_i. Let $y_i = (x_i, z_i)$ be the complete data.

(a) Derive the complete likelihood function;

(b) Use the EM algorithm to derive the estimator for π_0 and β;

(c) Apply your method to the data set, estimate π_0 and β and the calculate $\texttt{fdr}_i = P(Z_i = 0 \mid x_i)$. (This score is called the local $\texttt{fdr}$ score.)

(d) Classify x_i to the first group if $\texttt{fdr}_i(x_i) > 0.5$. Compare your classification with the actual group information, what is the total number of falsely classified data?

Answer:

(a) First, the *incomplete* likelihood function is given to be:

$$L(\theta\,; \mathbf{X}) = \prod_{i=1}^{n} \left(\pi_0 1 + \pi_1 \beta (1 - x_i)^{\beta-1}\right)$$

Then the *complete* likelihood function is:

$$\boxed{L(\theta\,; \mathbf{Y}) = \prod_{i=1}^{n} \left(1(Z_i = 0)\; \pi_0 + 1(Z_i = 1)\; \pi_1\; \beta(1 - x_i)^{\beta-1}\right)}$$

An alternative way of writing this likelihood is:

$$f(x_i, z_i \mid \theta) = \begin{cases} \pi_0 & \text{if } Z_i = 0 \\ \pi_1\; \beta(1 - x_i)^{\beta-1} & \text{if } Z_i = 1 \end{cases}$$

(b) To get the estimates for π_0 and β, we first find the expected value of the *log* of the *complete likelihood* function with respect to Z (the so called Q function). As in the notation used in class, let θ^t stand for the parameter estimates obtained at iteration t of the EM algorithm (so $\theta^t = (\pi_0^t, \beta^t)$).

$$\begin{aligned} Q(\theta \mid \theta^t) &= \mathrm{E}\ \log(L(\theta\,; \boldsymbol{Y})) \\ &= \mathrm{E}\ \log\left(\prod_{i=1}^{n}\left(1(Z_i=0)\ \pi_0 + 1(Z_i=1)\ \pi_1\ \beta(1-x_i)^{\beta-1}\right)\right) \\ &= \mathrm{E}\ \left[\sum_{i=1}^{n}\log\left(1(Z_i=0)\ \pi_0 + 1(Z_i=1)\ \pi_1\ \beta(1-x_i)^{\beta-1}\right)\right] \end{aligned}$$

The last expression in the brackets is either $\log(\pi_0)$ or $\log(\pi_1\ \beta(1-x_i)^{\beta-1})$, depending on the outcome of Z. So

$$\begin{aligned} Q(\theta \mid \theta^t) &= \sum_{i=1}^{n}\left(\ \mathrm{E}\ 1(Z_i=0)\ \log(\pi_0) + \mathrm{E}\ 1(Z_i=1)\ \log(\pi_1\ \beta(1-x_i)^{\beta-1})\ \right) \\ &= \sum_{i=1}^{n}\left(\ P(Z_i=0 \mid x_i,\theta)\ \log(\pi_0) + P(Z_i=1 \mid x_i,\theta)\ \log(\pi_1\ \beta(1-x_i)^{\beta-1})\ \right) \end{aligned}$$

Where the last equality follows because the expectation of the indicator function of a `r.v.` is simply the probability of the corresponding event.

These probabilities will be computed using Bayes rule and denoted by T_{ij}^t:

$$T_{ij}^t = P(Z_i = j \mid x_i, \theta) = \frac{P(x_i \mid Z_i = j)P(Z_i = j)}{\sum_{j=0}^{1} P(x_i \mid Z_i = j)P(Z_i = j)} \quad \text{for}\ \ j = 0, 1$$

Thus

$$T_{i0}^t = \frac{\pi_0^t}{\pi_0^t + \pi_1^t\ \beta^t(1-x_i)^{\beta^t-1}}$$

$$T_{i1}^t = \frac{\pi_1^t\ \beta^t(1-x_i)^{\beta^t-1}}{\pi_0^t + \pi_1^t\ \beta(1-x_i)^{\beta^t-1}}$$

(where the superscript ^{t} marks the values of the parameters obtained at the the t-th iteration.)

Rewriting the Q function:

$$\begin{aligned} Q(\theta \mid \theta^t) &= \sum_{i=1}^{n} \left(T_{i0}^t \log(\pi_0) + T_{i1}^t \log(\pi_1\, \beta(1-x_i)^{\beta-1}) \right) \\ &= \sum_{i=1}^{n} \left(T_{i0}^t \log(\pi_0) + T_{i1}^t \log(1-\pi_0) \; + T_{i1}^t \log(\beta(1-x_i)^{\beta-1} \right) \end{aligned}$$

We maximize it with respect to π_0 and β. Setting Q's partial derivatives to zero,

$$\begin{aligned} \frac{d}{d\pi_0} Q(\theta \mid \theta^t) &= \sum_{i=1}^{n} \left(\; T_{i0}^t \frac{1}{\pi_0} - T_{i1}^t \frac{1}{1-\pi_0} \; \right) = 0 \\ \frac{d}{d\beta} Q(\theta \mid \theta^t) &= \sum_{i=1}^{n} \left(\; T_{i1}^t \frac{1}{\beta} + T_{i1}^t \log(1-x_i) \; \right) = 0 \end{aligned}$$

We obtain

$$\boxed{\pi_0^{t+1} = \frac{\sum_{i=1}^n T_{i0}^t}{\sum_{i=1}^n (T_{i0}^t + T_{i1}^t)} = \frac{\sum_{i=1}^n T_{i0}^t}{n}}$$

$$\boxed{\beta^{t+1} = \frac{-\sum_{i=1}^n T_{i1}^t}{\sum_{i=1}^n T_{i1}^t \log(1-x_i)}}$$

(c) The EM algorithm converges to the following values of θ (code attached separately):

$$\boxed{\pi_0 = 0.696794 \quad \text{and} \quad \beta = 11.093249}$$

We use these parameters to obtain the `fdr` score for the i^{th} observation as follows:

$$\texttt{fdr}_i = P(Z_i = 0 \mid x_i) = T_{i0} = \frac{\pi_0}{\pi_0 + \pi_1\, \beta(1-x_i)^{\beta-1}}$$

The following code snippet was used to obtain the `fdr` scores:

```
# X.value # this is the given data
# beta # = 11.093249 , obtained by running EM algorithm
# pi0 # 0.696794 , obtained by running EM algorithm

fdr_score <- pi0 / (pi0 + pi1*beta*(1 - X.value)^(beta - 1))
```

(d) We can now classify data using the criterion that a data point belongs to the first group if its `fdr` score exceeds 0.5 and that it belongs to the second group otherwise. We can then compare our classification result with the actual group information:

```
##Find the local fdr and compare it with the data

greater_than_half = function(x){
    if( x > 0.5)
        0
    else
        1
}

fdr_score <- pi0 / (pi0 + pi1*beta*(1 - X.value)^(beta - 1))
Z.guess <- sapply(fdr_score,greater_than_half)

falsely_classed <- sum(abs(Z.guess - X.group))   # =321
```

Thus only 321 out of 2000 got falsely classified (about 16%).

Question 5.2. (Continued from Problem 1.) It is known that the local fdr score can be written as

$$fdr_i(x_i) = \frac{\pi_0 f_0(x_i)}{f(x_i)}$$

where $f(x_i)$ is the marginal density of x_i. Assume that $\pi = 0.7$.

(a) Estimate $f(x_i)$ by using the kernel density estimation with Gaussian kernel and Silverman's h;

(b) Estimate the local `fdr` score;

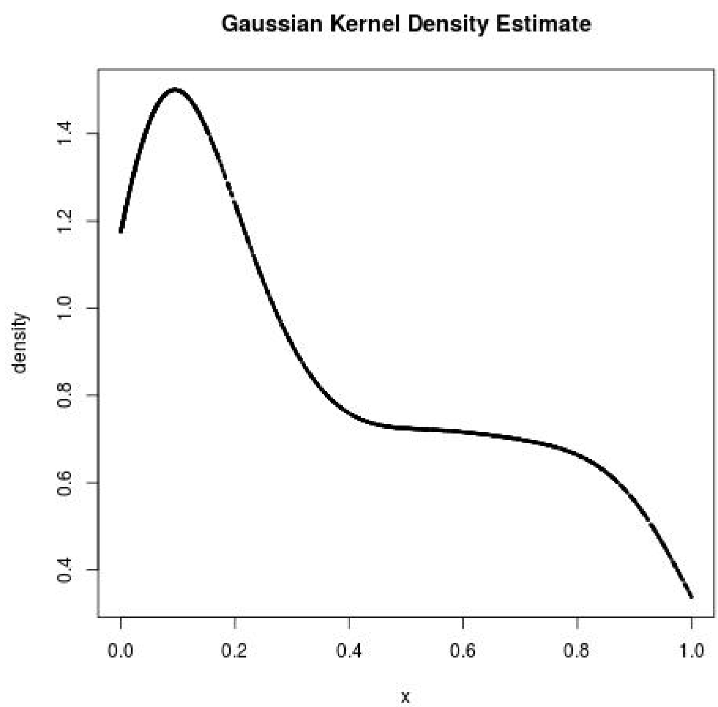
(c) Using the same rule as in 1(d), calculate the total number of falsely classified data;

(d) Choose the bandwidth using the maximum likelihood cross validation, repeat problem (a-c), what is the total number of falsely classified data?

(e) Which method works the best in terms of having the smallest classification error?

Answer:

(a) Using the Gaussian kernel, and Silverman's h, our density estimate looks like this:



(The code that generates this figure is submitted separately.)

(b) From part (a) we are able to estimate density at each observation x_i. We use the following formula to compute the `fdr` score:

$$\texttt{fdr}_i = \frac{0.7}{\text{value of the estimated density } f(x_i)}$$

The following code snippet does this computation in `R`:

```
pi0 <- 0.7
n <- length( X.value )
h <- 1.06 * sqrt( var(X.value) ) / (n^(1/5))

k_estimate = function(x){
    1/(h) * mean( dnorm( (x - X.value)/h, 0, h))
}

X.kdestimate <- pi0 / sapply(X.value,k_estimate)
```

(c) The total number of falsely classified observations using the score turns out to be 318:

```
X.kdestimate <- pi0 / sapply(X.value,k_estimate)
Z.guess.kde <- sapply(X.kdestimate,greater_than_half)

falsely_classed2 <- sum(abs(Z.guess.kde - X.group)) #answer: 318
```

(d) We now choose a different bandwidth h using the `kedd` library. Here h turns out to be higher than Silverman's h: `h.cv` = 0.1058126.

```
library(kedd)
h <- h.mlcv(X.value)$h    # output: h.cv = 0.1058126
```

Repeating steps (a) - (c) with this new h gives:

```
X.kdestimate.cv <- pi0 / sapply(X.value,k_estimate)
Z.guess.kde.cv <- sapply(X.kdestimate.cv,greater_than_half)

falsely_classed3 <- sum(abs(Z.guess.kde.cv - X.group)) # answer: 325
```

With the new bandwidth, we get a slightly higher number of falsely classified data.

The number of falsely classified here is 325.

(e) Which method works best? We expected cross-validation to work better, but it actually gave worse results: 325 falsely classified as opposed to 318 falsely classified with Silverman's h. The difference isn't great, but among the three estimates used here, Silverman's h gave us the lowest number of falsely classified.