Project-Stat 536

Hongjin, Mahedi and Swarnita

12/6/2019

Estimating the normal distribution parameters (μ and σ^2) from the censored data by using EM algorithm

Introduction

Often we face situation where the data are incomplete or censored but still our interest is to find the estimate of the parameter(s) from that incomplete data set. One such way to get the estimates of the parameter is the application of Expectation Maximization (EM) algorithm to the incomplete data set. In this work, we have applied EM algorithm to find the estimates of normal distribution parameters μ and σ^2 from a censored/incomplete data set.

Description of Data

The data set consists of $n = 45$ observations that comes from normal distribution with mean μ and variance σ^2. Suppose, $x_1, x_2......x_n$ observations are reported in the range $L_i \leq x_i \leq U_i$ where, L_i and U_i are known (but not necessarily finite). Here in this case, we have three types of observations, (i) the observation that is left censored, (ii) the observation that is right censored, and (iii) the observation that is confined in the interval (L_i, U_i).

Where,

$$A = \{i|U_i > L_i = -\infty\}$$
$$B = \{i|L_i < U_i = \infty\}$$
$$C = \{i|L_i < U_i, L_i \neq -\infty, U_i \neq \infty\}$$

Here, A idicates the observations that are left censored, B indicates the observations that are right censored and C indicates the observations that are confined between two limits.

This way, we have 14 observations that are left censored, 12 observations that are right censored, and rest 19 were observed at different levels of intervals.

Objective

The objective of this study is to find the maximum likelihood estimates for the normal distribution parameters (μ and σ^2) from the censored data by using Expectation Maximization (EM) algorithm.

Method

Let $x_1, x_2......x_n$ be a random sample from $N \sim (\mu, \sigma^2)$ and also let Z_i is the latent variable. We have the conditional distribution $X_i|Z_i = k \sim N(\mu_k, \sigma^2)$, so the marginal distribution of X_i is:

$$
\begin{aligned}
P(X_i = x) &= \sum_{k=1}^{k} P(Z_i = k) P(X_i = x | Z_i = k) \\
&= \sum_{k=1}^{k} \pi_k N(x; \mu_k, \sigma_k^2)
\end{aligned}
$$

Similarly, the joint probability of observations $x_1, x_2, ...x_n$ is therefore,

$$
P(X_i = x_1 ....X_n = x_n) = \prod_{i=1}^{n} \prod_{k=1}^{K} \pi_k N(x; \mu_k, \sigma_k^2)
$$

Now, with the help of EM algorithm we aims to obtain the maximum likelihood estimates of (μ_k, σ_k^2) given the data set of observations $x_1, x_2, ...x_n$

Intuitively, the latent variables Z_i should help us find the MLE's. We first compute the posterior distribution of Z_i,

$$
\begin{aligned}
P(Z_i = k | X_i) &= \frac{P(X_i | Z_i = k), P(Z_i = k)}{P(X_i)} \\
&= \frac{\pi_k N(x; \mu_k, \sigma_k^2)}{\sum_{k=1}^{K} \pi_k N(x; \mu_k, \sigma_k^2)} \\
&= \gamma_{Z_i}(K)
\end{aligned}
$$

Now, by deriving the log-likelihood function of the normal distribution distribution with respect to μ_k we get:

$$
\begin{aligned}
\hat{\mu}_k &= \frac{\sum_{i=1}^{n} \gamma_{Z_i}(k) x_i}{\sum_{i=1}^{n} \gamma_{Z_i}(k)} \\
&= \frac{1}{N_k} \sum_{i=1}^{n} \gamma_{Z_i}(k) x_i
\end{aligned}
$$

Therefore the $\hat{\mu}_k$ is the weighted average of data with weights $\gamma_{Z_i}(k)$.

Similarly, we can find the $\hat{\sigma}_k^2$ and we have,

$$
\hat{\sigma}_k^2 = \frac{1}{N_k} \sum_{i=1}^{n} \gamma_{Z_i}(k) (x_i - \mu_k)^2
$$

Steps for EM Algorithm

E-Step:

First we find the complete data likelihood
$Q(\theta | \hat{\theta}_m, X) = E[logL^c(\theta | X, Z)]$

M-Stem:

Then we maximize the complete data likelihood in the parameter and update
$Q(\theta | \hat{\theta}_m, X)$ in θ and update. $\hat{\theta}_{m+1} = argmaxQ(\theta | \hat{\theta}_m, X)$

Repeate this untill convergence

Results

The application of EM algorithm produces the following estimates for mean and standard deviation.
mean = 13.02413
SD = 1.010988

Statistical Algorithm

```
### Data
x_right_censor = c(12.25, 14.06, 12.44, 11.57, 14.76, 12.66,
                   13.62, 11.68, 13.81, 13.87, 12.63, 13.92)
length(x_right_censor)
```

```
## [1] 12
```

```
x_left_censor = c(14.03, 14.01, 12.69, 13.45, 14.89, 12.36, 13.08,
                  14.09, 12.73, 12.94, 14.91, 12.43, 14.42, 13.72)
length(x_left_censor)
```

```
## [1] 14
```

```
x_interval_censor = data.frame('lower' = c(11.96, 13.03, 11.61, 12.22, 11.92,
                                           12.96, 11.01, 13.79, 12.49, 13.14,
                                           11.83, 12.81, 13.46, 11.25, 11.58,
                                           13.69, 12.82, 12.78, 12.83),
                               'upper' = c(12.49, 13.28, 13.08, 12.84, 12.37,
                                           13.67, 11.85, 14.23, 13.31, 13.56,
                                           13.63, 13.43, 13.82, 12.04, 12.08,
                                           13.99, 12.98, 13.62, 14.74))

### Functions
Sx = function(x){
  s = dnorm(x)/(1-pnorm(x))
  return(s)
}

S1 = function(h, H){
  s1 = (dnorm(h)-dnorm(H))/(pnorm(H)-pnorm(h))
  return(s1)
}

S2 = function(h, H){
  s2 = -(h*dnorm(h) - H*dnorm(H))/(pnorm(H)-pnorm(h))
  return(s2)
}

Tx = function(x){
  t = Sx(x)*(Sx(x) - x)
  return(t)
}

T1 = function(h, H){
  t1 = S1(h, H)^2 + S2(h, H)
  return(t1)
}

### E step
wB = function(xleft, mu, sig){
  H = (xleft-mu)/sig
  e = mu - sig*Sx(-H)
  return(e)
}

wC = function(xright, mu, sig){
  h = (xright-mu)/sig
  e = mu + sig*Sx(h)
  return(e)
}

wD = function(xinterval, mu, sig){
  low = xinterval[,1]
  up = xinterval[,2]
  H = (up-mu)/sig
  h = (low-mu)/sig
  e = mu + sig*S1(h, H)
  return(e)
}


### M step
mu_hat = function(w1, w2, w3){
  n1 = length(w1); n2 = length(w2); n3 = length(w3)
  n = n1+n2+n3
  w = (sum(w1)+sum(w2)+sum(w3))/n
  return(w)
}

sig_hat = function(xleft, xright, xinterval, w1, w2, w3, mu_hat, sig_old){
  # numerator
  ns1 = sum((w1-mu_hat)^2); ns2 = sum((w2-mu_hat)^2); ns3 = sum((w3-mu_hat)^2)
  num_sum = ns1+ns2+ns3
  # H and h values
  H_b = (xleft - mu_hat)/sig_old
  h_c = (xright - mu_hat)/sig_old
  H_d_up = (xinterval[,2] - mu_hat)/sig_old
  h_d_low = (xinterval[,1] - mu_hat)/sig_old
  # denominator
  ds1= sum(Tx(-H_b)); ds2 = sum(Tx(h_c)); ds3 = sum(T1(h_d_low, H_d_up))
  de_sum = ds1+ds2+ds3
  # sigma square
  var = num_sum/de_sum
  return(sqrt(var))
}


### EM Implementation
err = 1; tol = 1e-5; its = 1; maxits = 10000
```

```
mu_old = 3; sig_old = 2

while(err>tol & its<maxits){
  wb = wB(x_left_censor, mu_old, sig_old)
  wc = wC(x_right_censor, mu_old, sig_old)
  wd = wD(x_interval_censor, mu_old, sig_old)
  mu_new = mu_hat(wb, wc, wd)
  sig_new = sig_hat(x_left_censor, x_right_censor, x_interval_censor,
                    wb, wc, wd,
                    mu_new, sig_old)
  err = max(abs(mu_new - mu_old), abs(sig_new - sig_old))
  its = its+1
  mu_old = mu_new; sig_old = sig_new
}

mu_new
```

```
## [1] 13.02413
```

```
sig_new
```

```
## [1] 1.010988
```

Contribution:

Hongjin,Mahedi and Swarnita read the paper and understood the steps to be done.
Hongjin wrote the code .
Swarnita and Mahedi wrote the code individually and matched with Hongjin’s.
Mahedi wrote the report.
Hongjin and Swarnita reviewed.