Homework 2016-03-21

Chuan Lu
13300180056

March 23, 2016

Problem 1.
Calculate the confidence interval with a confidence level of 0.95 of the samples given.

Proof. Firstly, the mean of the samples $\overline{x} = \frac{1}{10}\sum_i x_i = 10.05$, the variance of the samples $\sigma = \frac{1}{10}\sum_i (x_i - \overline{x})^2 = 0.0525$. We can then check from the t-distribution table that the confidence interval is $[\overline{X} - t_{\alpha/2}(n-1)\frac{\sigma}{\sqrt{n}}, \overline{X} + t_{\alpha/2}(n-1)\frac{\sigma}{\sqrt{n}}] = [9.9854, 10.1146]$ $\square$

Problem 2.
If the rate of abnormality in this area is below the average with the information provided.

Proof. $P(\text{Only one person is of abnormality}|\text{The rate of abnormality is } 0.01) = \binom{400}{1} * (1-0.01)^{399} * 0.01 = 0.0725 > 0.05$, which implies that this phenomenon is just possible, hence the rate of abnormality in this area can NOT be seen as below the average. $\square$

Problem 3.
Derive out the EM algorithm of a distribution mixed by three normal distributions.

Proof. Let

$$z_{ij} = \begin{cases} 1 & X_i \text{ belongs to the } j^{th} \text{ distribution,} \\ 0 & \text{others,} \end{cases}$$

and $P(z_i = 1) = t_i$, where $\sum_{i=1}^{3} t_i = 1$.
Then the likelihood function is $L(\theta; X, Z) = \prod_{i=1}^{n}\prod_{j=1}^{3} t_j \frac{1}{(\sqrt{2\pi})^n \sqrt{(det(\Sigma_j))}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}$, the log-likelihood of which is $l(\theta; x, z) = \sum_{i=1}^{n}\sum_{j=1}^{3} z_{ij}[log(t_j) - \frac{d}{2}log(2\pi) - \frac{1}{2}(x_i - \mu_j)^T \Sigma^{-1}(x-\mu)]$.
E-step:
Assume there are a vector of θ_n, then $Q(\theta|\theta_m) = E(l(\theta; x, z)) = \sum_{i=1}^{n}\sum_{j=1}^{3} T_{i,j}^{(m)}[log(t_j) - \frac{d}{2}log(2\pi) - \frac{1}{2}(x_i - \mu_j)^T \Sigma^{-1}(x-\mu)$
in which $T_{ij}^{(m)} = P(z_{ij} = 1 | X_i = x_i; \theta_m) = \frac{t_j^{(m)} f(x_i; \mu_j^{(m)}, \Sigma_j^{(m)})}{\sum_{k=1}^{3} t_k^{(m)} f(x_i; \mu_k^{(m)}, \Sigma_k^{(m)})}, j = 1, 2, 3.$
M-step:
For t_{m+1},
$t_{m+1} = argmax_t Q(\theta|\theta_m) = argmax_t \left([\sum_{i=1}^{n} T_{i,1}^{(m)}]log(t_1) + [\sum_{i=1}^{n} T_{i,2}^{(m)}]log(t_2) + [\sum_{i=1}^{n} T_{i,3}^{(m)}]log(t_3) \right)$, so $t_j^{(m+1)} = \frac{1}{n}\sum_{i=1}^{n} T_{i,j}^{(m)}$.
For μ_{m+1}, σ_{m+1},
$(\mu_{m+1}, \sigma_{m+1}) = argmax_{\mu,\sigma} Q(\theta|\theta_m)$, so $\mu_j^{(m+1)} = \frac{\sum_{i=1}^{n} T_{i,j}^{(m)} x_i}{\sum_{i=1}^{n} T_{i,j}^{(m)}}$, and $\sigma_j^{(m+1)} = \frac{\sum_{i=1}^{n} T_{i,j}^{(m)} (x_i - \mu_j^{(m)})(x_i - \mu_j^{(m)})^T}{\sum_{i=1}^{n} T_{i,j}^{(m)}}$ $\square$

Problem 4.
Estimate the arguments with the data given.

Proof. **0.1 The code is shown as follows.**

```
source('em_func.R')
em = function(x, dim, mu, sigma, t, maxit = 100, error = 1e-6) {
  # mu, sigma, t are listsa containing init data
  # Init
  n = nrow(x)
  T = matrix(rep(0, n*dim), ncol=dim)

  for(step in 1:maxit) {

    # E-step
    for(j in 1:dim) {
      for(i in 1:n) {
        # print(x[i, ])
        T[i, j] = em_FUNC(x[i, ], mu[[j]], sigma[[j]]) * t(j)
      }
    }
    T = T/rowSums(T)

    sigma0 = sigma
    t0 = t
    mu0 = mu

    # M-step
    for(j in 1:dim) {
      v1 = sum(T[, j])
      v2 = matrix(rep(0, ncol(x)), nrow=1)
      for(i in 1:n) {
        v2 = v2 + (T[i, j] * x[i, ])
      }
      mu[[j]] = v2 / v1
      t[j] = v1 / n
      v3 = matrix(rep(0, ncol(x)^2), nrow=ncol(x))
      for(i in 1:n) {
        temp1 = matrix(x[i, ])
        temp2 = matrix(mu[[j]])
        delta = temp1 - temp2
        v3 = v3 + (T[i, j] * delta %*% t(delta))
      }
      sigma[[j]] = (v3/v1)
    }
    mu_sum = 0
    sigma_sum = 0
    for(i in 1:length(mu)) {
      mu_sum = mu_sum + sum(abs(mu0[[i]] - mu[[i]]))
    }
    for(i in 1:length(sigma)) {
      sigma_sum = sigma_sum + sum(abs(sigma0[[i]] - sigma[[i]]))
    }
    # Check if converged
    if(mu_sum < error & sigma_sum < error & sum(abs(t-t0)) < error)
      break
  }
```

```
  returnlist = list(mu, sigma, t)

  return(returnlist)
}
```

```
em_FUNC <- function(x, mu, sigma) {
  n = ncol(t(matrix(x)))
  res = 1/((sqrt(2*pi))^n * sqrt(abs(det(sigma)))) * exp(-1/2 * (x-mu)%*%solve(sigma)%*%t(x
  return (res)
}
```

```
# Exercise 4
source('EM.R')
data = read.csv('Data1.csv')
x1 = data['V1']
x2 = data['V2']
x = data.frame(x1, x2)
x = data.matrix(x)
dim = 3
n = 2
mu = list(t(matrix(runif(n))), t(matrix(runif(n))), t(matrix(runif(n))))
sigma = list(matrix(runif(n*n), ncol = n), matrix(runif(n*n), ncol = n), matrix(runif(n*n)
t = c(0.1, 0.7, 0.2)
rlist = em(x, dim, mu, sigma, t)
print(rlist[[1]])
print(rlist[[2]])
print(rlist[[3]])
```

0.2 The result is shown as follows, in which rlist[[1] is mu, rlist[[2]] is sigma, rlist[[3]] is t.

```
> print(rlist[[1]])
[[1]]
           [,1]      [,2]
[1,] -2.274075 -3.209265

[[2]]
          [,1]      [,2]
[1,] 5.271368 -2.079697

[[3]]
           [,1]       [,2]
[1,] 0.8930116 -0.6144073

> print(rlist[[2]])
[[1]]
           [,1]       [,2]
[1,]  0.6014583 -0.1520107
[2,] -0.1520107  0.8660930

[[2]]
             [,1]         [,2]
[1,]  1.577730561 -0.009383941
[2,] -0.009383941  1.104548710
```

```

[[3]]
            [,1]      [,2]
[1,] 4.439728 3.414217
[2,] 3.414217 4.359960

> print(rlist[[3]])
[1] 0.1783595 0.4624902 0.3591504
```

□

Problem 5.
Estimate the average height of men and women with the data givem.

Proof. **0.3 The code is shown as follows, and the function 'EM' is given within the answer to the former exercise.**

```
# Exercise5
source('EM.R')
x = matrix(c(171, 174, 159, 176, 164, 169, 170, 173, 159,
             172, 166, 175, 161, 186, 160, 168, 166, 174,
             159, 178, 165, 189, 164, 168, 165, 185, 160,
             175, 172, 168, 167, 171, 160, 174, 168, 174,
             167, 175, 162, 177))
dim = 2
mu = list(matrix(c(163)), matrix(c(176)))
sigma = list(matrix(runif(1)), matrix(runif(1)))
t = c(0.2, 0.8)
rlist = em(x, dim, mu, sigma, t)
print(rlist[[1]])
print(rlist[[2]])
print(rlist[[3]])
```

0.4 The result is shown as follows, in which rlist[[1]] is mu, rlist[[2]] is sigma, rlist[[3]] is t.

```
> print(rlist[[1]])
[[1]]
            [,1]
[1,] 163.6147

[[2]]
            [,1]
[1,] 172.5814

> print(rlist[[2]])
[[1]]
          [,1]
[1,] 14.6682

[[2]]
            [,1]
[1,] 46.11688

> print(rlist[[3]])
```

```
[1] 0.3269247 0.6730753
```

□