# HW 11

## Problem 1

Since $\mathrm{E}_{\mathrm{old}} \log p(\gamma|\phi^{\mathrm{old}}, y)$ is a constant to $\phi$, we have

$$\begin{aligned}
&\operatorname{argmax}_\phi \mathrm{E}_{\mathrm{old}} \log p(\gamma|\phi, y) \\
&= \operatorname{argmin}_\phi - \mathrm{E}_{\mathrm{old}} \log p(\gamma|\phi, y) \\
&= \operatorname{argmin}_\phi \left\{ \mathrm{E}_{\mathrm{old}}(-\log p(\gamma|\phi, y) + \log p(\gamma|\phi^{\mathrm{old}}, y)) \right\} \\
&= \operatorname{argmin}_\phi \mathrm{E}_{\mathrm{old}} \log \frac{p(\gamma|\phi^{\mathrm{old}}, y)}{p(\gamma|\phi, y)}.
\end{aligned}$$

So it is equivalent to minimizing

$$\mathrm{E}_{\mathrm{old}} \log \frac{p(\gamma|\phi^{\mathrm{old}}, y)}{p(\gamma|\phi, y)},$$

the KL divergence between $p(\gamma|\phi^{\mathrm{old}}, y)$ and $p(\gamma|\phi, y)$. Since $-\log(x)$ is convex, by Jensen's inequality we have

$$\begin{aligned}
\mathrm{E}_{\mathrm{old}} \log \frac{p(\gamma|\phi^{\mathrm{old}}, y)}{p(\gamma|\phi, y)} &= \mathrm{E}_{\mathrm{old}} - \log \frac{p(\gamma|\phi, y)}{p(\gamma|\phi^{\mathrm{old}}, y)} \\
&\geq -\log \mathrm{E}_{\mathrm{old}} \frac{p(\gamma|\phi, y)}{p(\gamma|\phi^{\mathrm{old}}, y)} \\
&= -\log(1) = 0.
\end{aligned}$$

The minimum 0 is achieved when $\phi = \phi^{\mathrm{old}}$, because if $\phi = \phi^{\mathrm{old}}$ then

$$\mathrm{E}_{\mathrm{old}} \log \frac{p(\gamma|\phi^{\mathrm{old}}, y)}{p(\gamma|\phi, y)} = \mathrm{E}_{\mathrm{old}} \log(1) = 0.$$

This proves that the maximizer of $\mathrm{E}_{\mathrm{old}} \log p(\gamma|\phi, y)$ is $\phi^{\mathrm{old}}$.

## Problem 2

The log posterior is

$$\begin{aligned}
\ell &= \log p(\theta, \mu, \log\sigma, \log\tau|y) \\
&= -n\log\sigma - (J-1)\log\tau - \frac{1}{2\tau^2}\sum_{j=1}^{J}(\theta_j - \mu)^2 - \frac{1}{2\sigma^2}\sum_{j=1}^{J}\sum_{i=1}^{n_j}(y_{ij} - \theta_j)^2 + \text{constant}.
\end{aligned}$$

Taking derivative of with respect to $\mu$ we have

$$\frac{\partial \mathrm{E}_{\mathrm{old}}\ell}{\partial\mu} = \frac{1}{\tau^2}\sum_{j=1}^{J}\mathrm{E}_{\mathrm{old}}(\theta_j - \mu).$$

Also, the second order derivative is negative (formula omitted). Setting the derivative to be 0 we have $\mu^{\text{new}} = \frac{1}{J}\sum_{j=1}^{J}\hat{\theta}_j$.

Similarly,

$$\frac{\partial \mathrm{E}_{\text{old}}\ell}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3}\sum_{j=1}^{J}\sum_{i=1}^{n_j}\mathrm{E}_{\text{old}}(y_{ij}-\theta_j)^2,$$

and

$$\frac{\partial \mathrm{E}_{\text{old}}\ell}{\partial \sigma}\Big|_{\mu=\mu^{\text{new}}} = -\frac{J-1}{\tau} + \frac{1}{\tau^3}\sum_{j=1}^{J}\mathrm{E}_{\text{old}}(\theta_j-\mu^{\text{new}})^2.$$

Setting the derivatives to be 0 we have

$$\sigma^{\text{new}} = \left(\frac{1}{n}\sum_{j=1}^{J}\sum_{i=1}^{n_j}\mathrm{E}_{\text{old}}(y_{ij}-\theta_j)^2\right)^{1/2} = \left(\frac{1}{n}\sum_{j=1}^{J}\sum_{i=1}^{n_j}\left((y_{ij}-\hat{\theta}_j)^2 + V_{\theta_j}\right)\right)^{1/2},$$

and

$$\tau^{\text{new}} = \left(\frac{1}{J-1}\sum_{j=1}^{J}\mathrm{E}_{\text{old}}(\theta_j-\mu^{\text{new}})^2\right)^{1/2} = \left(\frac{1}{J-1}\sum_{j=1}^{J}\left((\hat{\theta}_j-\mu^{\text{new}})^2 + V_{\theta_j}\right)\right)^{1/2}.$$

## Simulate fake data with $n_j = 10$ and $J = 4$

```
set.seed(0)
mu <- 0
sigma <- 1
tau <- 2
J <- 4
nj <- 10
n <- J*nj

theta <- rnorm(J, mu, tau^2)
y <- t(matrix(rnorm(n, theta, sigma^2), J, nj))
```

## EM algorithem

```
iter <- 10

par <- matrix(0, iter+1, 3+J+1)
par[1, ] <- c(mean(y), sd(y), sd(y), colMeans(y), 1/(1/sd(y)^2 + nj/sd(y)^2))

obj <- rep(NA, iter)


for (i in 1:iter) {
  tau <- par[i, 3]
  sigma <- par[i, 2]
  mu <- par[i, 1]
```

```r
  # i-th E step (solved)
  V_theta <- 1 / (1/tau^2 + nj/sigma^2) # in R
  theta_hat <- (1/tau^2 * mu + nj/sigma^2 * colMeans(y)) * V_theta # in R^J

  # log posterior evaluated at (i-1)-th iteration
  # !NOT the log posterior of i-th iteration
  obj[i] <- log(tau) +
  sum(dnorm(theta_hat, mean=mu, sd=tau, log=T)) +
  sum(dnorm(t(y), mean=theta_hat, sd=sigma, log=T)) +
  1/2*J*log(V_theta)
  print(obj[i])

  # i-th M step (solved)
  mu_new <- mean(theta_hat)
  sigma_new <- sqrt( V_theta + sum((t(t(y) - theta_hat))^2)/n )
  tau_new <- sqrt( J/(J-1) * V_theta + sum((theta_hat - mu_new)^2) / (J-1))

  par[i+1, ] <- c(mu_new, sigma_new, tau_new, theta_hat, V_theta)

}
```

```
## [1] -90.72674
## [1] -66.79836
## [1] -63.27169
## [1] -63.18976
## [1] -63.18885
## [1] -63.18884
## [1] -63.18884
## [1] -63.18884
## [1] -63.18884
## [1] -63.18884
```

It is converged after several steps. The log posterior increases in each step so the program should be correct. The final results are:

```r
cat(' Posterior modes are:\n',
    'mu =', par[iter+1, 1],
    '\n sigma =', par[iter+1, 2],
    '\n tau =', par[iter+1, 3],
    '\n The log posterior value is:', obj[iter])
```

```
##  Posterior modes are:
##  mu = 3.531257
##  sigma = 0.8732692
##  tau = 3.226493
##  The log posterior value is: -63.18884
```

# Problem 3

```r
swaps <- function(fill) {

# round(fill*10000) = fill

  if (round(fill*10000) != fill*10000) {
    stop('.')
  }

  A <- matrix(sample(c(rep(1, fill*100*100), rep(0, (1-fill)*100*100))),
              100, 100)
  C <- colSums(A)
  R <- rowSums(A)

  iter <- 1e4

  Accept <- rep(TRUE, iter)

  indice_r <- seq(1, 100, 1)[R != 0]
  indice_c <- seq(1, 100, 1)[C != 0]

  checkboard <- list(matrix(c(1, 0, 0, 1), 2, 2), matrix(c(0, 1, 1, 0), 2, 2))

  for (i in 1:iter) {
    .row1 <- sample(indice_r, 1)
    .col1 <- sample(indice_c, 1)
    if (A[.row1, .col1] == 1) {
      .col2 <- sample(indice_c[A[.row1, indice_c] == 0], 1)
      .row2 <- sample(indice_r[A[indice_r, .col2] == 1], 1)
    } else {
      .row2 <- sample(indice_r[A[indice_r, .col1] == 1], 1)
      .col2 <- sample(indice_c[A[.row2, indice_c] == 0], 1)
    }

    if ( identical(A[c(.row1, .row2), c(.col1, .col2)], checkboard[[1]]) ) {
      A[c(.row1, .row2), c(.col1, .col2)] <- checkboard[[2]]
    } else if ( identical(A[c(.row1, .row2), c(.col1, .col2)], checkboard[[2]]) ) {
      A[c(.row1, .row2), c(.col1, .col2)] <- checkboard[[1]]
    } else {
      Accept[i] <- FALSE
    }
  }
  return(sum(Accept))
}
```

```r
swaps(.05)
```

```
## [1] 896
```

```r
swaps(.10)
```

```
## [1] 1828
```

```r
swaps(.5)
```

```
## [1] 5055
```

The reported number of swaps are consistent with Table 4 in the paper.