STAD80: Assignment 6

Vladislav Trukhin

Due: March 30th, 2022

Contents

Question 1

```r
# initialize data directory
data_dir <- "mnist-data"
dir.create(data_dir, recursive = TRUE, showWarnings = FALSE)

# download the MNIST data sets, and read them into R
sources <- list(

  train = list(
```

```r
    x = "https://storage.googleapis.com/cvdf-datasets/mnist/train-images-idx3-ubyte.gz",
    y = "https://storage.googleapis.com/cvdf-datasets/mnist/train-labels-idx1-ubyte.gz"
  ),

  test = list(
    x = "https://storage.googleapis.com/cvdf-datasets/mnist/t10k-images-idx3-ubyte.gz",
    y = "https://storage.googleapis.com/cvdf-datasets/mnist/t10k-labels-idx1-ubyte.gz"
  )

)

# read an MNIST file (encoded in IDX format)
read_idx <- function(file) {

  # create binary connection to file
  conn <- gzfile(file, open = "rb")
  on.exit(close(conn), add = TRUE)

  # read the magic number as sequence of 4 bytes
  magic <- readBin(conn, what = "raw", n = 4, endian = "big")
  ndims <- as.integer(magic[[4]])

  # read the dimensions (32-bit integers)
  dims <- readBin(conn, what = "integer", n = ndims, endian = "big")

  # read the rest in as a raw vector
  data <- readBin(conn, what = "raw", n = prod(dims), endian = "big")

  # convert to an integer vecto
  converted <- as.integer(data)

  # return plain vector for 1-dim array
  if (length(dims) == 1)
    return(converted)

  # wrap 3D data into matrix
  matrix(converted, nrow = dims[1], ncol = prod(dims[-1]), byrow = TRUE)
}

mnist <- rapply(sources, classes = "character", how = "list", function(url) {

  # download + extract the file at the URL
  target <- file.path(data_dir, basename(url))
  if (!file.exists(target))
    download.file(url, target)

  # read the IDX file
  read_idx(target)

})
```

```r
# Additional preprocessing

# convert training data intensities to 0-1 range
```

```
mnist$train$x <- mnist$train$x / 255
mnist$test$x <- mnist$test$x / 255

# Only cluster digits 0-4
ix_train <- mnist$train$y == 0 | mnist$train$y == 1 | mnist$train$y == 2 | mnist$train$y == 3 | mnist$t
ix_test <- mnist$test$y == 0 | mnist$test$y == 1 | mnist$test$y == 2 | mnist$test$y == 3 | mnist$test$y
mnist$train$x <- mnist$train$x[ix_train,]
mnist$train$y <- mnist$train$y[ix_train]
mnist$test$x <- mnist$test$x[ix_test,]
mnist$test$y <- mnist$test$y[ix_test]

# 1/4 size train
matrix <- c()
for (n in 1:dim(mnist$train$x)[1]) {
  im <- t(matrix(mnist$train$x[n,], ncol=28, nrow=28))
  list <- c()
  for (i in 1:14) {
    for (j in 1:14) {
      list <- cbind(list, mean(im[(2*i-1):(2*i), (2*j-1):(2*j)]))
    }
  }
  matrix <- rbind(matrix, list)
}
mnist$train$x <- matrix

# 1/4 size test
matrix <- c()
for (n in 1:dim(mnist$test$x)[1]) {
  im <- t(matrix(mnist$test$x[n,], ncol=28, nrow=28))
  list <- c()
  for (i in 1:14) {
    for (j in 1:14) {
      list <- cbind(list, mean(im[(2*i-1):(2*i), (2*j-1):(2*j)]))
    }
  }
  matrix <- rbind(matrix, list)
}
mnist$test$x <- matrix
```

```
N <- dim(mnist$train$x)[1]
D <- dim(mnist$train$x)[2]
K <- 5
```

Question 1.1

```
par(mfrow=c(10, 7))
set.seed(10)
mu = matrix(runif(D*K), nrow = D, ncol = K)
prev_loss <- 0
for (i in 1:100) {
  # Assignment
  closest <- list()
  loss <- 0
```

```r
  for (n in 1:N) {
    closest[n] <- which.min(colSums((mnist$train$x[n,] - mu)^2))
    loss <- loss + min(colSums((mnist$train$x[n,] - mu)^2))
  }

  # Update
  for (k in 1:K) {
    mu[,k] <- colMeans(rbind(mnist$train$x[closest == k,], rep(0, 196)))
  }

  # Current iteration loss
  message("Iteration: ", i, " Loss: ", loss)

  # Check for convergence
  if (abs(prev_loss - loss) < 0.0001) {
    break
  }
  prev_loss <- loss
}
```

```
## Iteration: 1 Loss: 1528387.54278973

## Iteration: 2 Loss: 302391.106983158

## Iteration: 3 Loss: 261158.550252353

## Iteration: 4 Loss: 244133.077778585

## Iteration: 5 Loss: 238663.608216232

## Iteration: 6 Loss: 235509.779660883

## Iteration: 7 Loss: 232529.208513017

## Iteration: 8 Loss: 229834.442619395

## Iteration: 9 Loss: 228370.21412767

## Iteration: 10 Loss: 227624.413714981

## Iteration: 11 Loss: 227276.805691325

## Iteration: 12 Loss: 227142.463723411

## Iteration: 13 Loss: 227098.293960547

## Iteration: 14 Loss: 227081.248184545

## Iteration: 15 Loss: 227076.938393217

## Iteration: 16 Loss: 227075.485606832

## Iteration: 17 Loss: 227074.990749326

## Iteration: 18 Loss: 227074.805799378

## Iteration: 19 Loss: 227074.733694462

## Iteration: 20 Loss: 227074.710339601

## Iteration: 21 Loss: 227074.672974529

## Iteration: 22 Loss: 227074.653392452

## Iteration: 23 Loss: 227074.636174259
```

```
## Iteration: 24 Loss: 227074.634472423
```

```
## Iteration: 25 Loss: 227074.634472423
```

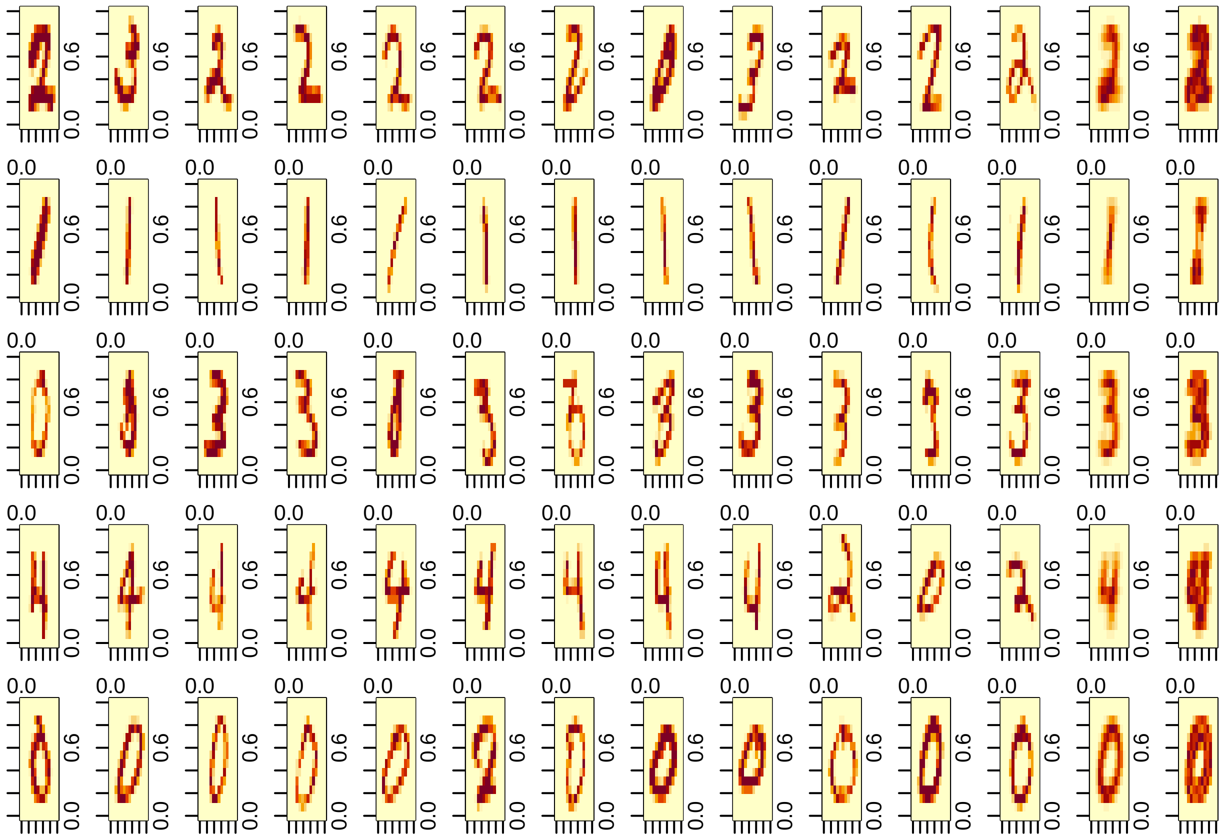
```r
par(mar=c(1,1,1,1))
par(mfrow=c(K, 14))
for (k in 1:K) {
  class <- mnist$train$x[closest == k,]
  number <- names(sort(-table(mnist$train$y[closest == k])))[1]
  acc <- sum(mnist$train$y[closest == k] == number) / length(mnist$train$y[closest == k])

  # Plot 1
  sampled <- mnist$train$x[sample(which(closest == k), 12),]
  for (i in 1:12) {
    image(matrix(sampled[i,], ncol=14, nrow=14)[,14:1])
  }

  # Plot 2
  image(matrix(mu[,k], ncol=14, nrow=14)[,14:1])

  # Plot 3
  var <- colSums(class^2) / dim(class)[1] - mu[,k]^2
  var <- var / max(var) * 255
  var[var < 51] <- 0
  image(matrix(var, ncol=14, nrow=14)[,14:1])

  # Plot 4
  print(sprintf("Cluster %d", as.integer(number)))
  # Number with True Label in Cluster vs Number with True Label
  print(c(sum(mnist$train$y[closest == k] == number), length(mnist$train$y[closest == k])))
  # Accuracy Rate
  print(acc)
  # Misclassification Rate
  print(1-acc)
}
```

```
## [1] "Cluster 2"
## [1] 4402 5168
## [1] 0.8517802
## [1] 0.1482198
```

```
## [1] "Cluster 1"
## [1] 6607 7923
## [1] 0.8339013
## [1] 0.1660987
```

```
## [1] "Cluster 3"
## [1] 5129 6007
## [1] 0.8538372
## [1] 0.1461628
```

```
## [1] "Cluster 4"
## [1] 5559 6156
## [1] 0.9030214
## [1] 0.09697856
```



```
## [1] "Cluster 0"
## [1] 5226 5342
## [1] 0.9782853
## [1] 0.02171471
```

Question 1.2

Question 1.2.I

a) $p(Z_i = j) = \eta_j$

b) $p(Z_i = j|x_i) = \frac{p(x_i|Z_i=j)p(Z_i=j)}{p(x_i)} = \frac{p(x_i|Z_i=j)\eta_j}{\Sigma_{j=1}^{k} p(x_i|Z_i=j)\eta_j}$

Question 1.2.II

c) $\ell(\theta) = \Sigma_{i=1}^{n} \log \Sigma_{j=1}^{k} \gamma_{ij} \frac{p_\theta(x_i, Z_i=j)}{\gamma_{ij}}$

$= \Sigma_{i=1}^{n} \log E_{\gamma_{ij}} \frac{p_\theta(x_i, Z_i=j)}{\gamma_{ij}}$

$\geq \Sigma_{i=1}^{n} E_{\gamma_{ij}} \log \frac{p_\theta(x_i, Z_i=j)}{\gamma_{ij}}$ By Jensen's Inequality

$= \Sigma_{i=1}^{n} \Sigma_{j=1}^{k} \gamma_{ij} \log \frac{p_\theta(x_i, Z_i=j)}{\gamma_{ij}}$

d) $F(\gamma, \theta^{old}) = \Sigma_{i=1}^{n} \Sigma_{j=1}^{k} \gamma_{ij} \log \frac{p_{\theta^{old}}(x_i, Z_i=j)}{\gamma_{ij}}$

$= \Sigma_{i=1}^{n} \Sigma_{j=1}^{k} p_{\theta^{old}}(Z_i = j|x_i) \log \frac{p_{\theta^{old}}(x_i, Z_i=j)}{p_{\theta^{old}}(Z_i=j|x_i)}$

$= \Sigma_{i=1}^{n} \Sigma_{j=1}^{k} p_{\theta^{old}}(Z_i = j|x_i) \log p_{\theta^{old}}(x_i)$

$= \Sigma_{i=1}^n \log p_{\theta^{old}}(x_i) \Sigma_{j=1}^k p_{\theta^{old}}(Z_i = j|x_i)$

$= \Sigma_{i=1}^n \log p_{\theta^{old}}(x_i)$

$= \Sigma_{i=1}^n \log \Sigma_{j=1}^k p_{\theta^{old}}(x_i, Z_i = j)$

$= \Sigma_{i=1}^n \log \Sigma_{j=1}^k p_{\theta^{old}}(Z_i = j|x_i) \frac{p_{\theta^{old}}(x_i, Z_i=j)}{p_{\theta^{old}}(Z_i=j|x_i)}$

$= \Sigma_{i=1}^n \log \Sigma_{j=1}^k \gamma_{ij} \frac{p_{\theta^{old}}(x_i, Z_i=j)}{\gamma_{ij}} = \ell(\theta^{old})$

Question 1.2.III

$argmax_\theta F_{\theta^{old}}(\theta) = argmax_\theta \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log \frac{p_\theta(x_i, Z_i=j)}{\gamma_{ij}} - \Sigma_{i=1}^n \Sigma_{j=1}^k \eta_j + N$

$= argmax_\theta \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log p_\theta(x_i, Z_i = j) - \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log \gamma_{ij} - \Sigma_{i=1}^n \Sigma_{j=1}^k \eta_j + N$

$= argmax_\theta \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log p_\theta(x_i|Z_i = j) + \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log p(Z_i = j) - \Sigma_{i=1}^n \Sigma_{j=1}^k \eta_j + N$

$= argmax_\theta \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log(\frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} exp(-\frac{1}{2}(x_i - \mu_j)^\top \Sigma_j^{-1}(x_i - \mu_j))) + \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log \eta_j - \Sigma_{i=1}^n \Sigma_{j=1}^k \eta_j + N$

$= argmax_\theta \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij}(-\frac{d}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma_j^{-1}| - \frac{1}{2}(x_i - \mu_j)^\top \Sigma_j^{-1}(x_i - \mu_j)) + \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log \eta_j - \Sigma_{i=1}^n \Sigma_{j=1}^k \eta_j + N$

$= argmax_\theta \frac{1}{2} \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log|\Sigma_j^{-1}| - \frac{1}{2} \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij}(x_i - \mu_j)^\top \Sigma_j^{-1}(x_i - \mu_j) + \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log \eta_j - \Sigma_{i=1}^n \Sigma_{j=1}^k \eta_j + N$

e)

For σ_j^2 :

$\frac{\partial}{\partial \sigma_j^{-2}} \frac{1}{2} \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log|\Sigma_j^{-1}| - \frac{1}{2} \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij}(x_i - \mu_j)^\top \Sigma_j^{-1}(x_i - \mu_j) + \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log \eta_j - \Sigma_{i=1}^n \Sigma_{j=1}^k \eta_j + N$

$= \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} \frac{\partial}{\partial \sigma_j^{-2}} \log|\sigma_j^{-2} I| - \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} \frac{\partial}{\partial \sigma_j^{-2}} (x_i - \mu_j)^\top \sigma_j^{-2} I (x_i - \mu_j)$

$= \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} \sigma_j^2 I - \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij}(x_i - \mu_j)^\top (x_i - \mu_j)$

$= \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} \sigma_j^2 - \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij}(x_i - \mu_j)^\top (x_i - \mu_j) = 0$

$\Rightarrow \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} \sigma_j^2 = \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij}(x_i - \mu_j)^\top (x_i - \mu_j)$

$\Rightarrow \sigma_j^2 \Sigma_{i=1}^n \gamma_{ij} = \Sigma_{i=1}^n \gamma_{ij}(x_i - \mu_j)^\top (x_i - \mu_j)$

$\Rightarrow \sigma_j^2 = \frac{\Sigma_{i=1}^n \gamma_{ij}(x_i - \mu_j)^\top (x_i - \mu_j)}{\Sigma_{i=1}^n \gamma_{ij}}$

For μ_j :

$\frac{\partial}{\partial \mu_j} \frac{1}{2} \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log|\Sigma_j^{-1}| - \frac{1}{2} \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij}(x_i - \mu_j)^\top \Sigma_j^{-1}(x_i - \mu_j) + \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log \eta_j - \Sigma_{i=1}^n \Sigma_{j=1}^k \eta_j + N$

$= -\frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} \frac{\partial}{\partial \mu_j} (x_i - \mu_j)^\top \Sigma_j^{-1}(x_i - \mu_j)$

$= \Sigma_{i=1}^n \gamma_{ij}(x_i - \mu_j)^\top \Sigma_j^{-1} = 0$

$\Rightarrow \Sigma_{i=1}^n \gamma_{ij}(x_i - \mu_j) = 0$

$\Rightarrow \Sigma_{i=1}^n \gamma_{ij} \mu_j = \Sigma_{i=1}^n \gamma_{ij} x_i$

$\Rightarrow \mu_j = \frac{\Sigma_{i=1}^n \gamma_{ij} x_i}{\Sigma_{i=1}^n \gamma_{ij}}$

For η_j :

$\frac{\partial}{\partial \eta_j} \frac{1}{2} \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log|\Sigma_j^{-1}| - \frac{1}{2} \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij}(x_i - \mu_j)^\top \Sigma_j^{-1}(x_i - \mu_j) + \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log \eta_j - \Sigma_{i=1}^n \Sigma_{j=1}^k \eta_j + N$

$\Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \frac{1}{\eta_j} - \Sigma_{i=1}^n \Sigma_{j=1}^k 1 = 0$

$\Rightarrow \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \frac{1}{\eta_j} = \Sigma_{i=1}^n \Sigma_{j=1}^k 1$

$\Rightarrow \Sigma_{i=1}^n \gamma_{ij} \frac{1}{\eta_j} = \Sigma_{i=1}^n 1$

$\Rightarrow \frac{1}{\eta_j} = \frac{N}{\Sigma_{i=1}^n \gamma_{ij}}$

$\Rightarrow \eta_j = \frac{\Sigma_{i=1}^n \gamma_{ij}}{N}$

f)

For σ_{jm}^2 :

$\frac{\partial}{\partial \sigma_{jm}^{-2}} \frac{1}{2} \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log |\Sigma_j^{-1}| - \frac{1}{2} \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} (x_i - \mu_j)^\top \Sigma_j^{-1} (x_i - \mu_j) + \Sigma_{i=1}^n \Sigma_{j=1}^k \gamma_{ij} \log \eta_j - \Sigma_{i=1}^n \Sigma_{j=1}^k \eta_j + N$

$= \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} \frac{\partial}{\partial \sigma_{jm}^{-2}} \log |[\sigma_{j1}^{-2}, ..., \sigma_{jd}^{-2}] I| - \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} \frac{\partial}{\partial \sigma_{jm}^{-2}} (x_i - \mu_j)^\top [\sigma_{j1}^{-2}, ..., \sigma_{jd}^{-2}] I (x_i - \mu_j)$

$= \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} [0, ..., \sigma_{jm}^2, ..., 0] I - \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} [0, ..., 1, ..., 0] I (x_i - \mu_j)(x_i - \mu_j)^\top$

$= \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} \sigma_{jm}^2 - \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} (x_{im} - \mu_{jm})^2 = 0$

$\Rightarrow \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} \sigma_{jm}^2 = \frac{1}{2} \Sigma_{i=1}^n \gamma_{ij} (x_{im} - \mu_{jm})^2$

$\Rightarrow \sigma_{jm}^2 = \frac{\Sigma_{i=1}^n \gamma_{ij} (x_{im} - \mu_{jm})^2}{\Sigma_{i=1}^n \gamma_{ij}}$

η_j and μ_j follow from e)

Question 1.2.IV

```
#NOTE: ALGORITHM DOES NOT WORK
mu = matrix(runif(196*5), nrow = 196, ncol = 5)
sigma = rep(0.5, K)
pi = rep(1, K) / K

prev_log_likelihood <- 0
for (i in 1:100) {
  # E-Step
  resp = matrix(0, nrow=N, ncol=K)
  log_likelihood = 0
  for (n in 1:N) {
    x <- mnist$train$x[n,]
    sum <- 0
    for (k in 1:K) {
      resp[n, k] = log(pi[k]) - (1/2)*log(sigma[k]*D) - (1/2)*(1/sigma[k])*norm(x - mu[, k], type="2")
    }
    resp[n, ] = exp(resp[n, ] - max(resp[n,]) - log(sum(exp(resp[n,]-max(resp[n,])))))
    log_likelihood = log_likelihood + log(sum)
  }

  # M-Step
  pi = colSums(resp) / N
  mu = t(t(t(mnist$train$x) %*% resp) / colSums(resp))
  for (k in 1:K) {
    sigma[k] = 0
    for (n in 1:N) {
      sigma[k] = sigma[k] + resp[n, k] * (mnist$train$x[n,] - mu[, k])%*%(mnist$train$x[n,] - mu[, k])
```

```
    }
    sigma[k] = sigma[k] / colSums(resp)[k] + 0.05
  }

  # Check for convergence
  if (abs(prev_log_likelihood  - log_likelihood) < 0.0001) {
    break
  }
  prev_log_likelihood <- log_likelihood
}
```

```
par(mfrow=c(5, 14))
# Training accuracy
for (k in 1:5) {
  class <- mnist$train$x[closest == k,]
  number <- names(sort(-table(mnist$train$y[closest == k])))[1]
  acc <- sum(mnist$train$y[closest == k] == number) / length(mnist$train$y[closest == k])

  # Plot 1
  sampled <- mnist$train$x[sample(which(closest == k), 12),]
  for (i in 1:12) {
    image(matrix(sampled[i,], ncol=14, nrow=14)[,14:1])
  }

  # Plot 2
  image(matrix(mu[,k], ncol=14, nrow=14)[,14:1])

  # Plot 3
  var <- colSums(class^2) / dim(class)[1] - mu[,k]^2
  var <- var / max(var) * 255
  var[var < 51] <- 0
  image(matrix(var, ncol=14, nrow=14)[,14:1])

  # Plot 4
  print(sprintf("Cluster %d", as.integer(number)))
  # Number with True Label in Cluster vs Number with True Label
  print(c(sum(mnist$train$y[closest == k] == number), length(mnist$train$y[closest == k])))
  # Accuracy Rate
  print(acc)
  # Misclassification Rate
  print(1-acc)
}
```

Question 1.2.V

Question 1.3

Assume a mixture of spherical Gaussians with same covariance matrix $\sigma^2 I$

E-step:

$lim_{\sigma^2 \to 0} \gamma_{ik} = lim_{\sigma^2 \to 0} p_{\theta^{old}}(Z_i = k|x_i)$

$= lim_{\sigma^2 \to 0} \frac{p(x_i|Z_i=k)\eta_j}{\Sigma_{j=1}^{k} p(x_i|Z_i=j)\eta_j}$

$= lim_{\sigma^2 \to 0} \frac{exp(-1/2(x_i-\mu_k)^\top \sigma^{-2} I(x_i-\mu_k))\eta_j}{\Sigma_{j=1}^K exp(-1/2(x_i-\mu_j)^\top \sigma^{-2} I(x_i-\mu_j))\eta_j}$

$= lim_{\sigma^2 \to 0} \frac{exp(-||x_i-\mu_k|||_2^2/(2\sigma^2))\eta_j}{\Sigma_{j=1}^K exp(-||x_i-\mu_j||_2^2/(2\sigma^2))\eta_j}$

$= r_{nk} = \begin{cases} 1, & \text{if } k = argmin_j ||x_i - \mu_j||_2^2 \\ 0, & \text{o.w.} \end{cases}$

Since the term with the smallest $||x_i - \mu_j||_2^2$ goes to 0 the slowest

M-step:

Since the only variable that is responsible for the E-step result is μ_j we only need to update μ_j:

$\frac{\partial}{\partial \mu_j} \Sigma_{i=1;r_{ij}=1}^N ||x_i - \mu_j||_2^2 = \Sigma_{i=1}^N r_{ij} \frac{\partial}{\partial \mu_j} (x_i - \mu_j)^\top (x_i - \mu_j)$

$= -2\Sigma_{i=1}^N r_{ij}(x_i - \mu_j) = 0$

$\Rightarrow \Sigma_{i=1}^N r_{ij}(x_i - \mu_j) = 0$

$\Rightarrow \Sigma_{i=1}^N x_i r_{ij} = \mu_j \Sigma_{i=1}^N r_{ij}$

$\Rightarrow \frac{\Sigma_{i=1}^N r_{ij} x_i}{\Sigma_{i=1}^N r_{ij}} = \mu_j$

Question 1.4

The K-means algorithm achieved an average accuracy of 90% for the digits. However, it is not satisfactory as it means the algorithm will fail 10% times when classifying digits. EM if implemented correctly would have achieved a higher accuracy for spherical and even higher for diagonal covariance. K means only uses mean to make classifications but EM uses far more parameters that increase model flexibility to more accurately model the underlying distribution of the data. The algorithms overlook the possibility of transforming the data to feature vectors, which could improve performance.

The K-means converged quickly with few steps, which is impressive given how well it performs. EM would have converged slower for spherical and even slower for diagonal due to more parameters for each to update and being more computationally expensive, however at the benefit of increasing the accuracy ceiling.

The mixture models for Gaussian distributions might not be the optimal model for classifying digits, instead one can experiment with other models such as a categorical mixture model.

The initialization strategy was to randomly sample an initial mean vector between 0 and 1 to fall in between possible values of the data. This strategy was successful as K-means managed to converge every initialization, and any further changes to the initialization strategy lead to no further improvement. EM would have been similar in regards to the mean initialization, with same pi for all classes, with only a focus on making sure the covariance initialization had its diagonal values of similar magnitude.

Question 2

Question 2.1

```
source("/Users/vladislavtrukhin/Downloads/_data_hw6/script1.R")
```

```
## Loading required package: NLP
```

```
head(dat, 3)[1]
```

```
##                 name
## 1         Michel Che
## 2 Hossein Modarressi
```

```
## 3      Xiao-Gang Wen
```

Michel Che - Chemistry Professor, Hossein Modarressi - Muslim Jurist, Xiao-Gang Wen - Chinese-born American Physicist

```
dim(dtm.mat.raw) # Number of Individuals; Number of Words
```

```
## [1]  812 6888
```

```
word_freq <- sort(colSums(dtm.mat.raw))
tail(word_freq, 10) # Top 10 Most Frequent Words
```

```
##     new    with     has    from     his     was     for univers     and     the
##    1174    1674    1708    2126    2353    2817    3165    3169   10729   17316
```

```
quantile(word_freq) # Quantiles
```

```
##    0%   25%   50%   75%  100%
##     2     3     5    14 17316
```

Question 2.2

```
source("/Users/vladislavtrukhin/Downloads/_data_hw6/script2.R")
word_weight <- sort(colSums(dtm.mat))
tail(word_weight, 10) # Top 10 Highest Weighted Words
```

```
##  research   histori       new  mathemat    scienc       law    econom     music
##  903.4017  905.1748  908.1503  975.4508  989.8247 1132.1644 1160.7760 1179.0219
##       her       she
## 1611.8679 2414.6500
```

Question 2.3

```
source("/Users/vladislavtrukhin/Downloads/_data_hw6/script3.R")
dim(dtm.mat.raw) # Number of Individuals; Number of Words
```

```
## [1]  812 6632
```

```
word_freq <- sort(colSums(dtm.mat.raw))
tail(word_freq, 10) # Top 10 Most Frequent Words
```

```
##   physic   theori mathemat    music      law    polit   econom  histori
##      310      323      329      364      389      390      397      418
##      her      she
##      633      956
```

Question 2.4

```
ix <- match("Ben Bernanke", dat$name)
dat$text[ix]
```

```
## [1] ben shalom bernanke brnki brnangkee born december 13 1953 is an american economist at the brookin
## 59071 Levels: 108 born 1978 is an italian artist in the field of street art and contemporary art fro
```

```
tail(sort(dtm.mat[ix,]), 10) # Top 10 Highest Weighted Words
```

```
##     janet   volatil    econom      bush   succeed      term     feder  chairman
##  8.665336  8.665336  8.771607  9.929792  9.929792 15.431924 20.107399 20.619103
```

```
##    reserv   bernank
## 27.087042 40.401867
```

```
tail(sort(dtm.mat.raw[ix,]), 10) # Top 10 Most Frequent Words
```

```
##   second  succeed    tenur     when   econom     term  bernank    feder
##        2        2        2        2        3        4        5        5
##   reserv chairman
##        5        6
```

Question 2.5

```
set.seed(10)
res <- norm.sim.ksc(quick.norm(t(dtm.mat), 2), 8)
res$size # Cluster sizes
```

```
## [1] 1719  450  713  839  616  918  916  717
```

```
# Remove words from clusters not present in dtm.mat.raw
idx = which(word.presence >= quantile(word.presence, prob = 0.99))
res$cluster = res$cluster[-idx]
idx = which(colnames(dtm.mat) %in% common.words[1:300,1])
res$cluster = res$cluster[-idx]

for (k in 1:8) {
  print(sprintf("Cluster %d", k))
  print(tail(sort(colSums(dtm.mat.raw[, res$cluster == k])), 25)) # Top 25 words
  print(quantile(colSums(dtm.mat.raw[, res$cluster == k]))) # Quantiles
}
```

```
## [1] "Cluster 1"
##    when   appear   london     live    organ america     both   record perform      won
##     159      160      162      166      167      172      173      174      175      182
##   elect   jersey     join      had     well     dure     name   former     mani     citi
##     194      204      204      205      208      223      223      227      231      248
##    team     play    music      her      she
##     267      295      364      633      956
##   0%  25%  50%  75% 100%
##    2    2    5   14  956
## [1] "Cluster 2"
##        aid    marshal      given       much     togeth       piec understand
##         46         47         48         50         50         52         54
##     differ       them     endow        way    russian     result    communic
##         55         55         56         56         60         62         73
##     recipi       form      creat  historian        san       film    interest
##         79         82         84         88        109        121        135
##    between        not       into      relat
##        145        145        164        172
##   0%  25%  50%  75% 100%
##    2    3    5   13  172
## [1] "Cluster 3"
##        same        jame      father         ann      centuri        essay
##          63          64          67          75          75          78
##        oper    honorari     collect     british      famili       museum
##          79          82          88          90          90         101
##        life        four     commiss       medal         edit architectur
```

```
##         104          106          108          110          122          130 
##      modern  fellowship       design       cultur          use        sever 
##         144          157          159          173          193          200 
##      recent 
##         201 
##   0%  25%  50%  75% 100% 
##    2    2    5   10  201 
## [1] "Cluster 4"
##   religion particular  philosoph      these       paul      wrote       issu 
##         87         88         90         90         91         92         95 
##        how       hold      earli    written      among     critic    william 
##         99        109        134        135        136        141        146 
##     oxford     church        but      write    theolog     articl   columbia 
##        148        153        172        175        177        187        192 
##    faculti philosophi      press    journal 
##        194        216        233        265 
##   0%  25%  50%  75% 100% 
##    2    3    5   14  265 
## [1] "Cluster 5"
##    medicin       next      along     strong       like     street    product 
##         47         47         48         49         55         56         61 
##     summer      medic     achiev     biolog      charl   investig laboratori 
##         61         62         67         73         73         78         81 
##     poetri    collabor     theatr     direct       than       they    project 
##         83         86         86        102        114        143        154 
##       more     review      human      their 
##        179        191        220        229 
##   0%  25%  50%  75% 100% 
##    2    3    5   11  229 
## [1] "Cluster 6"
##    coauthor   scientist     theoret       model        area       paper 
##          91          92          94         100         102         124 
##         now      system distinguish      number      doctor       engin 
##         128         130         132         135         155         167 
##   contribut      editor   technolog      advanc       field       known 
##         176         180         184         191         198         200 
##       under      comput       prize     academi      physic      theori 
##         207         235         261         290         310         323 
##    mathemat 
##         329 
##   0%  25%  50%  75% 100% 
##    2    2    5   10  329 
## [1] "Cluster 7"
##      chair       yale    general       were      later    appoint      posit 
##        170        172        173        175        177        178        180 
##     senior      found     taught      offic     polici      teach california 
##        180        187        190        194        205        222        229 
##    foundat       educ      visit washington       then    develop    program 
##        237        238        238        241        260        271        282 
##      board      polit     econom    histori 
##        287        390        397        418 
##   0%  25%  50%  75% 100% 
##    2    3    6   21  418 
## [1] "Cluster 8"
```

```
##  angel confer  start  media  spent   base report  georg   firm writer   juli
##     88     90     92     94     94    101    101    103    108    108    117
##   game  leagu   head season    him  coach   best   earn  befor  three   high
##    122    122    124    133    137    151    157    179    181    183    195
## career assist    law
##    216    224    389
##   0%  25%  50%  75% 100%
##    2    3    6   16  389
```

Question 3

```r
load("/Users/vladislavtrukhin/Downloads/_data_hw6/gdp.Rdata")
source("/Users/vladislavtrukhin/Downloads/_data_hw6/q4.R")
```

```
##
## Attaching package: 'igraph'

## The following objects are masked from 'package:stats':
##
##     decompose, spectrum

## The following object is masked from 'package:base':
##
##     union
```

Question 3.1

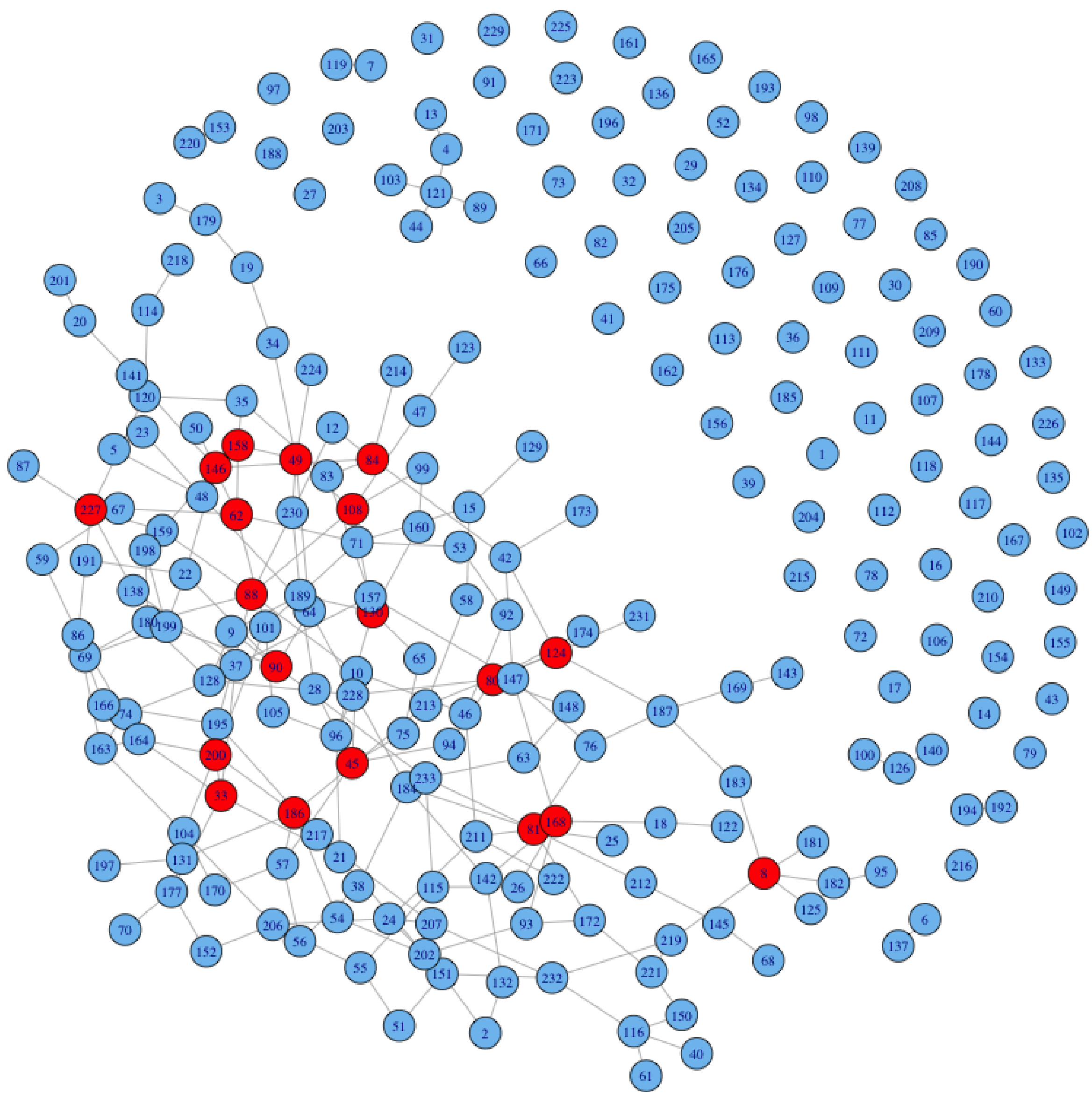
```r
# Remove rows with all NA values
ix <- which(is.na(rowMeans(gdp, na.rm=TRUE)))
gdp <- gdp[-ix, ]

# Replace NA entries with mean
for(i in 1:nrow(gdp)){
  gdp[i, is.na(gdp[i,])] <- rowMeans(gdp[i,], na.rm = TRUE)
}
```

Question 3.2

```r
# Regress over one country's GDP over all countries' GDP
mat <- matrix(0, nrow(gdp), nrow(gdp))
for (i in 1:nrow(gdp)){
  t <- cv.glmnet(t(gdp[-i, ]), t(gdp[i, ]), nfolds=10)
  coef <- coef(t, s="lambda.1se")[-1]
  coef <- c(coef[1:(i-1)], 0, coef[-(1:(i-1))])
  neighbors <- which(coef != 0)
  mat[i, neighbors] = 1
}

# Iterate diagonally to remove connections
I <- nrow(gdp)
for (i in 1:I) {
  for (j in 1:nrow(gdp)) {
    if (mat[i, j] != mat[j, i]) {
```

```
      mat[i, j] = 0
      mat[j, i] = 0
    }
  }
  I <- I - 1
}
graphplot(mat)
```

```
## pdf
##   2
```



The countries with the red nodes in the graph are: Chad, Nigeria, Iraq, St. Lucia, Cambodia, Comoros, West Bank and Gaza, Latvia, Montenegro, Czech Republic, Bosnia and Herzegovina, Bulgaria, Bermuda, Spain, Mongolia, Uruguay, Botswana, Bolivia, Central African Republic, Belize, Greece, Cyprus, Sudan, Trinidad and Tobago, Seychelles, Brunei Darussalam, Timor-Leste, Mauritania

Question 3.3

Two events of a coin toss are globally independent, but become conditionally dependent when information is provided about the sum of the results, treating heads as 1 and tails as -1, as one can narrow down the results to a greater probability. Two responses in the linear regression model with the same design covariate are conditionally independent when given the model but are globally dependent when the model is not provided, as its that dependency that allows one to derive an estimated model.

Question 3.4

N,Y,Y,Y,N

Question 3.5

$\Theta\Sigma = I$

$\Rightarrow \Theta = \Sigma^{-1}$

$\Rightarrow \begin{bmatrix} \Theta_{AA} & \Theta_{AA^c} \\ \Theta_{A^cA} & \Theta_{A^cA^c} \end{bmatrix} = \begin{bmatrix} \Sigma_{AA} & \Sigma_{AA^c} \\ \Sigma_{A^cA} & \Sigma_{A^cA^c} \end{bmatrix}^{-1}$

$\Rightarrow \begin{bmatrix} \Theta_{AA} & \Theta_{AA^c} \\ \Theta_{A^cA} & \Theta_{A^cA^c} \end{bmatrix} = \begin{bmatrix} (\Sigma_{AA} - \Sigma_{AA^c}\Sigma_{A^cA^c}^{-1}\Sigma_{A^cA})^{-1} & ... \\ ... & ... \end{bmatrix}$

$\Rightarrow \Theta_{AA} = (\Sigma_{AA} - \Sigma_{AA^c}\Sigma_{A^cA^c}^{-1}\Sigma_{A^cA})^{-1}$

$\Rightarrow \Theta_{AA}^{-1} = \Sigma_{AA} - \Sigma_{AA^c}\Sigma_{A^cA^c}^{-1}\Sigma_{A^cA}$

Question 3.6

X_i and X_j are independent given $X_{\backslash i,j}$ iff Θ_{AA}^{-1} is diagonal iff Θ_{AA} is diagonal iff $\Theta_{i,j} = 0$

Question 3.7

$Var(X_1) = Var(\Sigma_{1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1}X_{\backslash 1} + \epsilon)$

$= Var(\Sigma_{1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1}X_{\backslash 1}) + Var(\epsilon) + 2cov(\Sigma_{1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1}X_{\backslash 1}, \epsilon)$

$= \Sigma_{1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1}Var(X_{\backslash 1})\Sigma_{\backslash 1,\backslash 1}^{-1}\Sigma_{1,\backslash 1} + \Sigma_{1,1} - \Sigma_{1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1}\Sigma_{\backslash 1,1} + 2cov(\Sigma_{1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1}X_{\backslash 1}, \epsilon)$

$= \Sigma_{1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1}\Sigma_{\backslash 1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1}\Sigma_{1,\backslash 1} + \Sigma_{1,1} - \Sigma_{1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1}\Sigma_{\backslash 1,1} + 2cov(\Sigma_{1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1}X_{\backslash 1}, \epsilon)$

$= \Sigma_{1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1}\Sigma_{1,\backslash 1} + \Sigma_{1,1} - \Sigma_{1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1}\Sigma_{\backslash 1,1} + 2cov(\Sigma_{1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1}X_{\backslash 1}, \epsilon)$

$= \Sigma_{1,1} + 2(\Sigma_{1,\backslash 1}\Sigma_{\backslash 1,\backslash 1}^{-1})cov(X_{\backslash 1}, \epsilon)$

We know $Var(X_1) = \Sigma_{1,1}$ under this model so this is only true if $cov(X_{\backslash 1}, \epsilon) = 0 \implies X_{\backslash 1}$ independent of ϵ.

$\begin{bmatrix} \Theta_{11} & \Theta_{1\backslash 1} \\ \Theta_{\backslash 11} & \Theta_{\backslash 1\backslash 1} \end{bmatrix} \begin{bmatrix} \Sigma_{11} & \Sigma_{1\backslash 1} \\ \Sigma_{\backslash 11} & \Sigma_{\backslash 1\backslash 1} \end{bmatrix} = I$

$\Rightarrow \Theta_{11}\Sigma_{1\backslash 1} + \Theta_{1\backslash 1}\Sigma_{\backslash 1\backslash 1} = 0$

$\Rightarrow \Sigma_{1\backslash 1} = -\Theta_{11}^{-1}\Theta_{1\backslash 1}\Sigma_{\backslash 1\backslash 1}$

$\Rightarrow \Sigma_{\backslash 11} = -\Sigma_{\backslash 1\backslash 1}\Theta_{11}^{-1}\Theta_{\backslash 11}$

$\beta = \Sigma_{\backslash 1\backslash 1}^{-1}\Sigma_{\backslash 1,1}$

$= -\Sigma_{\backslash 1 \backslash 1}^{-1} \Sigma_{\backslash 1 \backslash 1} \Theta_{11}^{-1} \Theta_{\backslash 11}$

$= -\Theta_{11}^{-1} \Theta_{\backslash 11}$

Question 3.8

$P(X_j, X_k | X_{\backslash j,k}) = \frac{P(X)}{P(X_{\backslash j,k})}$

$= \frac{Z_{X_{\backslash j,k}} exp(\Sigma_{i=1}^d \beta_i x_i + \Sigma_{l<i} \beta_{li} x_l x_i)}{Z_X exp(\Sigma_{i=1}^d \beta_i x_i + \Sigma_{l<i} \beta_{li} x_l x_i - \beta_j x_j - \beta_k x_k - \Sigma_{j \neq i} \beta_{ji} x_j x_i - \Sigma_{k \neq i} \beta_{ki} x_k x_i + \beta_{jk} x_j x_k)}$

$= \frac{Z_{X_{\backslash j,k}} exp(\Sigma_{i=1}^d \beta_i x_i + \Sigma_{l<i} \beta_{li} x_l x_i)}{Z_X exp(\Sigma_{i=1}^d \beta_i x_i + \Sigma_{l<i} \beta_{li} x_l x_i) exp(-\beta_j x_j - \beta_k x_k - \Sigma_{j \neq i} \beta_{ji} x_j x_i - \Sigma_{k \neq i} \beta_{ki} x_k x_i + \beta_{jk} x_j x_k)}$

$= \frac{Z_{X_{\backslash j,k}}}{Z_X} exp(\beta_j x_j + \Sigma_{j \neq i} \beta_{ji} x_j x_i) exp(\beta_k x_k + \Sigma_{k \neq i} \beta_{ki} x_k x_i) exp(\beta_{jk} x_j x_k)$

$= \frac{Z_{X_{\backslash j,k}}}{Z_X} \frac{exp(\beta_j x_j + \Sigma_{j \neq i} \beta_{ji} x_j x_i) exp(\Sigma_{i=1}^d \beta_i x_i + \Sigma_{l<i} \beta_{li} x_l x_i - \beta_j x_j - \beta_k x_k - \Sigma_{j \neq i} \beta_{ji} x_j x_i - \Sigma_{k \neq i} \beta_{ki} x_k x_i + \beta_{jk} x_j x_k)}{exp(\Sigma_{i=1}^d \beta_i x_i + \Sigma_{l<i} \beta_{li} x_l x_i - \beta_j x_j - \beta_k x_k - \Sigma_{j \neq i} \beta_{ji} x_j x_i - \Sigma_{k \neq i} \beta_{ki} x_k x_i + \beta_{jk} x_j x_k)}$

$\frac{exp(\beta_k x_k + \Sigma_{k \neq i} \beta_{ki} x_k x_i) exp(\Sigma_{i=1}^d \beta_i x_i + \Sigma_{l<i} \beta_{li} x_l x_i - \beta_j x_j - \beta_k x_k - \Sigma_{j \neq i} \beta_{ji} x_j x_i - \Sigma_{k \neq i} \beta_{ki} x_k x_i + \beta_{jk} x_j x_k)}{exp(\Sigma_{i=1}^d \beta_i x_i + \Sigma_{l<i} \beta_{li} x_l x_i - \beta_j x_j - \beta_k x_k - \Sigma_{j \neq i} \beta_{ji} x_j x_i - \Sigma_{k \neq i} \beta_{ki} x_k x_i + \beta_{jk} x_j x_k)} exp(\beta_{jk} x_j x_k)$

$= \frac{Z_{X_{\backslash j,k}}}{Z_X} \frac{exp(\Sigma_{i=1}^d \beta_i x_i + \Sigma_{l<i} \beta_{li} x_l x_i - \beta_k x_k - \Sigma_{k \neq i} \beta_{ki} x_k x_i)}{exp(\Sigma_{i=1}^d \beta_i x_i + \Sigma_{l<i} \beta_{li} x_l x_i - \beta_j x_j - \beta_k x_k - \Sigma_{j \neq i} \beta_{ji} x_j x_i - \Sigma_{k \neq i} \beta_{ki} x_k x_i + \beta_{jk} x_j x_k)}$

$\frac{exp(\Sigma_{i=1}^d \beta_i x_i + \Sigma_{l<i} \beta_{li} x_l x_i - \beta_j x_j - \Sigma_{j \neq i} \beta_{ji} x_j x_i)}{exp(\Sigma_{i=1}^d \beta_i x_i + \Sigma_{l<i} \beta_{li} x_l x_i - \beta_j x_j - \beta_k x_k - \Sigma_{j \neq i} \beta_{ji} x_j x_i - \Sigma_{k \neq i} \beta_{ki} x_k x_i + \beta_{jk} x_j x_k)} exp(\beta_{jk} x_j x_k)^3$

$= \frac{Z_{X_{\backslash j,k}}}{Z_X} \frac{Z_{X_{\backslash k}} P(X_{\backslash k})}{Z_{X_{\backslash j,k}} P(X_{\backslash j,k})} \frac{Z_{X_{\backslash j}} P(X_{\backslash j})}{Z_{X_{\backslash j,k}} P(X_{\backslash j,k})} exp(\beta_{jk} x_j x_k)^3$

$= \frac{Z_{X_{\backslash j}} Z_{X_{\backslash k}}}{Z_{X_{\backslash j,k}} Z_X} P(X_j | X_{\backslash j,k}) P(X_k | X_{\backslash j,k}) exp(\beta_{jk} x_j x_k)^3$

$= P(X_j | X_{\backslash j,k}) P(X_k | X_{\backslash j,k}) exp(\beta_{jk} x_j x_k)^3$

$= P(X_j | X_{\backslash j,k}) P(X_k | X_{\backslash j,k})$ if and only if $\beta_{jk} = 0$

Therefore X_j independent of X_k given $X_{\backslash j,k}$ if and only if $\beta_{jk} = 0$

Question 3.9

$P(X_j = 1 | X_{\backslash j} = x_{\backslash j}) = \frac{P(X_j=1, X_{\backslash j}=x_{\backslash j})}{P(X_j=1, X_{\backslash j}=x_{\backslash j}) + P(X_j=-1, X_{\backslash j}=x_{\backslash j})}$

$= \frac{1}{1 + \frac{P(X_j=-1, X_{\backslash j}=x_{\backslash j})}{P(X_j=1, X_{\backslash j}=x_{\backslash j})}}$

$\Rightarrow \frac{P(X_j=-1, X_{\backslash j}=x_{\backslash j})}{P(X_j=1, X_{\backslash j}=x_{\backslash j})}$

$= \frac{Z^{-1} exp(-\beta_j - \Sigma_{i \neq j} \beta_{ij} x_i + \Sigma_{i=1; i \neq j}^d \beta_i x_i + \Sigma_{i<k; i,k \neq j} \beta_{ik} x_i x_k)}{Z^{-1} exp(\beta_j + \Sigma_{i \neq j} \beta_{ij} x_i + \Sigma_{i=1; i \neq j}^d \beta_i x_i + \Sigma_{i<k; i,k \neq j} \beta_{ik} x_i x_k)}$

$= \frac{exp(-\beta_j - \Sigma_{i \neq j} \beta_{ij} x_i) exp(\Sigma_{i=1; i \neq j}^d \beta_i x_i + \Sigma_{i<k; i,k \neq j} \beta_{ik} x_i x_k)}{exp(\beta_j + \Sigma_{i \neq j} \beta_{ij} x_i) exp(\Sigma_{i=1; i \neq j}^d \beta_i x_i + \Sigma_{i<k; i,k \neq j} \beta_{ik} x_i x_k)}$

$= \frac{exp(-\beta_j - \Sigma_{i \neq j} \beta_{ij} x_i)}{exp(\beta_j + \Sigma_{i \neq j} \beta_{ij} x_i)}$

$= exp(-2(\beta_j + \Sigma_{i \neq j} \beta_{ij} x_i))$

$\Rightarrow \frac{1}{1 + \frac{P(X_j=-1, X_{\backslash j}=x_{\backslash j})}{P(X_j=1, X_{\backslash j}=x_{\backslash j})}}$

$= \frac{1}{1 + exp(-2(\beta_j + \Sigma_{i \neq j} \beta_{ij} x_i))}$

Question 4

```r
# w[1]: w_chains,   w[2]: w_inter-chain
# w[3]: w_chain-empty,   w[4]: w_empty
# w[5]: h_stone
w <- c(2.47, 0.521, 0.442, 0.427, 0.265)

# Go board for game 2
C2 = as.matrix(read.table("/Users/vladislavtrukhin/Downloads/_data_hw6/AlphaGo-vs-Lee-game2_80.txt", hea
# Go board for game 4
C4 = as.matrix(read.table("/Users/vladislavtrukhin/Downloads/_data_hw6/AlphaGo-vs-Lee-game4_80.txt", hea

construct.ising.graph <- function(weight, c) { # Complete the following function
    g=matrix(0, 19^2, 19^2)
    for (i in 1:19){
        for (j in 1:19){ # Enumerate every point on the board. i is the row index. j is the column inde
            i0=(j-1)*19+i
            i1=j*19+i # right neighbor
            i2=(j-1)*19+i+1 # lower neighbor
            if (j<19){
                if (c[i,j]*c[i,j+1]==1){
                    g[i0, i1]=w[1]
                    g[i1, i0]=w[1]
                }
                else if (c[i,j]*c[i,j+1]==-1){
                    g[i0, i1]=w[2]
                    g[i1, i0]=w[2]
                }
                else if ((c[i,j]==0)&(c[i, j+1]==0)){
                    g[i0, i1]=w[4]
                    g[i1, i0]=w[4]
                }
                else {
                    g[i0, i1]=w[3]
                    g[i1, i0]=w[3]
                }
            }
            if (i<19){
                if (c[i,j]*c[i+1,j]==1){
                    g[i0, i2]=w[1]
                    g[i2, i0]=w[1]
                }
                else if (c[i,j]*c[i+1,j]==-1){
                    g[i0, i2]=w[2]
                    g[i2, i0]=w[2]
                }
                else if ((c[i,j]==0)&(c[i+1, j]==0)){
                    g[i0, i2]=w[4]
                    g[i2, i0]=w[4]
                }
                else {
                    g[i0, i2]=w[3]
                    g[i2, i0]=w[3]
```

```r
                }
            }
        }
    }
    return(g)
}

predict_go <- function(w, C, nsample){
  # C is a matrix, obtained from the current go board
  # do sampling
  W <- construct.ising.graph(w, C); # Coefficient matrix for the coupling term.

  # Complete the line below with your code (just one line of code)
  h <- w[5] * C # Coeffcient vector for the external field term.

  S_mat <- IsingSampler(nsample, graph = W,
                        thresholds = h,
                        nIter = 100,
                        response = c(-1L,1L)) # Each row of S_mat is a simulated realization of the fin

  s <- colMeans(S_mat); # s is the empirical expectation of the final territory outcome.
  val = sum(s) - 3.75; # Scoring adjustment
  if(val >0){
    result = TRUE
  }
  else{
    result = FALSE
  }
  return(list(mean = s, result = result))
}# return the expectation

set.seed(2016)

pred_game2 = predict_go(w, C2, nsample = 500)
expectation = pred_game2$mean;
result = pred_game2$result
if(result){
  cat(sprintf("Alpha-go wins game 2"))
}else{
  cat(sprintf("Lee Sedol wins game 2"))
}
```

```
## Alpha-go wins game 2
```

```r
pred_game4 = predict_go(w, C4, nsample = 500)
expectation = pred_game4$mean;
result = pred_game4$result
if(result){
  cat(sprintf("Alpha-go wins game 4"))
}else{
  cat(sprintf("Lee Sedol wins game 4"))
}
```

```
## Lee Sedol wins game 4
```

Question 5

Question 5.1

```r
source("/Users/vladislavtrukhin/Downloads/_data_hw6/deepfeature.R")
array_lasso <- cv.glmnet(train.image.array, train.label, family="binomial",
                         type.measure="class", nfolds=10)
feature_lasso <- cv.glmnet(train.deep.feature, train.label, family="binomial",
                           type.measure="class", nfolds=10)
array_pred <- predict(array_lasso, test.image.array, s="lambda.1se")
feature_pred <- predict(feature_lasso, test.deep.feature, s="lambda.1se")
# Misclassification rate for array
sum((test.label == "dog") != (array_pred > 0)) / length(test.label)
```

```
## [1] 0.4133333
```

```r
# Misclassification rate for feature
sum((test.label == "dog") != (feature_pred > 0)) / length(test.label)
```

```
## [1] 0.235
```

Question 5.2

```r
source("/Users/vladislavtrukhin/Downloads/_data_hw6/deepretrieval.R")
```

```r
dist <- colSums((t(deep.feature) - deep.feature[350,])^2)
closest <- order(dist)[2:4]
```

```r
plot.image(image.array[closest[1],])
```

```r
plot.image(image.array[closest[2],])
```

```
plot.image(image.array[closest[3],])
```