

# HW3

2024-11-07

```
X <- read.table("face.txt", sep = " ", header = FALSE)
X <- as.matrix(X)

dim(X) # Should be 2429 x 361

## [1] 2429 361

num_train <- 2100
num_test <- 2429 - num_train

X_train <- X[1:num_train, ]
X_test <- X[(num_train + 1):2429, ]

mean_face <- colMeans(X_train)

X_train_centered <- scale(X_train, center = mean_face, scale = FALSE)

cov_matrix <- t(X_train_centered) %*% X_train_centered / (num_train - 1)

eigen_decomp <- eigen(cov_matrix)
eigenvalues <- eigen_decomp$values
eigenvectors <- eigen_decomp$vectors

K <- 20
eigenfaces <- eigenvectors[, 1:K]

# Plotting Eigenfaces
windows(width = 12, height = 8)
par(mfrow = c(2, 5), mar = c(1, 1, 1, 1))

for (i in 1:10) {
  eigenface <- matrix(eigenfaces[, i], nrow = 19, ncol = 19)
  image(t(eigenface)[, nrow(eigenface):1], axes = FALSE, col = gray.colors(256))
  title(main = paste("Eigenface", i))
}

reconstruct_face <- function(face, eigenfaces, mean_face) {
  weights <- t(eigenfaces) %*% (face - mean_face)
  reconstructed <- mean_face + eigenfaces %*% weights
  return(reconstructed)
}

reconstructed_faces <- matrix(0, nrow = num_test, ncol = ncol(X))
```

```

for (i in 1:num_test) {
  reconstructed_faces[i, ] <- reconstruct_face(X_test[i, ], eigenfaces, mean_face)
}

windows(width = 12, height = 8)
par(mfrow = c(2, 10), mar = c(1, 1, 1, 1))

for (i in 1:10) {
  original_face <- matrix(X_test[i, ], nrow = 19, ncol = 19)
  reconstructed_face <- matrix(reconstructed_faces[i, ], nrow = 19, ncol = 19)
  image(t(original_face)[, nrow(original_face):1], axes = FALSE, col = gray.colors(256))
  title(main = paste("Original", i))
  image(t(reconstructed_face)[, nrow(reconstructed_face):1], axes = FALSE, col = gray.colors(256))
  title(main = paste("Reconstructed", i))
}

reconstruction_errors <- numeric(K)

for (k in 1:K) {
  eigenfaces_k <- eigenvectors[, 1:k]
  reconstructed_faces_k <- matrix(0, nrow = num_test, ncol = ncol(X))

  for (i in 1:num_test) {
    reconstructed_faces_k[i, ] <- reconstruct_face(X_test[i, ], eigenfaces_k, mean_face)
  }
  error <- (X_test - reconstructed_faces_k)^2
  reconstruction_errors[k] <- mean(error)
}

windows(width = 12, height = 8)
par(mar = c(4, 4, 2, 1))
plot(1:K, reconstruction_errors, type = "b", pch = 19, xlab = "Number of Eigenfaces K", ylab = "Reconstruction Error")

nnmf <- function(X, r, maxiter = 1000, tol = 1e-4, V = NULL, W = NULL) {

  if (is.null(V)) {
    V <- matrix(runif(nrow(X) * r), nrow = nrow(X), ncol = r)
  }

  if (is.null(W)) {
    W <- matrix(runif(r * ncol(X)), nrow = r, ncol = ncol(X))
  }

  for (t in 1:maxiter) {
    b <- V %*% W
    V <- V * (X %*% t(W)) / (b %*% t(W))
    b <- V %*% W
    W <- W * (t(V) %*% X) / (t(V) %*% b)
    if (t > 1 && abs(L(V, W, X) - L(V_prev, W_prev, X)) < tol) {
      break
    }
    V_prev <- V
    W_prev <- W
  }
}

```

```

    }
    return(list(V = V, W = W))
}

L <- function(V, W, X) {
  sum((X - V %*% W)^2)
}

X <- as.matrix(iris[, -5])

runtimes <- c()
rs <- c(10, 20, 30, 40, 50)

for (r in rs) {
  start_time <- Sys.time()
  nnmf(X, r, maxiter = 1000)
  end_time <- Sys.time()
  runtimes <- c(runtimes, as.numeric(end_time - start_time, units = "secs"))
}

windows(width = 12, height = 8)
par(mar = c(4, 4, 4, 2) + 0.1)
plot(rs, runtimes, type = "b", xlab = "Rank (r)", ylab = "Runtime (seconds)", main = "Runtime vs Rank")
flush.console()

for (r in rs) {
  result <- nnmf(X, r, maxiter = 1000)
  V <- result$V
  W <- result$W

  reconstructed_faces <- V %*% W

  windows(width = 12, height = 8)
  par(mfrow = c(2, 3), mar = c(2, 2, 2, 2) + 0.1)

  for (i in 1:3) {
    image(matrix(X[i, ], nrow = 4, ncol = 4), col = gray(seq(0, 1, length = 256)), main = "Original")
    image(matrix(reconstructed_faces[i, ], nrow = 4, ncol = 4), col = gray(seq(0, 1, length = 256)), main = "Reconstructed")
  }
  flush.console()

  windows(width = 12, height = 8)
  par(mfrow = c(5, 10), mar = c(2, 2, 2, 2) + 0.1)

  for (i in 1:nrow(W)) {
    image(matrix(W[i, ], nrow = 4, ncol = 4), col = gray(seq(0, 1, length = 256)), main = paste("Basis", i))
  }
  flush.console()
}

r <- 20
result1 <- nnmf(X, r, maxiter = 1000)

```

```
V1 <- result1$V
W1 <- result1$W

V2 <- matrix(runif(nrow(X) * r), nrow = nrow(X), ncol = r)
W2 <- matrix(runif(r * ncol(X)), nrow = r, ncol = ncol(X))

result2 <- nnmf(X, r, V = V2, W = W2, maxiter = 1000)

V2 <- result2$V
W2 <- result2$W

print(sum(abs(V1 - V2)))
```

```
## [1] 920.9187
```

```
print(sum(abs(W1 - W2)))
```

```
## [1] 29.54119
```