Statistical Learning week 1 - Answers to exercises

Exercise 1: Shrinkage

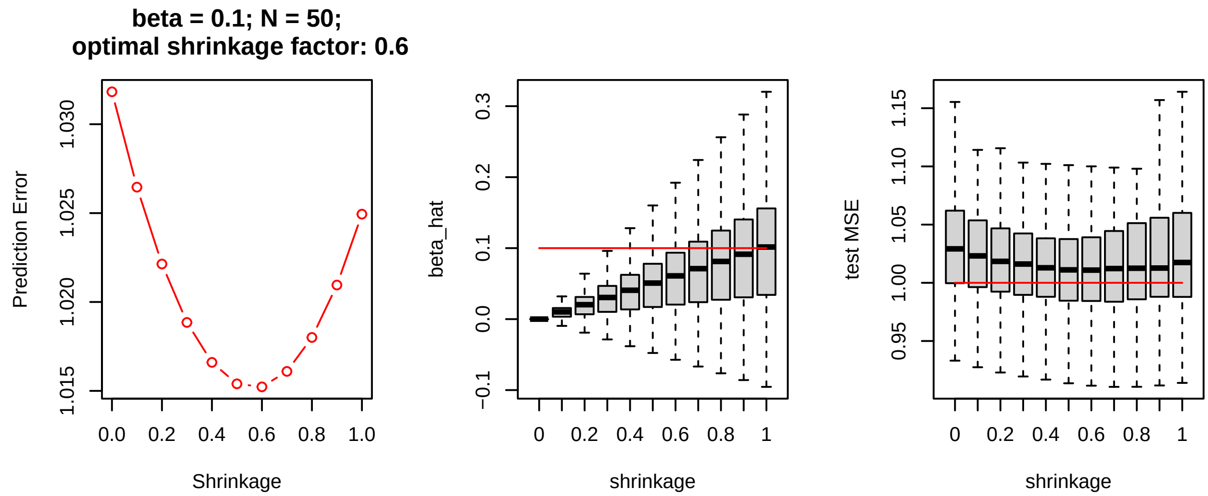
```
beta <- 0.1 # effect size
n <- 50 # sample size
n_reps <- 100 # no. of replications
shrinkage <- seq(0, 1, by = 0.1) # values for shrinkage parameter
error <- beta_hats <- matrix(0, nrow = n_reps, ncol = length(shrinkage))
colnames(error) <- colnames(beta_hats) <- shrinkage # objects for saving results
set.seed(42)
for (i in 1:n_reps) {
  # generate training data:
  x <- runif(n, min = -3, max = 3)
  y <- beta*x + rnorm(n)
  # fit OLS and get parameter estimates:
  fit <- lm(y ~ 0 + x)
  b_ols <- coef(fit)
  # generate test data:
  xtest <- runif(1000, min = -3, max = 3)
  ytest <- beta*xtest + rnorm(1000)
  ## apply shrinkage and obtain predictions:
  for (s in 1:length(shrinkage)) {
    # generate predictions for test observations:
    ypred <- xtest * shrinkage[s] * b_ols
    error[i, s] <- mean((ytest - ypred)^2)
    beta_hats[i, s] <- shrinkage[s] * b_ols
  }
}
```

a) The irreducible error $\sigma^2 = 1$ in this problem, because we generated it from a standard normal distribution. The bias of OLS will be 0 in this problem, because the data are generated from a linear model and we fit an unbiased linear model. It is difficult to predict what the variance is.

b) and d)

```
par(mfrow = c(1,3))
min_id <- which(colMeans(error) == min(colMeans(error)))
## Plot MSE versus shrinkage factor
plot(x = shrinkage, y = colMeans(error), type = 'b',
     col = "red", xlab = "Shrinkage", ylab = "Prediction Error",
     main = paste0("beta = ", beta, "; N = ", n,
                   ";\n optimal shrinkage factor: ",
                   shrinkage[min_id]))
## Plot distributions of beta estimates (add line for true value)
```

```r
boxplot(beta_hats, xlab = "shrinkage", ylab = "beta_hat", outline = FALSE)
lines(c(1, 11), c(0.1, 0.1), col = "red")
## Plot distributions of MSE (add line for irreducible error)
boxplot(error, ylab = "test MSE", xlab = "shrinkage", outline = FALSE)
lines(c(1, 11), c(1, 1), col = "red")
```




```r
## Compute variance of the shrunken and OLS estimates
round(apply(beta_hats, 2, var), digits = 3)
```

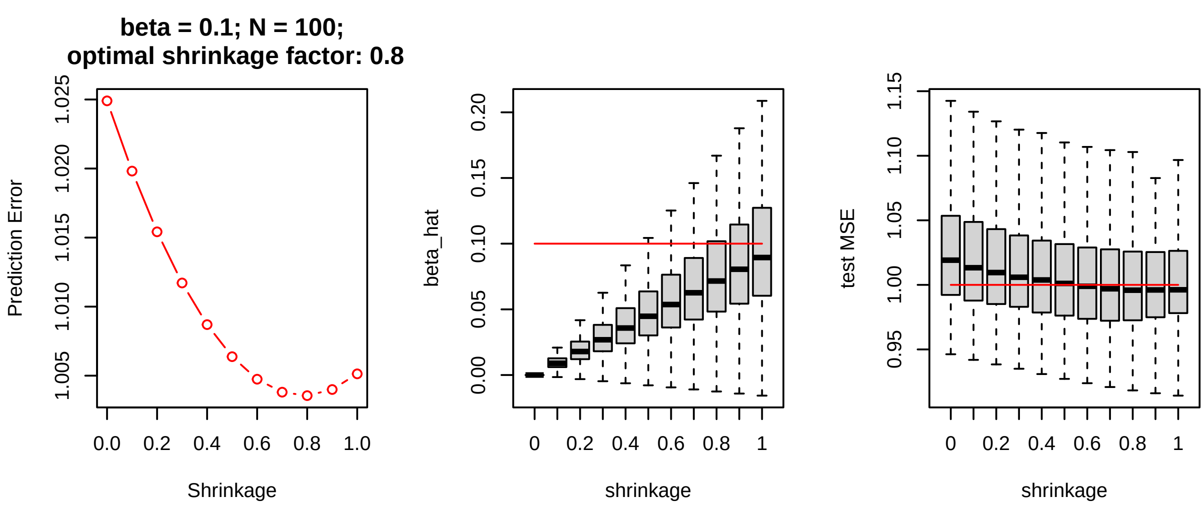
```
##     0   0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9     1
## 0.000 0.000 0.000 0.001 0.001 0.002 0.003 0.004 0.005 0.006 0.007
```

c) and e) With shrinkage, we see that the estimated coefficient $\hat{\beta}$ are biased towards zero (the red line in the middle plot indicates the true value of β), but this also reduces the variance of the estimated coefficients, which is beneficial for prediction. For this specific data problem, a shrinkage factor of 0.6 seems optimal. The red line in the right plot indicates the irreducible error.

d) With larger sample size:

```r
n <- 100 # adjust sample size
set.seed(42)
for (i in 1:n_reps) {
  # generate training data:
  x <- runif(n, min = -3, max = 3)
  y <- beta*x + rnorm(n)
  # fit OLS and get parameter estimates:
  fit <- lm(y ~ 0 + x)
  b_ols <- coef(fit)
  # generate test data:
  xtest <- runif(1000, min = -3, max = 3)
  ytest <- beta*xtest + rnorm(1000)
  ## apply shrinkage and obtain predictions:
  for (s in 1:length(shrinkage)) {
    # generate predictions for test observations:
    ypred <- xtest * shrinkage[s] * b_ols
```

```r
    error[i, s] <- mean((ytest - ypred)^2)
    beta_hats[i, s] <- shrinkage[s] * b_ols
  }
}
par(mfrow = c(1,3))
min_id <- which(colMeans(error) == min(colMeans(error)))
## Plot MSE versus shrinkage factor
plot(x = shrinkage, y = colMeans(error), type = 'b',
     col = "red", xlab = "Shrinkage", ylab = "Prediction Error",
     main = paste0("beta = ", beta, "; N = ", n,
                   ";\n optimal shrinkage factor: ",
                   shrinkage[min_id]))
## Plot distributions of beta estimates (add line for true value)
boxplot(beta_hats, xlab = "shrinkage", ylab = "beta_hat", outline = FALSE)
lines(c(1, 11), c(0.1, 0.1), col = "red")
## Plot distributions of MSE (add line for irreducible error)
boxplot(error, ylab = "test MSE", xlab = "shrinkage", outline = FALSE)
lines(c(1, 11), c(1, 1), col = "red")
```




```r
## Compute variance of the shrunken and OLS estimates
round(apply(beta_hats, 2, var), digits = 3)
```

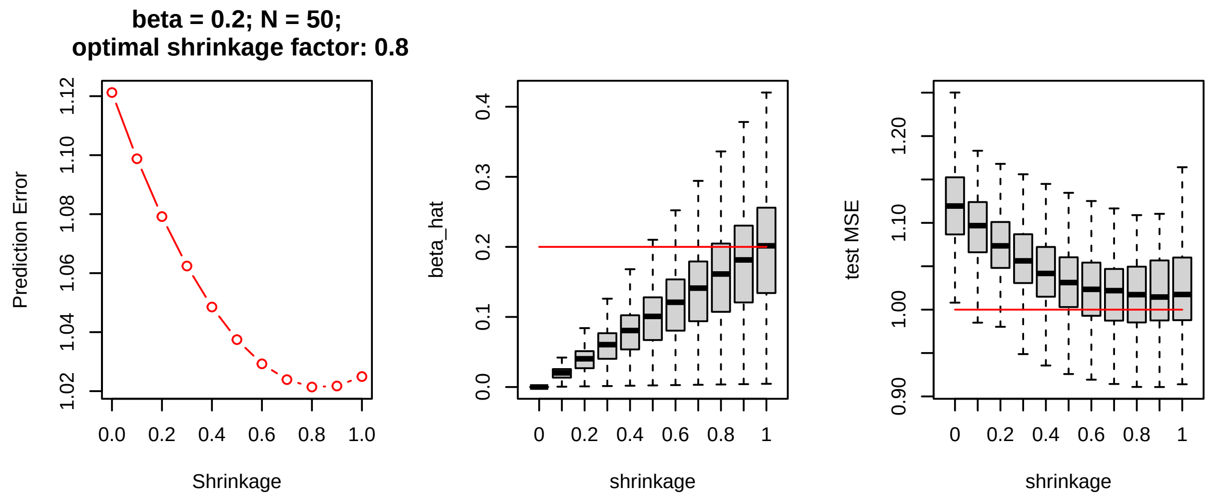
```
##     0   0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9     1
## 0.000 0.000 0.000 0.000 0.001 0.001 0.001 0.002 0.002 0.003 0.003
```

With larger sample size (i.e., more information in the sample), shrinkage is still beneficial. With larger sample size, however, the variance of $\hat{\beta}$ is lower, so we need less shrinkage (bias) to optimize prediction error. For this data problem, a shrinkage factor of 0.8 seems optimal.

g)

```r
beta <- 0.2 # adjust effect size
n <- 50 # set sample size to original value
set.seed(42)
```

```r
for (i in 1:n_reps) {
  # generate training data:
  x <- runif(n, min = -3, max = 3)
  y <- beta*x + rnorm(n)
  # fit OLS and get parameter estimates:
  fit <- lm(y ~ 0 + x)
  b_ols <- coef(fit)
  # generate test data:
  xtest <- runif(1000, min = -3, max = 3)
  ytest <- beta*xtest + rnorm(1000)
  ## apply shrinkage and obtain predictions:
  for (s in 1:length(shrinkage)) {
    # generate predictions for test observations:
    ypred <- xtest * shrinkage[s] * b_ols
    error[i, s] <- mean((ytest - ypred)^2)
    beta_hats[i, s] <- shrinkage[s] * b_ols
  }
}
par(mfrow = c(1,3))
min_id <- which(colMeans(error) == min(colMeans(error)))
## Plot MSE versus shrinkage factor
plot(x = shrinkage, y = colMeans(error), type = 'b',
     col = "red", xlab = "Shrinkage", ylab = "Prediction Error",
     main = paste0("beta = ", beta, "; N = ", n,
                   ";\n optimal shrinkage factor: ",
                   shrinkage[min_id]))
## Plot distributions of beta estimates (add line for true value)
boxplot(beta_hats, xlab = "shrinkage", ylab = "beta_hat", outline = FALSE)
lines(c(1, 11), c(0.2, 0.2), col = "red")
## Plot distributions of MSE (add line for irreducible error)
boxplot(error, ylab = "test MSE", xlab = "shrinkage", outline = FALSE)
lines(c(1, 11), c(1, 1), col = "red")
```




```r
## Compute variance of the shrunken and OLS estimates
round(apply(beta_hats, 2, var), digits = 3)
```

```
##     0   0.1   0.2   0.3   0.4   0.5   0.6   0.7   0.8   0.9     1
## 0.000 0.000 0.000 0.001 0.001 0.002 0.003 0.004 0.005 0.006 0.007
```

With larger effect size (i.e., larger β, i.e., higher signal-to-noise ratio so more information in the sample), shrinkage is still beneficial. With larger effect size, variance of $\hat{\beta}$ remains identical. However, the shrinkage factor has a stronger effect on larger coefficients, so we need less shrinkage (bias) to optimize prediction error. For this data problem, a shrinkage factor of 0.8 seems optimal.

Conclusion: Shrinkage is beneficial for prediction. With higher effect and/or training sample size (i.e., more information in the training data), a lower amount of shrinkage is optimal.