STATS 305A HW # 7

Adhitya Venkatesh

November 13, 2017

Theoretical Problems:

1 and 2 (Done together as they're inter-related):

Given that $y = X\beta + \epsilon$, we have the prior distribution

$$Y|\beta \sim N(X^T\beta, \sigma^2 I)$$

and of course we know

$$\beta \sim N(0, \tau I)$$

Now, because the conjugate prior of the posterior $Y|\beta$ is proportional to the product of the above pdfs by Bayes rule and the conjugate prior is distributed normally, we know that the distribution of the posterior will also be normal. Thus, the mean of the posterior is equivalent to the MLE estimator by virtue of a normal distribution's properties. Hence, we set out to find the MLE of the posterior:

$$f(\beta|Y) \sim f(\beta) * f(Y|\beta)$$

(Here I'm using $\sim$ as a proxy for the "proportional to" symbol due to compilation issues)

$$log(f(\beta|Y)) \sim log(f(\beta) * f(Y|\beta)) = K + log(f(Y|\beta)) + log(f(\beta))$$

where K is the proportionality constant between the LHS and RHS. Substituting the pdfs into the above and denoting the sum of any normalizing constants and terms independent of β as C, we have

$$log(f(\beta|Y)) = C - (y - X\beta)^T(y - X\beta)/2\sigma^2 - \beta^T\beta/2\tau$$

Multiplying by $2\sigma^2$ on both sides, we finally arrive at

$$\beta_{MLE} = argmin(|y - X\beta|^2 + (\sigma^2/\tau)|\beta|^2)$$

This is equivalent to minimizing the exact loss function with a ridge penalty term. Matching coefficient, we obtain the relationship

$$\lambda = \sigma^2/\tau$$

Computational Problems:

1.)

```
library(pls)
```

```
## Warning: package 'pls' was built under R version 3.3.2
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

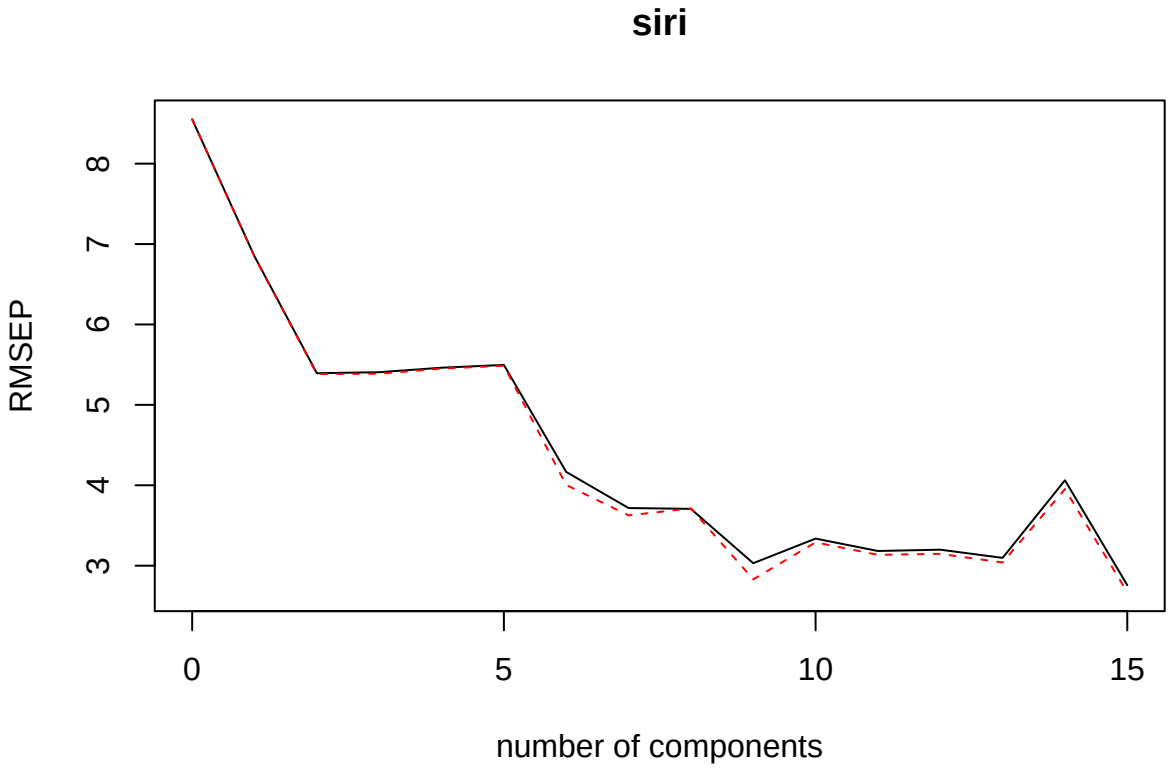
```
fat_data = faraway::fat
test_indices = seq(10, nrow(fat_data) - 2, length.out = 25)

test_data = fat_data[test_indices,]
train_data = fat_data[-test_indices,]
train_data = train_data[,-1]
train_data = train_data[,-2]

pcr_mod = pcr(siri~. , data = train_data, scale = TRUE, validation = "CV")
summary(pcr_mod)
```

```
## Data:    X dimension: 227 15
##  Y dimension: 227 1
## Fit method: svdpc
## Number of components considered: 15
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           8.551    6.847    5.393    5.406    5.462    5.497    4.166
## adjCV        8.551    6.841    5.383    5.388    5.451    5.486    4.004
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       3.717    3.707    3.031     3.337     3.183     3.199     3.097
## adjCV    3.626    3.713    2.830     3.290     3.135     3.147     3.040
##        14 comps  15 comps
## CV        4.062     2.758
## adjCV     3.952     2.670
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X       62.90    73.17    80.41    84.99    88.98    91.43    93.60
## siri    36.65    62.19    63.67    63.71    65.59    82.69    84.05
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X       95.42    96.74     97.95     98.80     99.31     99.62     99.88
## siri    84.06    91.02     91.86     92.52     92.77     93.76     93.81
##       15 comps
## X       100.00
## siri     96.92
```

```
validationplot(pcr_mod)
```



2.)

```
#Refitting the standard linear model with the same dataset
linReg_mod = lm(siri ~., data = train_data)
summary(linReg_mod)
```

```
##
## Call:
## lm(formula = siri ~ ., data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8314 -0.6722  0.1828  0.9150  6.6619
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -12.591885   6.448868  -1.953 0.052193 .
## age           0.007978   0.012320   0.648 0.517983
## weight        0.362999   0.023314  15.570  < 2e-16 ***
## height        0.049026   0.040315   1.216 0.225315
## adipos       -0.514032   0.114074  -4.506 1.09e-05 ***
## free         -0.564773   0.014889 -37.933  < 2e-16 ***
## neck          0.016525   0.089863   0.184 0.854272
## chest         0.120219   0.039590   3.037 0.002694 **
## abdom         0.140108   0.042186   3.321 0.001056 **
## hip           0.006197   0.056101   0.110 0.912148
## thigh         0.195057   0.054460   3.582 0.000424 ***
## knee          0.106637   0.093534   1.140 0.255542
## ankle         0.125118   0.081303   1.539 0.125325
## biceps        0.096199   0.064656   1.488 0.138278
```

```
## forearm       0.230775   0.073332   3.147 0.001888 **
## wrist         0.139279   0.206804   0.673 0.501378
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.55 on 211 degrees of freedom
## Multiple R-squared:  0.9692, Adjusted R-squared:  0.967
## F-statistic: 442.5 on 15 and 211 DF,  p-value: < 2.2e-16
```

```
test_data = test_data[,-1]
test_data = test_data[,-2]
test_x = test_data[,2:16]
test_y = test_data[,1]

#the number of principal components chosen is based on smallest MSE in validation plot
pcr_pred = predict(pcr_mod, test_x, ncomp = 9)
linmod_pred = predict(linReg_mod, test_x)

#MSE for PCR model
mean((pcr_pred - test_y)^2)
```

```
## [1] 3.717102
```

```
#MSE for standard regression model
mean((linmod_pred - test_y)^2)
```

```
## [1] 1.280357
```

The MSE on the test data suggests that the standard linear regression model is better as the MSE is smaller compared to that of the principal components regression model.