STATS315A HW#3 Individual

Adhitya Venkatesh

January 31, 2018

1.)

18.7:

a.)

We define V^c to be a p x (N - p) dimensional matrix that's the orthogonal complement of V. Further, let θ^* be the solution to $R\theta = y$. Now, let $B(\lambda) = V\theta^* + V^c\lambda$ where lambda is an arbitrary vector of N - p dimensions. Notice that

$$\begin{aligned} XB(\lambda) &= RV^T(V\theta^* + V^c\lambda) \\ &= R\theta^* + RV^TV^c\lambda = R\theta^* \end{aligned}$$

Hence, as there exists a solution $B(\lambda)$ for any λ, there are infinite solutions.

b.)

For ridge regression, we have $\beta = (X^TX + \lambda I)^{-1}X^Ty$. Now as $X = RV^T$,

$$\begin{aligned} \beta &= (VR^TRV^T + \lambda VV^T)^{-1}VR^Ty \\ &= V(R^TR + \lambda I)^{-1}V^TVR^Ty \\ &= V(R^TR + \lambda I)^{-1}R^Ty \end{aligned}$$

Here we used the fact that V is orthgonal i.e. $VV^T = I = V^TV$.

c.)

By definition, $residuals = y - X\beta_0^{hat}$

$$\begin{aligned} &= y - (UDV^T)(VD^{-1}U^T)y \\ &= y - (UDD^{-1}U^T)y = y - Iy = 0 \end{aligned}$$

Consider some solution β^* with 0 residuals. Then,

$$||\beta^*||^2 = ||\beta^* - \beta_0^{hat} + \beta_0^{hat}||^2 = ||\beta^* - \beta_0^{hat}||^2 + ||\beta_0^{hat}||^2 + 2\beta_0^{hat}(\beta^* - \beta_0^{hat})$$

Now, we evaluate the last term $\beta_0^{hat}(\beta^* - \beta_0^{hat})$:

$$\begin{aligned} &= y^TUD^{-1}V^T(\beta^* - VD^{-1}U^Ty) = y^TUD^{-2}U^TUDV^T\beta^* - y^TUD^{-2}U^Ty \\ &= y^TUD^{-2}U^TX\beta^* - y^TUD^{-2}U^Ty = 0 \end{aligned}$$

Hence, the Euclidean norm of β^* is minimized when $\beta^* = \beta_0^{hat}$.

18.8:

a.)

Note that if β^{hat} is a vector onto which x is projected, then the distance of the projected point to the origin is given by the formula $\beta^{hat,T}x/||\beta^{hat}||$ (the first term in numerator being transpose of beta hat).Hence, any

solution to $X\beta = y$ will project onto precisely two points and by the same argument presented in 18.7a.), we conclude there are infinitely such points/solutions.

b.)

The fitted values β^{hat} i.e. $X\beta^{hat}$ yield residuals of 0. Hence, the signed distance, by the formula mentioned in previous part, for both points are $1/||\beta^{hat}||$ with opposite signs. Therefore the distance between these points is $2/||\beta^{hat}||$.

c.)

From b.), it's clear that distance is maximized when the Euclidean norm $||\beta^{hat}||$ is minimized. Using the result from 18.7c.), we know that this occurs when $\beta = \beta_0^{hat} = VD^{-1}U^Ty$.

18.9:

```
#We start by generating data(20 points, 40 dimensions)

x = matrix(rnorm(800), ncol = 40)
y = rep(c(-1,1), times = c(10,10))

x_svd = svd(x)
b_0 = x_svd$v%*%diag(1/x_svd$d)%*%t(x_svd$u)
x_project = x%*%b_0/sqrt(sum(b_0^2))
pile_dist = max(x_project[y==-1]) - min(x_project[y==1])
pile_dist
```

```
## [1] 0.9763949
```

```
library(e1071)
mod_svm = svm(x, y, scale = FALSE, type = 'C-classification', kernel = "linear", cost = 10^6)

b_svm = t(mod_svm$SV)%*%mod_svm$coefs
b_svm = b_svm/sqrt(sum(b_svm^2))
x_svm_project = x%*%b_svm
svm_dist =  max(x_svm_project[y==-1]) - min(x_svm_project[y==1])
svm_dist
```

```
## [1] 5.555306
```

It's apparent that the distance for the SVM is greater than the piling distance as expected.

2.)

a.)

Via integration by parts, we have

$$\int_a^b g^{''}(x)h^{''}(x)dx = [g^{'''}(x)h^{'}(x)]_a^b - \int_a^b g^{'''}(x)h^{'}(x)dx$$

Now as $g^{'''}(x) = 0$ for all $x \leq a, x \geq b$, we are simply left with the 2nd term on the RHS:

$$\int_a^b g^{'''}(x)h^{'}(x)dx = \sum_{i=1}^{N-1} \int_{x_i}^{x_{i+1}} g^{'''}(x)h^{'}(x)dx$$

Again using integration by parts, we have

$$= \sum_{i=1}^{N-1} [g^{'''}(x)h^{'}(x)]_{x_i}^{x_{i+1}} - \sum_{i=1}^{N-1} \int_{x_i}^{x_{i+1}} g^{(4)}(x)h(x)dx$$

Now since g is a cubic function and by definition of h, $h(x_i) = 0$ for all $1 \leq i \leq N$, $g^{'''}(x) = 0$, and $g^{(4)}(x) = 0$. Hence, all terms become 0 and we have

$$\int_a^b g^{''}(x)h^{''}(x)dx = 0$$

b.) This integral becomes...

$$= \int_a^b (g^{''}(t) + h^{''}(t))^2 dt = \int_a^b (g^{''}(t))^2 + (h^{''}(t))^2 dt + 2\int_a^b g^{''}(t)h^{''}(t)$$

Utilizing the result from the previous part, we know the 2nd term on the RHS is 0. Hence,

$$\int_a^b (g_{tilde}^{''}(t))^2 \geq \int_a^b (g^{''}(t))^2$$

In order for equality to hold, $h^{''}(t) = 0$, implying $h(t)$ is a linear function. However, as $h(x_i) = 0$ for all $1 \leq i \leq N$, $h(t) = 0$ for all $a \leq t \leq b$.

c.)

To prove this, suppose there exists some function g that is the minimizer of

$$\sum_{i=1}^{N} (y_i - f(x_i))^2 + \lambda \int_a^b (f^{''}(t))^2 dt$$

The squared-error of a natural cubic spline, say h, with knots at the same x_i that interpolates $g(i)$ for all i will be the same as that for g. Further, from the previous parts, we note that the penalty term of this natural cubic spline cannot exceed that of g. This implies that the total loss with h is not greater than that with g. Hence, g = h.

3.)

a.), b.)

The equivalence is simple to see once we realize that:

$$X^*\beta^* = XQQ^T\beta = X\beta$$

Additionally,

$$\beta^{*T}\beta^* = \beta^T QQ^T\beta = \beta^T\beta$$

From these relations, it's apparent that the problems are equivalent. Thus, we have proven equivalence under orthogonal transform of X.

c.)

We have $X^T = [R_1^T Q_1^T, 0]$. Now, by equation (10) in the paper "Efficient Quadratic Regularization for Expression Arrays" (Tibshirani, 2004), we see that the optimization problem is split into two distinct parts and each can be separately optimized. One of the parts corresponds to running just R_1. However, the second part simplifies to minimizing $\beta_2^T\beta_2$, which occurs when $\beta_2 = 0$. Thus, the logistic regression can be solved by just running R_1 and then multiplying the result by Q_1.

d.)

As lambda tends to 0, the optimization problem reduces to that of standard logistic regression with only the binomial log-likelihood term. Hence, it will yield the same solution as running X on standard logistic regression.

e.)

4.)

a.)

We have for $y_i = x_i^T\beta + f(z_i) + \epsilon_i$ for all $i = 1, ..., n$ and unpenalized β

$$RSS(f, \beta) = \sum_{i=1}^{N}(y_i - f(z_i) - x_i^T\beta)^2 + \int (f^{''}(t))^2 dt$$

b.)

Setting the partial derivative of $RSS(f, \beta)$ with respect to both β and f to 0 and solving, we obtain the normal equation

$$X\beta^* = H(y - f^*)$$

Additionally, we arrive at the smoothing spline equation

$$f^* = S_\lambda(y - X\beta^*)$$

Further, H is the basis matrix, which in this case is...

c.)

As we have two variables f^*, β^* and two equations from b.), we can solve by substituting the expression for f^* in the first equation into the second equation, yielding:

$$S_\lambda(y - X\beta^*) = y - H^{-1}X\beta^*$$

From this we obtain

$$\beta^* = X^{-1}(H^{-1} - S_\lambda)^{-1}(I - S_\lambda)y$$

Upon substition, we find

$$f^* = (I - H^{-1}(H^{-1} - S_\lambda)^{-1}(I - S_\lambda))y$$

d.)

Yes this simplifies the problem as the loss function no longer includes the roughness penalty. To compute β^* in this case, we simply fix the location of the knots and process to directly minimize $\sum_{i=1}^{N}(y_i - f(z_i) - x_i^T\beta)^2$ for f^*, β^*.

e.)

This regression coefficient makes intuitive sense considering that the expression is smoothing out the y vector and then projecting onto the subspace.

f.)

The best approach from a computational standpoint is to first solve for β^* from the determined equation and then substitute into expression for f^*. The runtime will be dominated by matrix inversion step, which requires $O(np^3)$ runtime.

g.)

We have from substitution into equation $y_i^* = x_i^T \beta^* + f(z_i)$ to obtain

$$y^* = ((H^{-1} - S_\lambda)^{-1}(I - S_\lambda) + I - H^{-1}(H^{-1} - S_\lambda)^{-1}(I - S_\lambda))y$$

Hence,

$$M = (H^{-1} - S_\lambda)^{-1}(I - S_\lambda) + I - H^{-1}(H^{-1} - S_\lambda)^{-1}(I - S_\lambda)$$

h.)

Here we use the relationship

$$cov(Ay) = Acov(y)A^T$$

and substitute $X^{-1}(H^{-1} - S_\lambda)^{-1}(I - S_\lambda)$ for A and σ^2 for cov(y) and proceed to solve for conditional covariance of β^*. Similarly, we substitute $I - H^{-1}(H^{-1} - S_\lambda)^{-1}(I - S_\lambda)$ for A and still σ^2 for cov(y) to obtain conditional covariance of f^*.

5.)

a.)

In order to fit a natural cubic spline to create a time-varying coefficient model, we determine the paramters that minimize the sum of squared error loss and roughness penalty. From here, we use cross-validation to determine the optimal hyperparamter choice for λ.

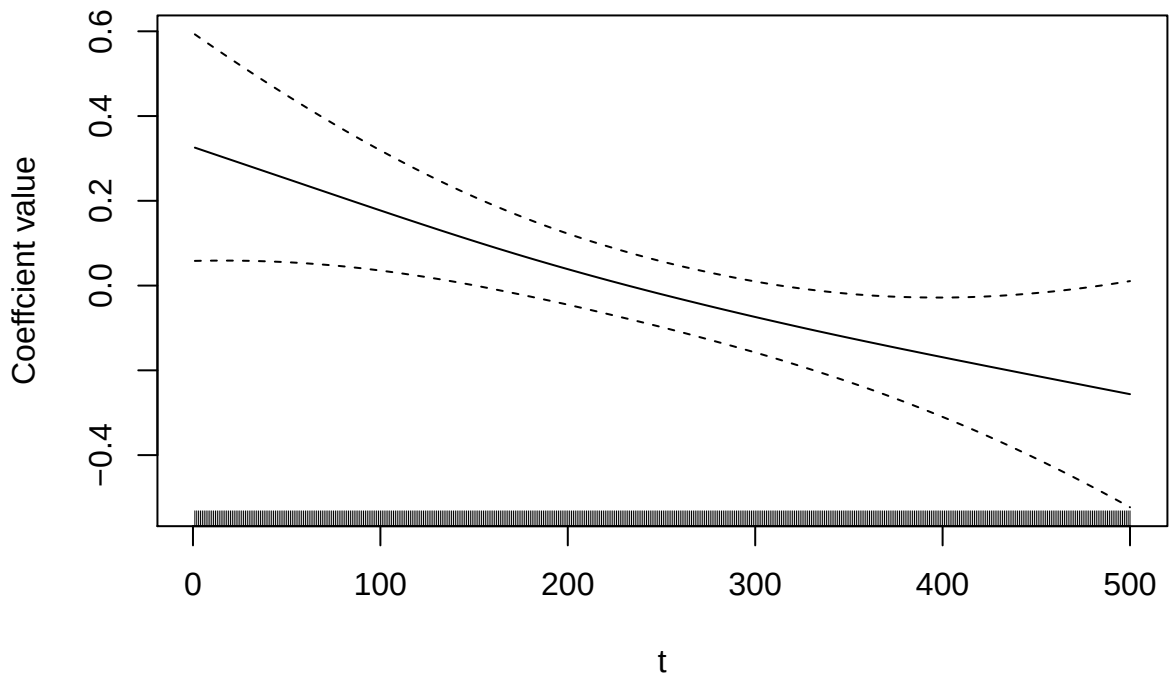
b.)

```
file_loc = "/Users/Adi/Documents/vcdata.csv"
data = read.csv(file_loc)
library(mgcv)
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-23. For overview type 'help("mgcv-package")'.
```

```
model_0 = y ~ s(t, bs = 'cr') + x.1 + x.2 + x.1:x.2
fit_0 = gam(model_0, data = data)
plot.gam(fit_0, ylab = "Coeffcient value", main = "Varying Coefficient Model")
```



c.)

```
se = sqrt(diag(vcov(fit_0)))
print(se)
```

```
## (Intercept)         x.1         x.2     x.1:x.2      s(t).1      s(t).2
##  0.06658536  0.06960549  0.06792946  0.07270743  0.05284906  0.04260094
##      s(t).3      s(t).4      s(t).5      s(t).6      s(t).7      s(t).8
##  0.04826326  0.06028322  0.07206317  0.08466376  0.09916000  0.12903411
##      s(t).9
##  0.14291256
```

d.)

```
summary(fit_0)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## y ~ s(t, bs = "cr") + x.1 + x.2 + x.1:x.2
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.00716    0.06659  45.162   <2e-16 ***
## x.1          1.91426    0.06961  27.502   <2e-16 ***
## x.2         -0.07948    0.06793  -1.170    0.243
## x.1:x.2      0.06044    0.07271   0.831    0.406
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
```

```
##        edf Ref.df     F p-value
## s(t) 1.265  1.482 5.053  0.0265 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.609   Deviance explained = 61.2%
## GCV = 2.2184  Scale est. = 2.1951    n = 500
```

Although the data was fit to a natural cubic spline regression with an interaction term, we can see from the summary of the model that the interaction term is not statistically significant.