MATH HW2 Solutions

Problem 1:

1. In order to prove that αT ^β is an unbiased estimator of θ, it is necessary to show that the expected value of T ^β is equivalent to θ.

The Least Squares estimate, denoted by ^β, is obtained by minimizing the sum of squared residuals, and it can be expressed as the value of ^β as

^β = (X’X)-1(X’Y)

By taking the expected value of both sides, we obtain the expression for the expected value of the Least Squares estimate

E\*^β = E[(X’X)-1(X’Y)] = (X’X)-1(X’E[Y])

Since the expected value of Y is β\*X

E[^β] = (X’X)-1(X’ βX) = β

E[αT ^β] = (αT)E[^β] = αTβ = θ

2. In order to demonstrate that the variance of αT ^β is not greater than the variance of cTy. we need to compare the variances of the two variables and show that the variance of αT ^β is less than or equal to the variance of cTy.

i.e Var[αT ^β] <= Var[cTy]

The variance of αT ^β =

Var[αT ^β] = αT(X’X)-1X’ Var[E] (X(X’X)-1(X’)-1α

Where Var[E] is an identity matrix.

The variance of cTy =

Var[cTy] = cTVar[y]c = cTX\* Var[β]X’c

The Least Squares estimate ^β is referred to as the Best Linear Unbiased Estimator of the true regression coefficient, as it is both unbiased and has the smallest variance among all unbiased linear estimators.

Var[αT ^β] = αT(X’X)-1X’ Var[E] (X(X’X)-1(X’)-1α. <= cTX

Var[αT ^β] = Var[cTy]

we can conclude that the variance of αT ^β is less than or equal to the variance of cTy.

Problem 2:

1. The main difference between the two methods is that the first method forms a set of points such that there is 95% confidence that the predicted value ˆf(x0) is within that set, while the second method provides a 95% confidence interval for an arbitrary point. This reflects the distinction between a pointwise approach and a global confidence estimate. The pointwise approach involves estimating the variance of individual predictions.

σ02 = Var(f^(x0)|x0)

= Var(X0T ^β|x0)

= X0T Var(^β)x0

= ^ σ02 X0T(XTX)-1x0

R code:

library(reshape2)

simulation.xs <- c(1949, 1950, 1941, 1942, 1943, 1944, 1945, 1946, 1947, 1948, 1939)

simulation.ys <- c(4567, 4865, 5005, 5245, 5290, 5469, 5298, 5225, 5445, 5275, 5700)

simulation.df <- data.frame(pop = simulation.ys, year = simulation.xs)

# Rescale years

simulation.df$year <- simulation.df$year - 1946

# Generate regression, construct confidence intervals

fit <- lm(pop ~ year + I(year^2) + I(year^3), data = simulation.df)

xs <- seq(-5, 5, 0.1)

fit.confidence <- predict(fit, data.frame(year = xs), interval = "confidence", level = 0.95)

# Create data frame containing variables of interest

df <- as.data.frame(fit.confidence)

df$year <- xs

df <- melt(df, id.vars = "year")

# Create the plot

p <- ggplot() +

geom\_line(aes(x = year, y = value, colour = variable), df) +

geom\_point(aes(x = simulation.df$year, y = simulation.df$pop)) +

scale\_x\_continuous("Year") +

scale\_y\_continuous("Population") +

ggtitle("Cubic regression with confidence intervals") +

scale\_color\_brewer(name = "Legend", labels = c("Fit", "95% Lower Bound", "95% Upper Bound"), palette = "Set1")

print(p)

result:

Chart, line chart

Description automatically generated

Problem 3:

a. Best Subset (M)

In the orthogonal case, the matrix XTX is an identity matrix (I).

For the best subset (size M), we can write X as X = X.I.

The estimator ^β is given by ^β = (XY)(XTX)-1 = XY.

Using the concepts of QR decomposition, for each step q, we choose K such that:

K = argmax(XTkY) where q < k <= p

which is equivalent to K = argmax(^β) where q < k <= p

The best subset with k(M) predictors gives the smallest residual sum of squares, which is equivalent to finding the largest M(k) coefficient.

rj = (Y- xj ^βj)T (Y- xj ^βj)

= YTY -2^βj xjTy + ^βj2

= YTY -2(xjTy)2 + xjTy2

= YTY - |^βj|2

Which can be minimized by having |^βj| as large as possible.

b. Ridge Regression

^β = (XTY)(XTX +λI)-1

= (XTY)(1/1+λ)

= (^β)(1/1+λ)

c. Lasso

min ½ |y- xβ|22 + λ|β|1

Taking derivative above equations and putting ^β != 0

Which gives -xjT (Y-xj^β) + sign(^β) λ =0

|^β|=1 of ^β > 0 else -1

^β = XjTY – sign(^β) λ

^β =^βj – sign(^β) λ

At this point we have two scenarios

1. If sign(^β) < 0, then ^βj + λ > 0

Here, Here, lasso estimation is given by ^β = ^βj – λ = sign(|^βj|+ λ)(^β)

2. If sign(^β) > 0, then ^βj – λ > 0

Here, lasso estimation is given by ^β = ^βj – λ = sign(|^βj|- λ)(^β)

Problem 4:

To obtain the least square estimate of the coefficient β1, we can minimize the sum of squared residuals S, which is given by the equation:

S = ∑(Yi - β1Xi)^2

By taking the derivative of S with respect to β1 and setting it to zero, we can solve for β1 as follows:

dS/dβ1 = 2∑ -Xi(Yi - β1Xi) = 0

The resulting value of β1 is given by:

β1 = ∑(XiYi)/∑(Xi^2)

To demonstrate that the vector (Y - Ŷ) is orthogonal to the vector X for the training set (X, Y), we start with the definition of orthogonality:

(Y - Ŷ)X = 0

Expanding the dot product, we get:

(Y - β1X)·X = Y·X - β1X·X = Y·X

As β1 is a constant, the last term simplifies to: YX = 0

This implies that the vector (Y - Ŷ) is orthogonal to the vector X.

Extra Credit:

To show that SSE/σ^2 is distributed as a chi-squared random variable with N-p-1 degrees of freedom, we first note that SSE can be written as:

SSE = (Y - Xβ)^T(Y - Xβ)

where Y is the N × 1 vector of responses, X is the N × (p + 1) design matrix with the first column being all ones, and β is the (p + 1) × 1 vector of parameters to be estimated. The hat over the Y indicates that it is the predicted value of Y from the model.

Expanding SSE, we get:

SSE = Y^TY - 2β^TX^TY + β^TX^TXβ

Taking the expectation of SSE, we have:

E[SSE] = E[Y^TY] - 2β^TX^TE[Y] + β^TX^TXβ

Since E[Y] = Xβ, we can simplify this to:

E[SSE] = E[Y^TY] - β^TX^TXβ

Now, since Y ~ N(Xβ, σ^2I), we have:

E[Y^TY] = E[(Xβ + ϵ)^T(Xβ + ϵ)] = E[β^TX^TXβ] + σ^2N

Substituting this into the above equation, we get:

E[SSE] = σ^2N

Next, we compute the covariance matrix of the residuals e = Y - Xβ:

Cov(e) = E[ee^T] - E[e]E[e^T]

Since E[e] = 0, this simplifies to:

Cov(e) = E[ee^T]

Now, we can write SSE/σ^2 as:

SSE/σ^2 = e^Te/σ^2

Taking the transpose of both sides, we have:

(SSE/σ^2)^T = e^T(e^T)^T/σ^2 = e^Te/σ^2

Therefore, SSE/σ^2 is a symmetric matrix, and we can use the spectral decomposition to show that it is distributed as a chi-squared random variable with N-p-1 degrees of freedom.

Let λ\_1, λ\_2, ..., λ\_N be the eigenvalues of X^TX. Then, X^TX can be decomposed as X^TX = QΛQ^T, where Q is an orthonormal matrix whose columns are the eigenvectors of X^TX, and Λ is a diagonal matrix whose diagonal entries are the eigenvalues.

Now, consider the matrix Q^TY. Since Q is orthonormal, Q^TY has the same distribution as Y. Also, since Q is orthonormal, QQ^T = I, so we have:

SSE = (Y - Xβ)^T(Y - Xβ) = (Q^TY - Λβ)^T(Q^TY - Λβ)

Expanding this out, we get:

SSE = Y^TY - 2β^TΛ^TQ^TY + β^TΛ^TΛβ

SSE = (Q^TY)^T(Q^TY) - 2β^TΛ^T(Q^TY) + β^TΛ^TΛβ

Now, note that Q^TY ~ N(0, σ^2I) and Q^TQ = I. Therefore, the vector Q^TY has independent components that are normally distributed with mean 0 and variance σ^2. Since the distribution of Q^TY is invariant