Jared Yu

Module 4 Assignment

Data Mining

1. Show that, if , are jointly Gaussian, the regression of on is given by

Ans: (Reference: [1], [2])

From the question, it can be seen that and follow a bivariate normal distribution,

where

This is the joint p.d.f. of and . The conditional distribution is as follows:

The term in the denominator is the marginal density for , which can be shown as follows:

Then the conditional distribution can be shown as follows:

*Note: Let and .*

In the above equation, , is the mean of the conditional density, since it can be seen that the above equation is also a normal density with mean and variance . Therefore, it follows that the expectation of the conditional density . To simplify,

where .

1. Given data , consider the regression through the origin model
   1. Find , the least squares estimate for .

Ans:

Let be the prediction of based on , the value of , then is the residual:

Then, define the residual sum of squares to be

The goal then is to find to minimize . Taking the partial derivative:

where

* 1. Find the standard error of the estimate, .

Ans:

Next, the standard error will be shown.

Then, the standard error can be shown as,

Two major assumptions are made. The first is that is a fixed variable as explained in part a), which allows to be treated as a constant within the variance. The other assumption is that . This second assumption will be explained as follows:

In the above steps, is considered constant and so it zeros out within the variance (e.g., for constants , and random variable ). In other words, the sum of the variance is equal to the variance of the sums.

* 1. Find conditions that guarantee that the estimate is consistent:

Ans: (Reference: [3])

In the above problem, the constant is shown as the probability limit of the sequence, which will be abbreviated as . So, the statement can be rewritten as,

The statement is saying that the probability of being able to differ from by some arbitrary finite value tends towards zero as grows towards . This problem is looking to understand the asymptotic properties of the estimator . First, will be restated as follows,

The expectation of , is as follows,

In the above steps, an assumption is that since . It can be seen that as , the expectation of remains unchanged at .

From part b), it has been shown that,

The numerator is a constant, . The denominator is a summation of terms indexed by for . Then it can be seen that as , .

Therefore, it has been shown that since as that the of approaches its true value , then it is a consistent estimator.

1. The columns in the file polynomial\_data.txt are the and values of a polynomial function with added Gaussian noise.
   1. For each , fit an degree polynomial to the data. What would you say is the most likely value of ?

Ans:

The first step was to create the relevant design matrices, , for each . This involves creating a column of ones, , to represent the intercept term in the model. The terms are created simply by taking to the power for . The design matrices are then simply the concatenation of the column and the corresponding variables.

The next step is to calculate the term, which is as follows,

Then, the corresponding estimates , can be calculated as follows,

From here, the is calculated, which has the following formula,

where , , , , (where here refers to the number of observations in the dataset i.e. 50), and is the number of parameters (e.g., for polynomial , , etc.).

Based on the above formulas, they were plugged into RStudio to calculate the for each of the degree polynomial models.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Table 1 The above table shows the corresponding for each of the degree polynomial models.

From the above table, it can be seen that has the best . However, comparing the three models, the jump from to seems relatively large. The difference between and though is comparatively quite small. Since adding more degrees to the model increases the complexity of the model that is based on the sample dataset, the generalization ability tends to weaken. Therefore, it seems that the most likely value of is 4.

* 1. Estimate the level of the noise.

Ans:

With linear regression, often times it is assumed that the random error, is normally distributed with mean zero and some unknown variance , i.e., . To estimate the level of noise in the dataset, then we can try to look at this random error term based on the polynomial model calculated in part a). We can do this by analyzing the residuals, , from the sample data. These terms are analogous to the random error, .

First, we can try to check the normality of these residuals using a Q-Q plot (a.k.a. quantile-quantile plot). Below in Figure 1, it shows that the residuals based on the model fall quite closely on the line. The sorted residuals are plotted against a normal distribution with mean and standard deviation based on the residuals that are derived from the sample data. This seems to indicate that there’s a strong probability that the residuals themselves are normally distributed, with mean equal to the mean of the residuals and variance equal to the variance of the residuals.

Chart, scatter chart

Description automatically generated

Figure 1 The above figure shows a Q-Q plot of the residuals from the degree polynomial model. They are compared to a normal distribution with a mean and standard deviation based on the set of residuals. Additionally, a fitted line is plotted through the data to show how linear the data are.

The mean and variance calculated from the sample of residuals, , can be seen in the table below. The sample mean is quite close to zero, and the sample variance is roughly 2,922.326. The calculations used are,

for the sample mean and sample variance respectively. These are then used as the unbiased estimates, for the population mean () and population variance (). Therefore, it is estimated that the level of the noise is , where is estimated with and is estimated with .

|  |  |
| --- | --- |
|  |  |
|  |  |

Table 2 The above table shows the sample mean and sample variance for the residuals, , calculated from the sample data using the polynomial model with degree .

**References**

[1] Montgomery, D.C., Peck, E. A., & Vining, G. G. (2012). *Introduction to Linear Regression Analysis* (5th ed.). Wiley.

[2] Piazza post 109. Can be found at <https://piazza.com/class/kc0jkwru805u1?cid=109>.

[3] Dougherty, C. (2016). Introduction to Econometrics (5th ed.). Oxford University Press.

**Code Appendix**

# Load data  
polynomial\_data <- read.csv('polynomial\_data.txt', header = FALSE, sep = "")  
colnames(polynomial\_data) <- c("X", "Y")  
  
# Create initial variables  
n <- nrow(polynomial\_data)  
X <- polynomial\_data[,1]  
Y <- polynomial\_data[,2]  
  
# Calculate the nth degree for X  
nth\_degree <- function(x, n) {  
 return(x^n)  
}  
  
x2 <- nth\_degree(x = X, n = 2)  
x3 <- nth\_degree(x = X, n = 3)  
x4 <- nth\_degree(x = X, n = 4)  
x5 <- nth\_degree(x = X, n = 5)  
  
# Create the matrices for the nth degree polynomials  
ones <- rep(1, n)  
X3 <- cbind(ones, X, x2, x3)  
X4 <- cbind(ones, X, x2, x3, x4)  
X5 <- cbind(ones, X, x2, x3, x4, x5)  
  
# Calculate the beta hat  
beta\_hat <- function(X, y) {  
 beta\_hat <- solve(t(X) %\*% X) %\*% t(X) %\*% y  
 return(beta\_hat)  
}  
  
beta\_hat\_3 <- beta\_hat(X = X3, y = Y)  
beta\_hat\_4 <- beta\_hat(X = X4, y = Y)  
beta\_hat\_5 <- beta\_hat(X = X5, y = Y)  
  
# Calculate the nth degree y-hat  
y3 <- X3 %\*% beta\_hat\_3  
y4 <- X4 %\*% beta\_hat\_4  
y5 <- X5 %\*% beta\_hat\_5  
  
# Calculate the adjusted R-squared  
adj\_r\_squared <- function(X, betahat, y) {  
 y\_bar <- mean(y)  
 y\_hat <- X %\*% betahat  
  
 SS\_tot <- sum((y - y\_bar)^2)  
 SS\_res <- sum((y - y\_hat)^2)  
   
 n <- length(y)  
 p <- ncol(X) - 1  
 df\_e <- n - p - 1  
 df\_t <- n - 1  
   
 adj\_r\_square <- 1 - ((SS\_res / df\_e) / (SS\_tot / df\_t))  
 return(adj\_r\_square)  
}  
  
adj\_r\_squared\_3 <- adj\_r\_squared(X = X3, betahat = beta\_hat\_3, y = Y)  
adj\_r\_squared\_4 <- adj\_r\_squared(X = X4, betahat = beta\_hat\_4, y = Y)  
adj\_r\_squared\_5 <- adj\_r\_squared(X = X5, betahat = beta\_hat\_5, y = Y)  
  
### Chosen model: n=4  
y\_hat\_4 <- X4 %\*% beta\_hat\_4  
e\_4 <- Y- y\_hat\_4  
mean(e\_4)  
sd(e\_4)  
  
sorted\_residuals <- sort(e\_4)  
e\_quantile <- qnorm((1:n)/n, mean = mean(e\_4), sd = sd(e\_4))  
plot(e\_quantile, sorted\_residuals, main = 'Q-Q Plot',  
 xlab = latex2exp::TeX('$Normal(\\bar{e}, \\sigma\_{e}^2)$'),  
 ylab = latex2exp::TeX('Sorted Residuals'))  
df <- data.frame(x = e\_quantile, y = sorted\_residuals)  
fit <- lm(formula = y~x, data = df[1:(n-1),])  
abline(fit)