Simulation of Confidence Intervals on a Cubic Equation

Lucy L.

2023-12-30

This write-up is derived from the textbook “The Elements of Statistical Learning” from Chapter 3, exercise 3.2:

“Given data on two variables X and Y, consider fitting a cubic polynomial regression model $f(X) = \sum_{j=0}^{3} \beta_j X^j$. In addition to plotting the fitted curve, you would like a 95% confidence band about the curve. Consider the following two approaches:

(a) At each point x_0, form a 95% confidence interval for the linear function $a^T\beta = \sum_{j=0}^{3} \beta_j x_0^j$;

(b) Form a 95% confidence set for β as in (3.15), which in turn generates confidence intervals for $f(x_0)$.

How do these two approaches differ? Which band is likely to be wider? Conduct a small simulation experiment to compare the two methods.”

The following code conducts said simulation experiment. First, 10 values are drawn uniformly from [0,1]. Then $y_i = 1 + x_i + 2x_i^2 + 3x_i^3 + \epsilon_i$ with ϵ_i following the $N(0, 0.5)$ distribution are generated.

```
library(MASS)
library(ggplot2)
set.seed(302)

n <- 10
sigma <- sqrt(0.5)

# prepare data
ones <- rep(1, n)
x <- sort(runif(n))
x_square <- x^2
x_cubic <- x^3

X <- cbind(ones, x, x_square, x_cubic)
X_T <- t(X)

epsilon <- rnorm(n, mean = 0, sd = sigma)

beta <- c(1, 1, 2, 3)
y_theory <- X %*% beta
y_realized <- y_theory + epsilon

beta_hat <- solve(t(X) %*% X) %*% t(X) %*% y_realized
y_estimated <- X %*% beta_hat
```

Following the first method, confidence regions are generated. The variance of $\hat{y_0}$ is $Var(\hat{y_o}) = x_0(X^TX)^{-1}x_0^T$. Thus at each sample pair, the confidence interval is calculated as $\hat{y_0} \pm 1.96\sqrt{x_0(X^TX)^{-1}x_0^T}$.

```r
# method 1
var_beta_hat <- solve(t(X) %*% X) * (sigma^2)
tmp <- X %*% var_beta_hat %*% t(X)
width <- sqrt(diag(tmp))
width_upper <- y_estimated + 1.96 * width
width_lower <- y_estimated - 1.96 * width
```

For the second method, 100 different vectors are sampled, so 100 different $\hat{\beta}$'s are obtained.

```r
# method 2
U_T <- chol(t(X) %*% X)
U <- t(U_T)
U_inv <- solve(U)

p <- 0.95
df <- 4
num <- 100

region_arr <- matrix(0, nrow = num, ncol = n)

for (i in 1:num) {
  a <- mvrnorm(n = 1, mu = rep(0, df), Sigma = diag(df))
  a2 <- solve(U) %*% a
  a_norm <- sqrt(sum(a2^2))

  r <- sigma * sqrt(qchisq(p, df))
  a3 <- (a2 * (r / a_norm))

  beta2 <- beta + a3
  region <- X %*% beta2
  region_arr[i, ] <- region
}


df_plot <- data.frame(x = x, y_estimated = y_estimated,
                      width_upper = width_upper,
                      width_lower = width_lower)

for (i in 1:num) {
  df_plot[paste0("region", i)] <- region_arr[i, ]
}
```

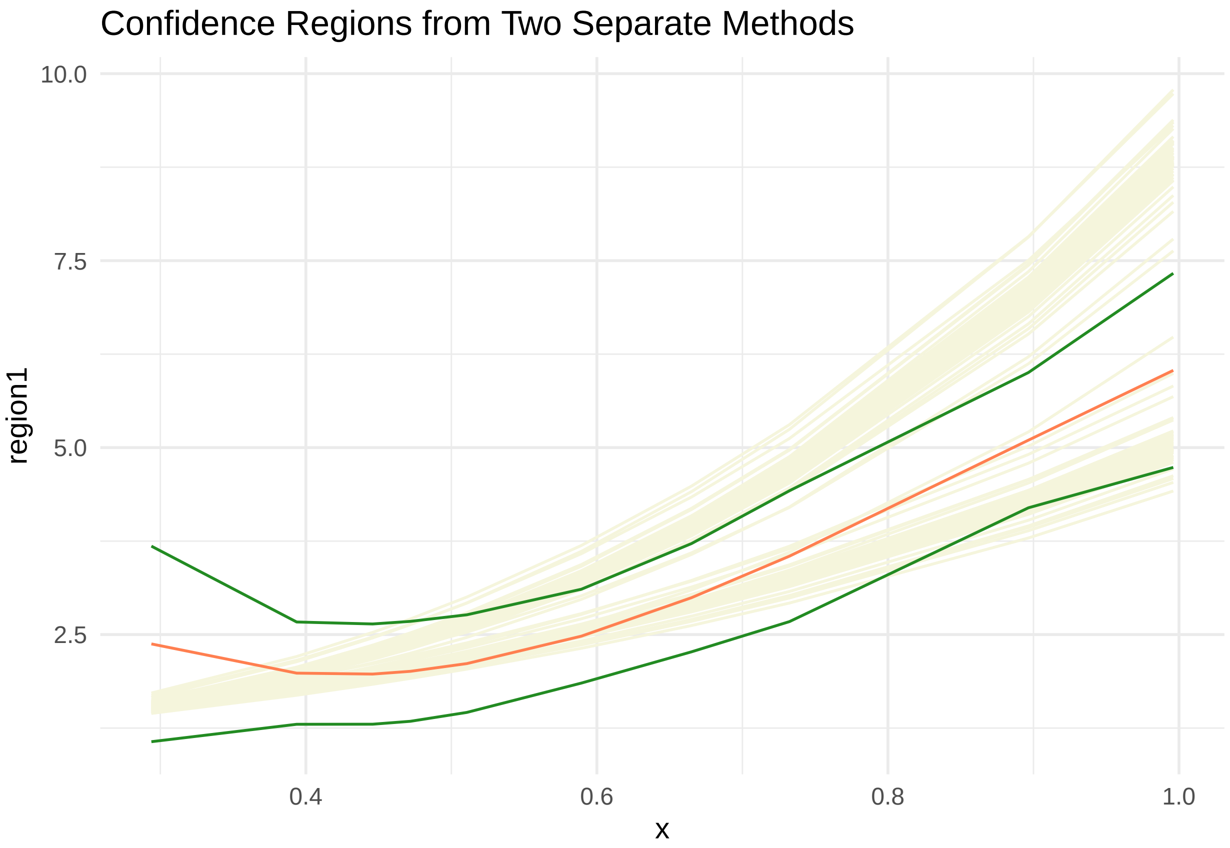
Finally, the representations of the two methods are plotted.

```r
# Plot setup
plot_initial <- ggplot(df_plot, aes(x = x))

# Adding dynamic layers for region_arr
for (i in 1:100) {
  region_col <- paste0("region", i)
  plot_initial <- plot_initial +
    geom_line(aes_string(y = region_col),
              linetype = "solid", color = "beige")
}
```

```
## Warning: ‘aes_string()‘ was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with ‘aes()‘.
## i See also ‘vignette("ggplot2-in-packages")‘ for more information.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
# Add boundaries, beta_ols
plot_initial <- plot_initial + geom_line(aes(y = y_estimated),
                                         linetype = "solid", color = "coral") +
  geom_line(aes(y = width_upper), linetype = "solid", color = "forestgreen") +
  geom_line(aes(y = width_lower), linetype = "solid", color = "forestgreen") +
  labs(title = "Confidence Regions from Two Separate Methods") +
  theme_minimal()
print(plot_initial)
```



The coral/orange colored line is calculated from the estimates of beta from ordinary least squares. The two green lines denote the 95% confidence region band (from the first method), and the beige lines are sampled from the boundary of the 95% confidence set (from the second method). The region from the first method is a simultaneous confidence region, while the one from the second method is an elliptical confidence region.

The latter is less strict compared to the former, hence the confidence region from the first method was expected to be wider than that from the second method.