

STAT 30030 HW 5, Exercise 5:

Confidence Intervals

Robert Winter

Table of Contents

1	Introduction	1
2	Analysis	2
3	Results	4
	References	5

1 Introduction

In class, we talked about several ways to build confidence intervals for the parameter of the exponential distribution. The setting is as follows: suppose that T_1, \dots, T_n are independent and that each is distributed $\text{Exp}(\lambda)$. We denote with \bar{T}_n the sample mean based on these observations and we calculated the MLE:

$$\hat{\lambda} = \frac{1}{\bar{T}_n} = \frac{n}{\sum_{i=1}^n T_i}.$$

We will empirically explore three different confidence intervals:

1. Approximate the distribution on $\hat{\lambda}$ by its normal asymptotic distribution and calculate a 95% CI based on that (you need to calculate the Fisher information).
2. Use the pivot statistic we introduced in class, $\lambda \sum_{i=1}^n T_i$ and its exact distribution (Gamma) to calculate a 95% CI.
3. Use the pivot statistic above, but, instead of using the Gamma, estimate its distribution quantiles using the bootstrap, and use those quantiles to construct a 95% CI.

Choose $\lambda = 3$ and generate 10,000 samples of 20 random variables T_1, \dots, T_{20} with distribution $\text{Exp}(3)$. For each of the 10,000 samples, construct the three CIs. For each CI, calculate its coverage (the proportion of simulations where the true value is in the interval), and report the 3 numbers.

2 Analysis

```
library(dplyr)
library(ggplot2)
```

```
lambda = 3
```

```
# Create empty dataframe to store simulation results
coverages = data.frame(matrix(0, nrow = 10000, ncol = 3))
colnames(coverages) = c("coverage_normal", "coverage_gamma",
  ↪  "coverage_bootstrap")

#10,000 trials
for(i in c(1:10000)){
  # Draw 20 observations from the Exp(3) and compute MLE
  Ts = rexp(20, lambda)
  lambda_hat = 1 / mean(Ts)

  # Approach 1: Normal Asymptotic Distribution of MLE
  # From HW 4, lambda_hat ~ N(lambda, lambda^2 / n)
  # Fisher Information is 1 / lambda^2
  # Approximate CI is lambda_hat +/- z/sqrt(n * I(lambda))

  # Build CI
  ci_normal = c(lambda_hat - 1.96 * (lambda_hat/sqrt(20)),
                lambda_hat + 1.96 * (lambda_hat/sqrt(20)))

  # Check/store if lambda is in CI
  if(ci_normal[1] <= lambda & lambda <= ci_normal[2]){
    coverages$coverage_normal[i] = 1
  }
  else{
    coverages$coverage_normal[i] = 0
  }

  # Approach 2: Pivot Statistic and Gamma Distribution
```

```

#  $X_i \sim \text{Exp}(\lambda) = \text{Gamma}(1, \lambda)$ 
# So  $\lambda * \sum(X_i) \sim \text{Gamma}(n, 1)$ 
# CI is  $(A/\sum(X_i), B/\sum(X_i))$ , where A and B from Gamma quantile
A = qgamma(0.025, shape = 20, rate = 1)
B = qgamma(0.975, shape = 20, rate = 1)

# Build CI
ci_gamma = c(A / sum(Ts), B / sum(Ts))

# Check/store if lambda is in CI
if(ci_gamma[1] <= lambda & lambda <= ci_gamma[2]){
  coverages$coverage_gamma[i] = 1
}
else{
  coverages$coverage_gamma[i] = 0
}

# Approach 3: Pivot Statistic and Bootstrap
# We find that this approach "underperforms," with a coverage closer to
  ↪ 90%
# than 95%. See
  ↪ https://journals.sagepub.com/doi/pdf/10.1177/1094428105280059,
# page 14 for an explanation.

# Take 120 bootstrap samples
num_bootstrap_samples = 120
bootstrap_pivots = c()
for(k in c(1:num_bootstrap_samples)){
  # Draw 20 samples with replacement from Ts
  Ts_star = c()
  indices = sample(1:20, 20, replace = T)
  for(j in c(1:length(indices))){
    Ts_star[j] = Ts[indices[j]]
  }
  # Construct & store a new estimate of pivot based on our bootstrap
    ↪ sample
  bootstrap_pivots[k] = lambda_hat * sum(Ts_star)
}

# Build CI
bootstrap_pivots = bootstrap_pivots %>% sort()
L = bootstrap_pivots[0.025*num_bootstrap_samples + 1]
U = bootstrap_pivots[0.975*num_bootstrap_samples]
ci_bootstrap = c(L / sum(Ts), U / sum(Ts))

```

```

# Check/store if lambda is in CI
if(ci_bootstrap[1] <= lambda & lambda <= ci_bootstrap[2]){
  coverages$coverage_bootstrap[i] = 1
}
else{
  coverages$coverage_bootstrap[i] = 0
}
}

# Coverage Results
coverage_normal = sum(coverages$coverage_normal) / nrow(coverages)
coverage_gamma = sum(coverages$coverage_gamma) / nrow(coverages)
coverage_bootstrap = sum(coverages$coverage_bootstrap) / nrow(coverages)

```

3 Results

The coverage for our first class of confidence intervals (using the normal asymptotic distribution of $\hat{\lambda}$) was 0.9532, approximately equal to the 95% that we would expect.

The coverage for our second class of confidence intervals (using the pivot statistic $\lambda \sum_{i=1}^n T_i$ and its exact distribution $\text{Gamma}(n, 1)$) was 0.9512, also approximately equal to the 95% that we would expect.

The coverage for our third class of confidence intervals (using the pivot statistic $\lambda \sum_{i=1}^n T_i$ and its approximate distribution using the bootstrap) was 0.9003, which is closer to 90% than the expected 95%. However, there is precedent for this result. Indeed, (Wood 2005) writes:

The whole problem of confidence interval estimation is that the parent population is unknown. Dissimilarities between the two populations may mean that the accuracy of the resulting confidence intervals is disappointing. For example, Efron and Tibshirani (1993, p. 175) drew 300 samples of 10 from a standard normal distribution and worked out 95% bootstrap percentile intervals for each of them. They found that 10% of the intervals failed to include the population mean—twice the intended value of 5%. They explain this by pointing out that “the percentile interval has no knowledge of the underlying normal distribution and uses the empirical distribution in its place. In this case, it underestimates the tails of the distribution.”

Our case is similar: our confidence intervals use the empirical distribution of $\lambda \sum_{i=1}^n T_i$ in place of the true $\text{Gamma}(n, 1)$ distribution, causing twice as many confidence intervals as intended failing to include the true value of λ .

References

- Wood, Michael. 2005. "Bootstrapped Confidence Intervals as an Approach to Statistical Inference." *Organizational Research Methods* 8 (4): 454–70. <https://doi.org/10.1177/1094428105280059>.