Empirical Power For The Sign-Test,T-test and The Wilcoxon Signed Rank Test

Acquah Theophilus B.K.

December 25, 2023

Contents

Let $X_1, \ldots, X_n$ be a random sample from an unknown continuous distribution which is symmetric about θ with distribution function $G_0(x) = F(x-\theta)$. That is $P_0(X \leq x) = F(x-\theta)$. Now, we test

$$H_0 : \theta = 0 \quad \text{versus} \quad H_1 : \theta > 0. \tag{1}$$

Note that this model is indexed by two unknown parameters, θ and the underlying distribution $F(.)$. Here F is an unknown function which is symmetric about zero. That is $F(x) = F(-x), x \in \mathbb{R}$.

1. **sign-test**
   The sign test, with test statistic $B = [\text{number of } X_i's > 0]$, is appropriate for this problem. Under H_0, B is a binomial random variable with probability of success equal to 0.5. For any $\theta > 0$, B is a binomial random variable with probability of success equal to

   $$1 - G_0(0) = 1 - F(-\theta) = F(\theta). \tag{2}$$

   The last equality follows from the fact that F is a symmetric distribution. So if the rule rejects whenever $B \geq b_0$, the power function is calculated by

   $$p((\theta, F) = \sum_{b=b_0}^{n} \binom{n}{b} F(\theta)^b \left[1 - F(\theta)\right]^{n-b}.$$

2. **t-test**
   The one-sample Student t-test for this location problem. We reject H_0 if

   $$T^+ = \frac{\bar{X}}{S/\sqrt{n}}$$

   is greater than $t_{1-\alpha,n-1}$, the upper $100(1-\alpha)$th quantile of a t-distribution with $n-1$ degree of freedom. If the underlying distribution is normal with mean θ and variance σ^2, the power function of the test based on T^+ is

   $$p(T^+ > t(\alpha, n-1)|\theta, \sigma) = P\left(\frac{\bar{X}}{S/\sqrt{n}} > t(\alpha, n-1)|\theta, \sigma\right) \tag{3}$$

   $$= P\left(\frac{\frac{\bar{X}-\theta}{\sigma/\sqrt{n}} + \frac{\theta}{\sigma/\sqrt{n}}}{(s/\sqrt{n})} > t(\alpha, n-1)|\theta, \sigma\right) \tag{4}$$

$$= P\left(T^* > t(\alpha, n-1) | n-1, \frac{\theta}{\sigma/\sqrt{n}}\right), \tag{5}$$

where T^* has a non-central t-distribution with degree of freedom $n-1$,non-centrality parameter $\frac{\theta}{\sigma/\sqrt{n}} = \sqrt{n}\theta/\sigma$

3. **Wilcoxon Signed Rank Test**
The power function of the Wilcoxon signed rank test is more complex, and we use simulation to calculate the empirical power function. You can calculate empirical power for the sign-test and t-test, if you do not want to use the exact power function given above.Replace the distribution function F with the following symmetric distributions:

   (a) Uniform distribution with mean 0 and variance 1,

   (b) Normal distribution with mean 0 and variance 1, and

   (c) Double exponential distribution with mean 0 and variance 1.

   Compute the empirical power for the sign-test, t-test, and the Wilcoxon signed rank test based on generating random samples of sizes 10, 15, and 20 from the above distributions.

   Please refer to the attached PDF files on Canvas for more details on power function and empirical power function.

CODE IMPLEMENTATION

The following cumulative distribution functions (CDFs) are defined for distributions that are symmetric around zero and standardized to have a variance of 1:

1. **Uniform distribution:** The uniform distribution is defined over an interval $[a, b]$. To achieve a mean of 0 and a variance of 1, we set the bounds to $a = -\sqrt{3}$ and $b = \sqrt{3}$ since the variance of a uniform distribution over $[a, b]$ is $\frac{(b-a)^2}{12}$.
2. **Normal distribution:** The normal distribution is parameterized by its mean (μ) and standard deviation (σ). A standard normal distribution, which has a mean of 0 and a standard deviation of 1, naturally has a variance of 1.
3. **Double exponential (Laplace) distribution:** The variance of the Laplace distribution is given by $2 \cdot \text{scale}^2$. To obtain a variance of 1, the scale parameter is set to $\sqrt{0.5}$.

Define the cumulative distribution functions 'F' for the specified distributions

```r
# Define the cumulative distribution functions 'F' for the specified distributions

F_uniform <- function(x) punif(x, min = -sqrt(3), max = sqrt(3))
F_normal <- function(x) pnorm(x, mean = 0, sd = 1)
F_double_exponential <- function(x) plaplace(x, location = 0, scale = sqrt(0.5))
```

Define the power function for the sign test using a symmetric distribution

The `switch` function in R evaluates an expression and matches the result against a list of possible cases to execute corresponding code blocks. For example:

```
switch(expression,
case1 = code_for_case1,
case2 = code_for_case2,
...
caseN = code_for_caseN)
```

Here, `expression` is compared with `case1`, `case2`, ..., `caseN`. If there is a match, the code associated with that case is executed.

```r
# Define the power function for the sign test using a symmetric distribution
sign_test_power <- function(n, theta, F, alpha = 0.05){
  b0 <- qbinom(alpha, size = n, prob = 0.5, lower.tail = FALSE)
  sum(sapply(b0:n, function(b) choose(n, b) * F(theta)^b * (1 - F(theta))^(n - b)))
}

# Function to calculate empirical power using simulation
compute_empirical_power <- function(n, theta_over_sigma, dist_name,
                                    nsim = 5000, alpha = 0.05) {
  power_sign_test <- numeric(nsim)
  F <- switch(dist_name,
              uniform = F_uniform,
              normal = F_normal,
              double_exponential = F_double_exponential)

for (i in 1:nsim) {
  data <- switch(dist_name,
            uniform = runif(n, min = -sqrt(3), max = sqrt(3)),
            normal = rnorm(n, mean = 0, sd = 1),
            double_exponential = rlaplace(n, location = 0, scale = sqrt(0.5)))
    data <- data + theta_over_sigma
    power_sign_test[i] <- sign_test_power(n, theta_over_sigma, F, alpha)
  }
  mean(power_sign_test)
}
```

Function to calculate empirical power for t-test and Wilcoxon test

```r
# Function to calculate empirical power for t-test and Wilcoxon test
calculate_test_power <- function(n, theta, dist_func, nsim, alpha, test_type) {
  replicate(nsim, {
    data <- dist_func(n) + theta
    if (test_type == "t") {
      p_value <- t.test(data, mu = 0, alternative = 'greater')$p.value
    } else if (test_type == "wilcox") {
      p_value <- wilcox.test(data, mu = 0, alternative = 'greater')$p.value
    }
    p_value <= alpha
  }) %>% mean() * 1000
}
```

Set Parameters, Calculate Power,Run Simulations

```r
# Set parameters
nsim <- 5000
n_values <- c(10, 15, 20)
theta_values <- seq(0, 0.8, by = 0.2)
alpha <- 0.05
set.seed(123)
```

```r
# Calculate power
results <- expand.grid(n = n_values, theta_over_sigma = theta_values,
                       dist = c("uniform", "normal", "double_exponential"))
results$T_plus <- NA
results$W_plus <- NA
results$B <- NA

for (i in 1:nrow(results)) {
  n <- results$n[i]
  theta <- results$theta_over_sigma[i]
  dist_name <- results$dist[i]

  # Calculate power for t-test
  results$T_plus[i] <- calculate_test_power(n, theta,
                 switch(dist_name,uniform = function(n) runif(n, min = -sqrt(3),
                 max = sqrt(3)),normal = rnorm,
                 double_exponential = function(n) rlaplace(n, location = 0,
                 scale = sqrt(0.5))), nsim, alpha, "t")

  # Calculate power for Wilcoxon test
  results$W_plus[i] <- calculate_test_power(n, theta,
                 switch(dist_name,uniform = function(n) runif(n, min = -sqrt(3),
                 max = sqrt(3)),normal = rnorm,
                 double_exponential = function(n) rlaplace(n, location = 0,
                 scale = sqrt(0.5))), nsim, alpha, "wilcox")

  # Calculate power for sign test
  results$B[i] <- 1000 * compute_empirical_power(n, theta, dist_name, nsim, alpha)
}
```

Create Table For Uniform Distribution

```r
# Reshape the data for Uniform distribution
uniform_table <- results %>%
  filter(dist == "uniform") %>%
  select(-dist) %>%
  gather(key = "Test", value = "Power",B, T_plus, W_plus) %>%
   mutate(Test = factor(Test, levels = c("T_plus", "W_plus", "B"))) %>%
  spread(key = theta_over_sigma, value = Power) %>%
  mutate(across(where(is.numeric), round, digits = 0))


# Print the table for Uniform distribution
kable(uniform_table, caption = "Empirical Power Times 1000 for Uniform Distribution")
```

Table 1: Empirical Power Times 1000 for Uniform Distribution

n	Test	0	0.2	0.4	0.6	0.8
10	T_plus	53	137	285	509	747
10	W_plus	42	113	240	436	651
10	B	55	108	194	315	469
15	T_plus	50	172	414	703	908

n	Test	0	0.2	0.4	0.6	0.8
15	W_plus	46	153	378	630	840
15	B	59	133	255	425	622
20	T_plus	50	209	524	824	974
20	W_plus	45	199	478	767	938
20	B	58	145	297	504	722

Create Table For Normal Distribution

```
# Repeat similar steps for Normal and Double Exponential distributions
normal_table <- results %>%
  filter(dist == "normal") %>%
  select(-dist) %>%
  gather(key = "Test", value = "Power", T_plus, W_plus, B) %>%
  mutate(Test = factor(Test, levels = c("T_plus", "W_plus", "B"))) %>%
  spread(key = theta_over_sigma, value = Power) %>%
  mutate(across(where(is.numeric), round, digits = 0))


kable(normal_table, caption = "Empirical Power Times 1000 for Normal Distribution")
```

Table 2: Empirical Power Times 1000 for Normal Distribution

n	Test	0	0.2	0.4	0.6	0.8
10	T_plus	50	151	332	551	756
10	W_plus	42	120	276	499	700
10	B	55	136	273	454	642
15	T_plus	46	183	440	714	910
15	W_plus	43	168	408	688	882
15	B	59	172	369	604	804
20	T_plus	59	218	518	821	968
20	W_plus	48	195	514	801	954
20	B	58	194	437	704	889

Create Table For Double Exponential Distribution

```
double_exp_table <- results %>%
  filter(dist == "double_exponential") %>%
  select(-dist) %>%
  gather(key = "Test", value = "Power", T_plus, W_plus, B) %>%
  mutate(Test = factor(Test, levels = c("T_plus", "W_plus", "B"))) %>%
  spread(key = theta_over_sigma, value = Power) %>%
  mutate(across(where(is.numeric), round, digits = 0))

kable(double_exp_table, caption = "Empirical Power Times 1000 for Double Exponential Distribution")
```

Table 3: Empirical Power Times 1000 for Double Exponential Distribution

n	Test	0	0.2	0.4	0.6	0.8
10	T_plus	47	160	364	601	779
10	W_plus	41	161	357	597	765
10	B	55	208	427	635	790
15	T_plus	51	209	470	742	895
15	W_plus	53	226	519	769	922
15	B	59	275	571	798	920
20	T_plus	51	234	562	826	954
20	W_plus	45	271	641	886	974
20	B	58	322	668	884	968