Examples

The following document demonstrates the use of the functions in `implementation.R` for a one-sample t-test. That is, we have data $X_1, \dots, X_n \sim \mathcal{N}(\mu, \sigma^2)$ and we will test the hypotheses

$$H_0 : \mu = 0 \qquad \text{and} \qquad H_A : \mu \neq 0.$$

Running the test

We begin by demonstrating how the test can be run for a given dataset. The following code generates a simple dataset with $n = 30$ and an effect size of $\mu/\sigma = 1$.

```
set.seed(42)

n <- 30
data <- data.frame(Y = rnorm(n, mean = 1, sd = 1))
```

The following code creates an appropriate function `pub_test` and runs the test of tests for $\varepsilon = 1$ and input parameters $m = 3, \alpha_0 = 0.2$.

```
set.seed(777)

pub_test <- function(data){
  t.test(x = data)$p.value
}

test.of.tests(data, pub_test, epsilon = 1, m = 3, alpha0 = 0.2)
```

```
## $Z
## [1] 3.716806
##
## $p.value
## [1] 0.02803017
```

Computing Theoretical Power

We now describe how to compute the theoretical power of the test of tests. The following creates an appropriate function `pub_power`.

```
pub_power <- function(effect, n, alpha){
  power.t.test(n = n, delta = effect, sd = 1, sig.level = alpha,
               type = "one.sample", alternative = "two.sided",
               strict = T)$power
}
```

The power of the test for an effect size of $\mu/\sigma = 1$ and input parameters $m = 3, \alpha_0 = 0.2$ is thus

```
power.test.of.tests(effect = 1, pub_power, epsilon = 1, n = 30, m = 3,
                    alpha0 = 0.2)
```

```
## [1] 0.3818626
```

We can confirm this result via simulation using the `test.of.tests` function described above.

```
set.seed(1)
nsims <- 1000

samps <- rep(NA, nsims)
for(i in 1:nsims){
  data <- data.frame(Y = rnorm(n, mean = 1, sd = 1))
  samps[i] <- test.of.tests(data, pub_test, epsilon = 1, m = 3, alpha0 = 0.2)$p.value
}
mean(samps <= 0.05)
```

```
## [1] 0.382
```

Finding m, α_0

We now describe how to use the remaining functions to select the input parameters. When the effect size is not known, the function `practical.m.alpha0` should be used. Here we demonstrate this function searching over a grid of effect sizes for the smallest effect that achieves target power $\rho = 0.8$.

```
pars <- practical.m.alpha0(pub_power, epsilon = 1, n = 30, rho = 0.8,
                           effect_grid = seq(0.1, 2, 0.1))
pars
```

```
## $power
## [1] 0.8158562
##
## $alpha0
## [1] 0.2308264
##
## $m
## [1] 7
##
## $effect
## [1] 1.2
```

Using the corresponding input parameters in `power.test.of.tests` gives substantially higher power in the $\mu/\sigma = 1$ case examined above.

```
power.test.of.tests(effect = 1, pub_power, epsilon = 1, n = 30, m = pars$m,
                    alpha0 = pars$alpha0)
```

```
## [1] 0.6607279
```

For comparison, if it were known a priori that the effect size was approximately $\mu/\sigma = 1$, then we could use the function `optimal.m.alpha0` to find the maximum achievable power.

```
optimal.m.alpha0(effect = 1, pub_power, epsilon = 1, n = 30)
```

```
## $power
## [1] 0.6610846
##
## $alpha0
## [1] 0.175706
##
## $m
## [1] 5
```

We see that this power is only marginally larger than the power from when the effect size was unknown.