Estimating Equations

Example: Consider the $\boldsymbol{Z} = (Z_1, \ldots, Z_5)^\top$ with cdf

$$F(\boldsymbol{z}; \alpha) = \exp\left\{ -\left(z_1^{-\frac{1}{\alpha}} + z_2^{-\frac{1}{\alpha}} + z_3^{-\frac{1}{\alpha}} + z_4^{-\frac{1}{\alpha}} + z_5^{-\frac{1}{\alpha}} \right)^{\alpha} \right\}, \quad \boldsymbol{z} \geq \mathbf{0}, \alpha \in (0, 1].$$

Comments:

1. F is max-stable.

2. $Z_1, \ldots, Z_5$ are exchangeable.

Let’s consider the likelihood.

How about if we were to just use pairs of points to estimate α?

If we just used $(z_{1i}, z_{2i}), i = 1, \ldots, n$ would the likelihood based on the bivariate density be a good estimator for α?

Let's try it.

```
library(evd)
# simulate data with alpha = 0.5
alpha <- 0.5
z <- rmvevd(500, dep = alpha, d = 5, mar = c(1, 1, 1))

## bivariate density
d_bivar <- function(z, alpha){
    #here "z" is a single observation (ordered pair)
    inside <- z[1]^(-1/alpha) + z[2]^(-1/alpha)
    one <- exp(-inside^alpha)
    two <- (z[1]*z[2])^(-1 / alpha - 1)
    three <- (1 / alpha - 1)*inside^(alpha - 2)
    four <- inside^(2 * alpha - 2)
    one*two*(three + four)
}

d_bivar(c(4, 5), alpha = alpha)
```

```
## [1] 0.003650963
```

```
dmvevd(c(4,5), dep = alpha, d = 2, mar = c(1,1,1))
```

```
## [1] 0.003650963
```

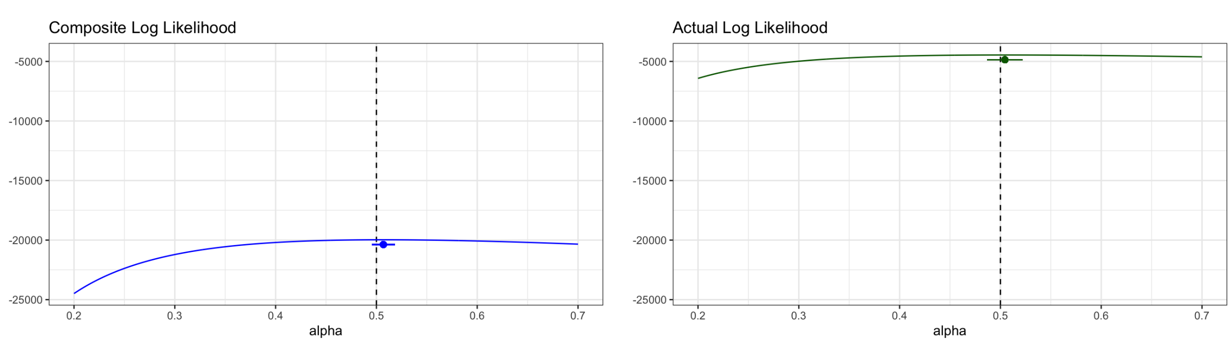
```
## estimate alpha
log_pair_lhood <- function(alpha, z) {
    #here "z" is bivariate matrix of observations
    inside <- z[, 1]^(-1 / alpha) + z[, 2]^(-1 / alpha)
    log_one <- -inside^alpha
    log_two <- (-1 / alpha - 1) * (log(z[, 1]) + log(z[, 2]))
    three <- (1 / alpha - 1) * inside^(alpha - 2)
    four <- inside^(2 * alpha - 2)
    contrib <- log_one + log_two + log(three + four)
    return(sum(contrib))
}

all_pairs_lhood <- function(alpha, z) {
```

```
    expand.grid(dim1 = seq_len(ncol(z)),
                dim2 = seq_len(ncol(z))) |>
      filter(dim1 < dim2) |>       rowwise() |>
      mutate(log_pair_lhood = log_pair_lhood(alpha, cbind(z[, dim1],
        z[, dim2]))) |>
    ungroup() |>      summarise(res = sum(log_pair_lhood)) |>
      pull(res)}
alpha_mple <- optim(.2, lower = .01, upper = .99, all_pairs_lhood, z
        = z, method = "Brent", hessian = TRUE, control =
        list(fnscale = -1))
(ci_mple <- alpha_mple$par + c(-1.96, 1.96)*sqrt(-1 /
        alpha_mple$hessian[1, 1]))
```

```
## [1] 0.4954979 0.5182678
```

```
## checking coverage
#checking coverage
B <- 200
coverage <- numeric(B)
for(k in seq_len(B)) {
    z_k <- rmvevd(500, dep = .5, d = 5, mar = c(1, 1, 1))
    alpha_mple_k <- optim(.2, lower = .01, upper = .99,
        all_pairs_lhood, z = z_k, method = "Brent", hessian = TRUE,
        control = list(fnscale = -1))
    ci <- alpha_mple_k$par + c(-1.96, 1.96)*sqrt(-1 /
        alpha_mple_k$hessian[1, 1])
    coverage[k] <- as.numeric(ci[1] < alpha & ci[2] > alpha)
}
mean(coverage)
```

```
## [1] 0.745
```



So, it looks like the point estimate from the pairwise likelihood is ok, but we need to be able to get an appropriate measure of uncertainty.

The proper adjustment is

1 Introduction

M-estimators are solutions of the vector equation

$$\sum_{i=1}^{n} \boldsymbol{\psi}(\mathbf{Y}_i, \boldsymbol{\theta}) = \mathbf{0}.$$

In the likelihood setting, what is $\boldsymbol{\psi}$?

Example: Let $Y_1, \ldots, Y_n$ be independent, univariate random variables. Is $\theta = \overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$ an M-estimator?

Example: Consider the mean deviation from the sample mean,

$$\hat{\theta}_1 = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \overline{Y}|.$$

Is this an M-estimator?