`refitME`: Tutorial for fitting MCEM models when covariates are subject to measurement error/error-in-variables

Jakub Stoklosa, Hwang W-H., & David Warton

16 September, 2020

This tutorial documents fitting an MCEM algorithm via the `refitME` R-package. For more specific details see: "A general algorithm for error-in-variables modelling using Monte Carlo expectation maximization." Also, see `?refitME` for further details on the fitting function, input arguments and output.

Example 1: A simple GLM example taken from Carroll *et al.* (2006).

The Framingham heart study data set. We also fit SIMEX models and compare them with MCEM. Computational times for both models are also reported.

Load data and R-packages.

```
suppressMessages(library(refitME))
suppressMessages(library(simex))

epsilon <- 0.00001  # A set convergence threshold.
B <- 100  # The number of Monte Carlo replication values/SIMEX simulations.
family <- "binomial"
data(Framinghamdata)
```

Setup all variables (the construction below follows the Carroll et al. (2006) monograph).

```
W <- as.matrix(Framinghamdata$w1) # Matrix of error-contaminated covariate.
sigma.sq.u <- 0.01259/2 # ME variance, obtained from Carroll et al. (2006) monograph.
```

Fit the naive model.

The first stored variable `w1` is the error contaminated variable used in the analysis.

```
mod_naiv1 <- glm(Y ~ w1 + z1 + z2 + z3, x = TRUE, family = binomial, data = Framinghamdata)
```

Fit the SIMEX model.

```
start <- Sys.time()
mod_simex1 <- simex(mod_naiv1, SIMEXvariable = c("w1"),
                    measurement.error = cbind(sqrt(sigma.sq.u)), B = B) # SIMEX.
end <- Sys.time()
t1 <- difftime(end,start, units = "secs")
comp.time <- c(t1)
```

Fit the MCEM model.

```
start <- Sys.time()
est <- refitME(mod_naiv1, sigma.sq.u, W, B)
```

```
## [1] "One specified error-contaminated covariate."
## [1] "convergence :-)"
## [1] 5
```

```
end <- Sys.time()
t2 <- difftime(end, start, units = "secs")
comp.time <- c(comp.time, t2)
```

Report and compare times and model estimates.

```
est.beta <- rbind(coef(mod_naiv1), coef(mod_simex1), est$beta)
est.beta.se <- rbind(sqrt(diag(vcov(mod_naiv1))),
                     sqrt(diag(mod_simex1$variance.jackknife)), est$beta.se2)
round(est.beta, digits = 3)
```

```
##      (Intercept)    w1    z1    z2    z3
## [1,]     -14.951 1.707 0.008 0.055 0.592
## [2,]     -15.720 1.900 0.008 0.054 0.599
## [3,]     -16.108 1.966 0.008 0.056 0.594
```

```
round(est.beta.se, digits = 3)  # Standard error estimates.
```

```
##      (Intercept)    w1    z1    z2    z3
## [1,]       1.900 0.418 0.002 0.012 0.250
## [2,]       2.101 0.473 0.002 0.012 0.251
## [3,]       2.188 0.488 0.002 0.012 0.250
```

```
comp.time  # SIMEX and MCEM.
```

```
## Time differences in secs
## [1] 9.700411 5.044528
```

Example 2: A GAM example taken from Ganguli *et al.* (2005).

The Milan mortality air pollution data set. Here, we fit GAM models via the `mgcv` package where one covariate is error-contaminated.

Load data and R-packages.

```
suppressMessages(library(refitME))
suppressMessages(library(SemiPar))

epsilon <- 0.00001  # A set convergence threshold.
B <- 5  # The number of Monte Carlo replication values.

family <- "poisson"

data(milan.mort)
dat.air <- milan.mort
```

Setup all variables.

```
Y <- dat.air[, 6]  # Mortality counts.
n <- length(Y)

z1 <- (dat.air[, 1])
z2 <- (dat.air[, 4])
z3 <- (dat.air[, 5])
w1 <- log(dat.air[, 9])
W <- as.matrix(w1)
dat <- data.frame(cbind(Y, z1, z2, z3, w1))

sigma.sq.u <- 0.0915 # This gives a reliability ratio of 0.7.
rel.rat <- round((1 - sigma.sq.u/var(dat$w1))*100, digits = 0)
```

Fit the naive model.

```
mod_naiv1 <- gam(Y ~ s(w1) + s(z1, k = 25) + s(z2) + s(z3), family = "poisson", data = dat)
```

Fit the MCEM model.

```
est <- refitME(mod_naiv1, sigma.sq.u, W, B)
```

```
## [1] "One specified error-contaminated covariate."
## [1] "convergence :-)"
## [1] 10
```

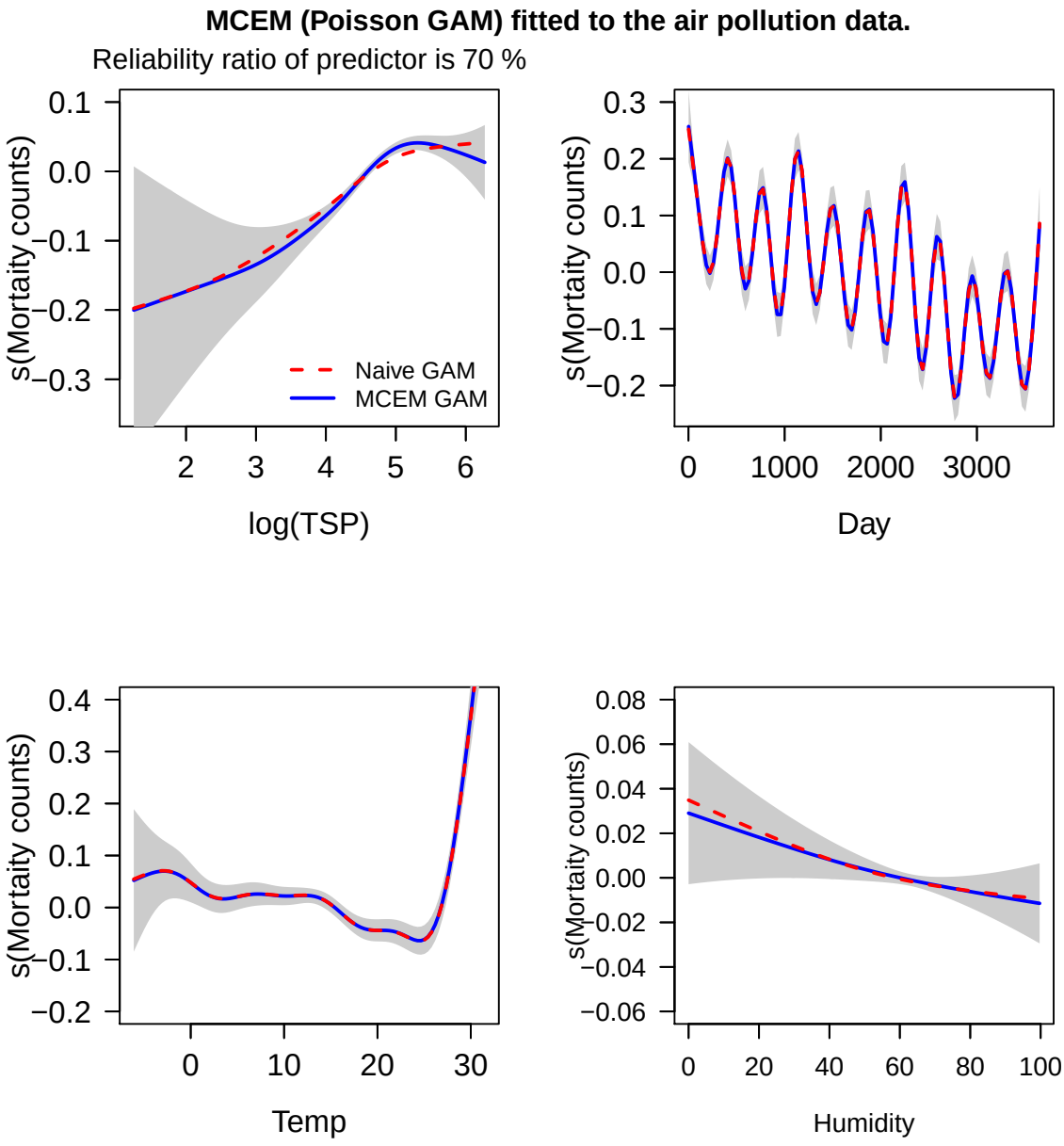

Figure 1: Plots of smooths against covariate. TSP (top left is the error contaminated variable).