Taylor approximation - based methods for bias correction

Bicko, Jonathan & Ben

2021 Aug 11 (Mon)

Introduction

We consider two approaches:

- Second order Taylor approximation
- Delta method

and then compare our approximations to *closed form integral of Gaussian over logistic curve.*

Second order Taylor approximation

We consider Taylor series approximation approach for bias correction. Suppose $f(.)$ represents a back-transformation function, μ the mean and σ^2_μ on the linear predictor scale (**univariate case** // **what happens in multivariate case?**) predictor. Then using Taylor expansion around μ, the approximation on the original scale is given by

$$f(\mu) + \frac{1}{2}\sigma^2_\mu f''(\mu)$$

To start with, we consider logistic function

$$g(\mu) = \frac{1}{1 + \exp(-\mu)},$$

with the first and second derivatives give by

$$g'(\mu) = g(\mu)(1 - g(\mu))$$

and

$$g''(\mu) = g(\mu)g(-\mu)(1 - 2g(\mu))$$

respectively.

```
taylorapprox <- function(fun, var="x", mu, sigma, ...) {
    ff <- as.expression(substitute(fun))
    x <- mu
    untrans <- eval(ff, list(x))
    der <- D(ff, var)       # 1s derivative
    der2 <- D(der, var) # 2nd derivative
    der2 <- eval(der2, list(x))
    corrected <- untrans + der2*sigma^2/2
    return(corrected)
}
```

Delta method

Consider a first order Taylor approximation of $f(.)$ about the mean μ, evaluated at the random variable x_i defined as:

$$f(x_i) \approx f(\mu) + f'(x_i)(x_i - \mu)$$

with a variance of

$$\text{Var}\left(f(x_i)\right) = f'(\mu)\text{Var}(x_i)f'(\mu)$$

```
deltaapprox <- function(fun, var="x", z, ...) {
    ff <- as.expression(substitute(fun))
    der <- D(ff, var)
    x <- z
    ymu <- eval(ff, list(x))
    x <- mean(z)
    der <- eval(der, list(x))
    y <- ymu + der * (z - x)
    yvar <- der * var(z) * der      # Check with BB
    out <- list(fx = y, var = yvar)
    return(out)
}
```

Let us consider a simple univariate case

```
b0 <- 6
b1 <- 5
N <- 100
x <- scale(rnorm(N))
eta <- b0 + b1*x

taylor <- taylorapprox(fun=1/(1+exp(-x)), mu=mean(eta), sigma=sd(eta))
delta <- deltaapprox(fun=1/(1+exp(-x)), z=eta)
normal_approx <- logitnorm::momentsLogitnorm(mu= mean(eta), sigma = sd(eta))
out <- c(NULL
    , truth = mean(plogis(eta))
    , naive = plogis(mean(eta))
    , taylor = taylor
    , delta = mean(delta$fx)
    , normal = normal_approx[[1]]
)
print(out)
```

```
##     truth     naive    taylor     delta    normal
## 0.8729394 0.9975274 0.9668485 0.8729394 0.8706198
```

```
delta$var
```

```
##                [,1]
## [1,] 0.0001520917
```

```
normal_approx[[2]]
```

```
## [1] 0.07301504
```

Simple logistic model example

```
N <- 1e4
beta0 <- 0.7
betaA <- 0.2
betaW <- 0.5

age_max <- 1
age_min <- 0.2
age <- runif(N, age_min, age_max)
# age <- rnorm(N, age_max, age_max)

wealthindex <- rnorm(N, 0, 1)

eta <- beta0 + betaA * age + betaW * wealthindex
sim_df <- (data.frame(age=age, wealthindex=wealthindex, eta=eta)
    %>% mutate(status = rbinom(N, 1, plogis(eta)))
    %>% select(-eta)
)
true_prop <- mean(sim_df$status)
print(true_prop)
```

```
## [1] 0.6918
```

```
simple_mod <- glm(status ~ age + wealthindex, data = sim_df, family="binomial")
```

Applying `taylor`, `delta` and `normal approximation` method

```
genpred <- function(mod, focal, non.focal, level=0.95, steps=100, pop=FALSE, bias.adjust=c("none", "del

    bias.adjust <- match.arg(bias.adjust)
    mf <- model.frame(mod)
    mm <- (mf
        %>% select_at(c(focal, non.focal))
    )
    quant <- seq(0, 1, length.out=steps)
    mm1 <- sapply(focal, function(x)as.vector(quantile(mm[,x], quant)), simplify = FALSE)
    if (pop) {
        mm1[[non.focal]] <- mm[[non.focal]]
        mm <- do.call("expand.grid", mm1)
        mm <- model.matrix(formula(mod)[c(1,3)], mm)
    } else {
        mm2 <- sapply(non.focal, function(x)mean(mm[,x]), simplify = FALSE)
        mm <- do.call("data.frame", c(mm1, mm2))
        mm <- model.matrix(formula(mod)[c(1,3)], mm)
    }
    linpred <- as.vector(mm %*% coef(mod))
    out <- (mm
        %>% data.frame()
        %>% select_at(focal)
        %>% mutate(lp = linpred, model=modelname)
    )

    if (bias.adjust=="delta") {
        out <- (out
            %>% mutate(fit=deltaapprox(fun=1/(1+exp(-x)), z=lp)$fx)
        )
```

```
    } else if (bias.adjust=="taylor") {
        vc <- vcov(mod)
        pse_var <- sqrt(rowSums(mm * t(tcrossprod(data.matrix(vc), mm))))
        lp <- out$lp
        fit <- unlist(lapply(1:length(lp), function(i){
            taylorapprox(fun=1/(1+exp(-x)), mu=lp[[i]], sigma=pse_var[[i]])
        }))
        out <- (out
            %>% select(-lp)
            %>% mutate(fit = fit)
        )
    }   else if (bias.adjust=="normal") {
        vc <- vcov(mod)
        pse_var <- sqrt(rowSums(mm * t(tcrossprod(data.matrix(vc), mm))))
        lp <- out$lp
        fit <- unlist(lapply(1:length(lp), function(i){
            logitnorm::momentsLogitnorm(mu = lp[[i]], sigma = pse_var[[i]])[[1]]
        }))
        out <- (out
            %>% select(-lp)
            %>% mutate(fit = fit)
        )
    } else if (bias.adjust=="none") {
        out <- (out
            %>% mutate(fit = plogis(lp))
        )
    }
    if (pop) {
        out <- (out
            %>% group_by_at(focal)
            %>% summarise_at("fit", mean)
            %>% mutate(model=modelname)
        )
    }
    return(out)
}

## Population averaging
pop_none <- genpred(simple_mod, "age", "wealthindex", pop=TRUE, bias.adjust="none", modelname="pop-none
pop_delta <- genpred(simple_mod, "age", "wealthindex", pop=TRUE, bias.adjust="delta", modelname="pop-del

## Averaged // centered??
centered_none <- genpred(simple_mod, "age", "wealthindex", pop=FALSE, bias.adjust="none", modelname="cen
centered_delta <- genpred(simple_mod, "age", "wealthindex", pop=FALSE, bias.adjust="delta", modelname="c
centered_taylor <- genpred(simple_mod, "age", "wealthindex", pop=FALSE, bias.adjust="taylor", modelname=
```

```
genpred2 <- function(mod, focal, level=0.95, steps=100, modelname="none", ...) {

    mm <- model.matrix(mod)
    quant <- seq(0, 1, length.out=steps)
    xvals <- as.vector(quantile(mm[, focal], quant))
    vnames <- vareffects:::get_vnames(mod)$termnames
    check <- !vnames %in% focal
    non.focal.terms <- vnames[check]
```

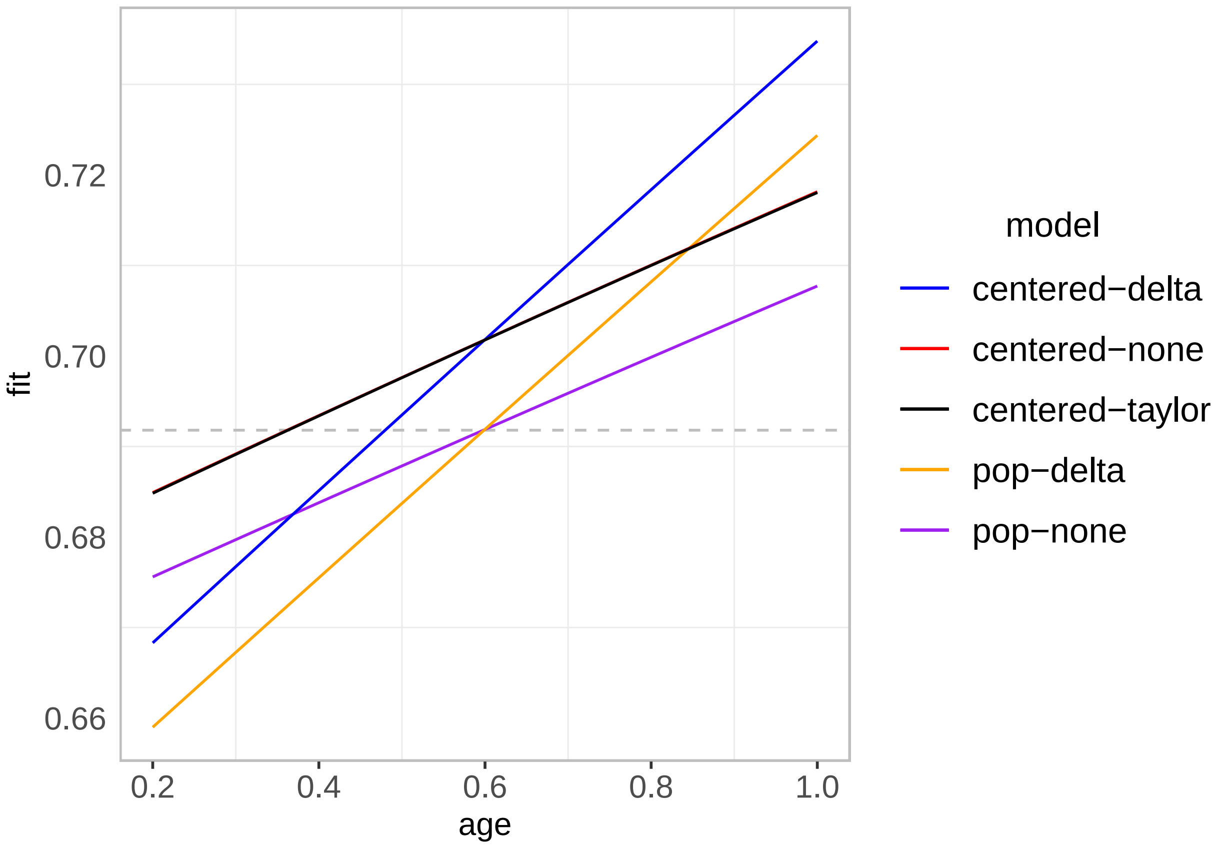
```
    betas <- coef(mod)
    non.focal.coefs <- betas[check]
    focal.coefs <- betas[!check]
    lp.non.focal <- as.vector(as.matrix(mm[, non.focal.terms, drop=FALSE]) %*% non.focal.coefs)
    lp.all <- unlist(lapply(xvals, function(x){
        mu <- mean(plogis(as.vector(c(x %*% focal.coefs) + lp.non.focal)))
        return(mu)
    }))

    out <- list(xx = xvals, xx2 = lp.all)
    return(out)
}

x1 <- genpred2(simple_mod, "wealthindex")
mean(x1$xx2)
```

```
## [1] 0.6903991
```

```
p1 <- (ggplot(pop_none, aes(x=age, y=fit, colour=model))
    + geom_hline(yintercept=true_prop, lty=2, colour="grey")
    + geom_line()
    + geom_line(data=pop_delta, aes(x=age, y=fit, colour=model))
    + geom_line(data=centered_none, aes(x=age, y=fit, colour=model))
    + geom_line(data=centered_delta, aes(x=age, y=fit, colour=model))
    + geom_line(data=centered_taylor, aes(x=age, y=fit, colour=model))
    + scale_colour_manual(values=c("blue", "red", "black", "orange", "purple"))
    + theme(legend.position="right")
)
print(p1)
```


0.72
0.70
0.68
0.66
fit
0.2
0.4
0.6
0.8
1.0
age
model
centered−delta
centered−none
centered−taylor
pop−delta
pop−none