Bias correction in GLMs

Bicko, Jonathan & Ben

2021 Jun 21 (Mon)

Introduction

We intend to investigate our prediction based on known truth and any bias potentially introduced by non-linear averaging, conditioning or random effect. We'll start with a simple case of a only fixed effect model and then consider a mixed effect model.

Simulation

We perform a simple simulation for a fixed effect model

$$
\begin{aligned}
\mathrm{logit}(\mathrm{status} = 1) &= \eta \\
\eta &= \beta_0 + \beta_{\mathrm{A}}\mathrm{Age} + \beta_{\mathrm{W}}\mathrm{Wealthindex} \\
\mathrm{Age} &\sim \mathrm{Uniform}(0.2, 1) \\
\mathrm{Wealthindex} &\sim \mathrm{Normal}(0, 1) \\
\beta_0 &= 0.7 \\
\beta_{\mathrm{A}} &= 0.3 \\
\beta_{\mathrm{W}} &= 0.6
\end{aligned}
$$

```
N <- 10000
beta0 <- 0.7
betaA <- 0.2
betaW <- 0.5

age_max <- 1
age_min <- 0.2
age <- runif(N, age_min, 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] 0.6916
```

```
head(sim_df)
```

```
##         age wealthindex status
## 1 0.8452918   1.1198420      1
## 2 0.2563395  -0.6219684      0
## 3 0.4192913  -1.5949657      1
## 4 0.7882493  -1.2565989      1
## 5 0.3893671   1.7148530      1
## 6 0.8806260  -0.1938844      1
```

Simple logistic model

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

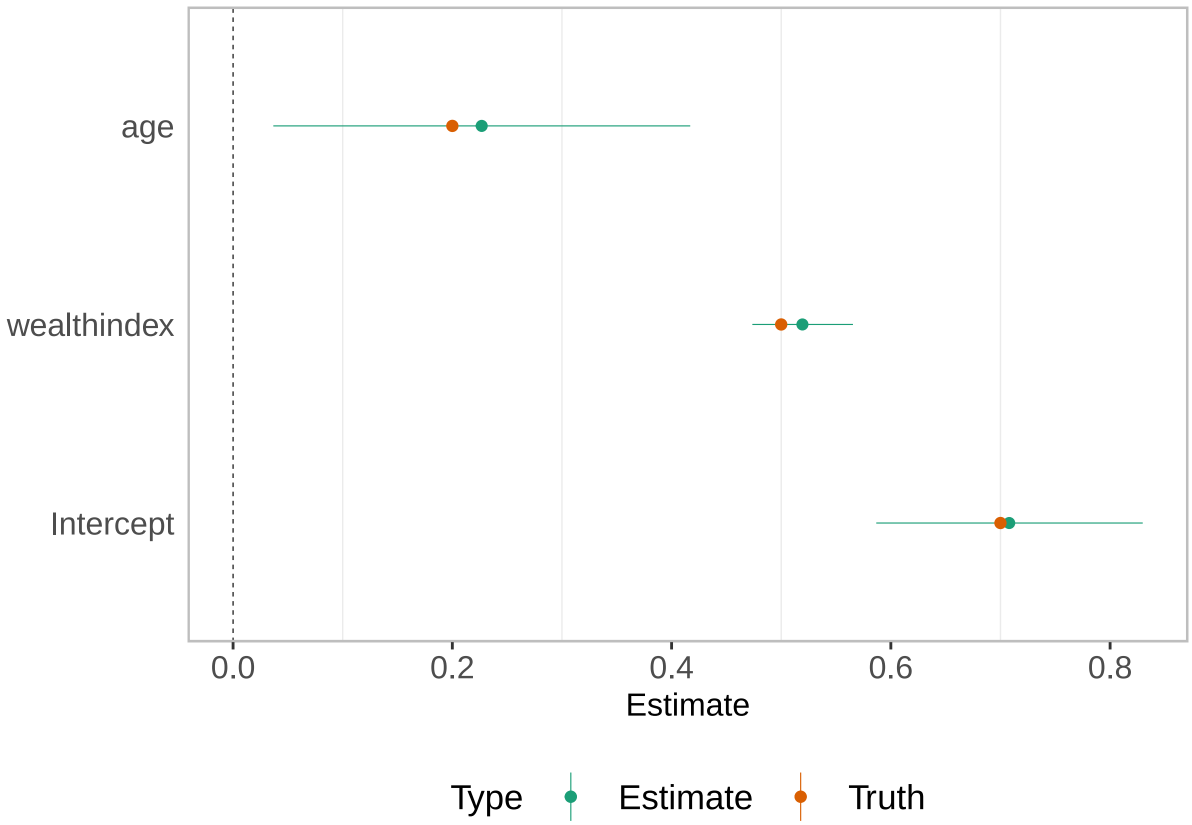
Coefficient plots

```
## True beta
true_beta_df <- data.frame(term=c("Intercept", "age", "wealthindex")
    , estimate=c(beta0, betaA, betaW)
)

## Tidy coef estimates
coef_df <- (broom::tidy(simple_mod, conf.int=TRUE)
#   %>% dotwhisker::by_2sd(sim_df)
    %>% mutate(term = gsub("\\(|\\)", "", term))
)
print(coef_df)
```

```
## # A tibble: 3 x 7
##   term        estimate std.error statistic   p.value conf.low conf.high
##   <chr>          <dbl>     <dbl>     <dbl>     <dbl>    <dbl>     <dbl>
## 1 Intercept      0.708    0.0620     11.4  3.46e- 30   0.587      0.830
## 2 age            0.227    0.0970      2.34 1.94e-  2   0.0367     0.417
## 3 wealthindex    0.519    0.0234     22.2  5.09e-109   0.474      0.565
```

```
simple_coef_plot <- (plotEsize(coef_df)
    + geom_point(data=true_beta_df, aes(x=term, y=estimate, colour="Truth"))
    + labs(colour="Type")
    + scale_colour_brewer(palette="Dark2")
)
print(simple_coef_plot)
```



Variable effect plots – with and without bias adjustment

```
# Age
## varpred way
simple_vareff_age <- varpred(simple_mod, "age", isolate=FALSE, modelname="varpred")

# Wealth index
## Not bias adjusted
simple_vareff_wealthindex <- varpred(simple_mod, "wealthindex", isolate=TRUE, modelname="varpred")

## Bias adjusted
simple_vareff_wealthindex_adjust <- varpred(simple_mod, "wealthindex", isolate=TRUE, pop.ave=TRUE, mode

vareff_wealthindex <- simple_vareff_wealthindex
vareff_wealthindex$preds <- do.call("rbind", list(vareff_wealthindex$preds, simple_vareff_wealthindex_a
wealthindex_plot <- (plot(vareff_wealthindex)
    + labs(y="", colour="Model")
    + geom_hline(yintercept=true_prop, lty=2, colour="grey")
    + theme(legend.position="bottom")
)
```

Effect sizes on logit scale:

```
coef_df_logit <- (coef_df
    %>% select(term, estimate, conf.low, conf.high)
    %>% group_by(term)
    %>% summarise_all(plogis)
)
print(coef_df_logit)
```

```
## # A tibble: 3 x 4
##   term        estimate conf.low conf.high
##   <chr>          <dbl>    <dbl>     <dbl>
## 1 age            0.556    0.509     0.603
## 2 Intercept      0.670    0.643     0.696
## 3 wealthindex    0.627    0.616     0.638
```

Population averaging

- Averages of the entire population of the non-focal predictor

```
popavefun <- function(mod, focal, non.focal, level=0.95, modelname="Pop. ave", ...) {
    mf <- model.matrix(mod)
    mm <- (mf
        %>% data.frame()
        %>% mutate_at(non.focal, mean)
        %>% as.matrix()
    )
    vc <- vcov(mod)
    linpred <- as.vector(mm %*% coef(mod))
    pse_var <- sqrt(rowSums(mm * t(tcrossprod(data.matrix(vc), mm))))
    z.val <- qnorm(1 - (1 - level)/2)
    pred_df <- (mf
        %>% data.frame()
        %>% select_at(focal)
        %>% mutate(fit = linpred
            , lwr = plogis(fit - z.val*pse_var)
            , upr = plogis(fit + z.val*pse_var)
            , fit = plogis(fit)
            , model = modelname
            , se = NA
        )
    )
    return(pred_df)
}

# simple_vareff_age_pop <- varpred(simple_mod, "age", isolate=TRUE, modelname="Pop. ave")
simple_vareff_age_pop <- popavefun(simple_mod, "age", "wealthindex", modelname = "Pop. ave")

vareff_age <- simple_vareff_age
vareff_age$preds <- do.call("rbind", list(vareff_age$preds, simple_vareff_age_pop))
age_plot <- (plot(vareff_age)
    + labs(y="Prob. of improved \n service", colour="Model")
    + geom_hline(yintercept=true_prop, lty=2, colour="grey")
    + theme(legend.position="bottom")
)

ggarrange(age_plot, wealthindex_plot, common.legend=TRUE)
```

The observed population average is 0.6916 while the estimated population averages are:

- age: 0.7018488
- wealthindex: 0.6916761

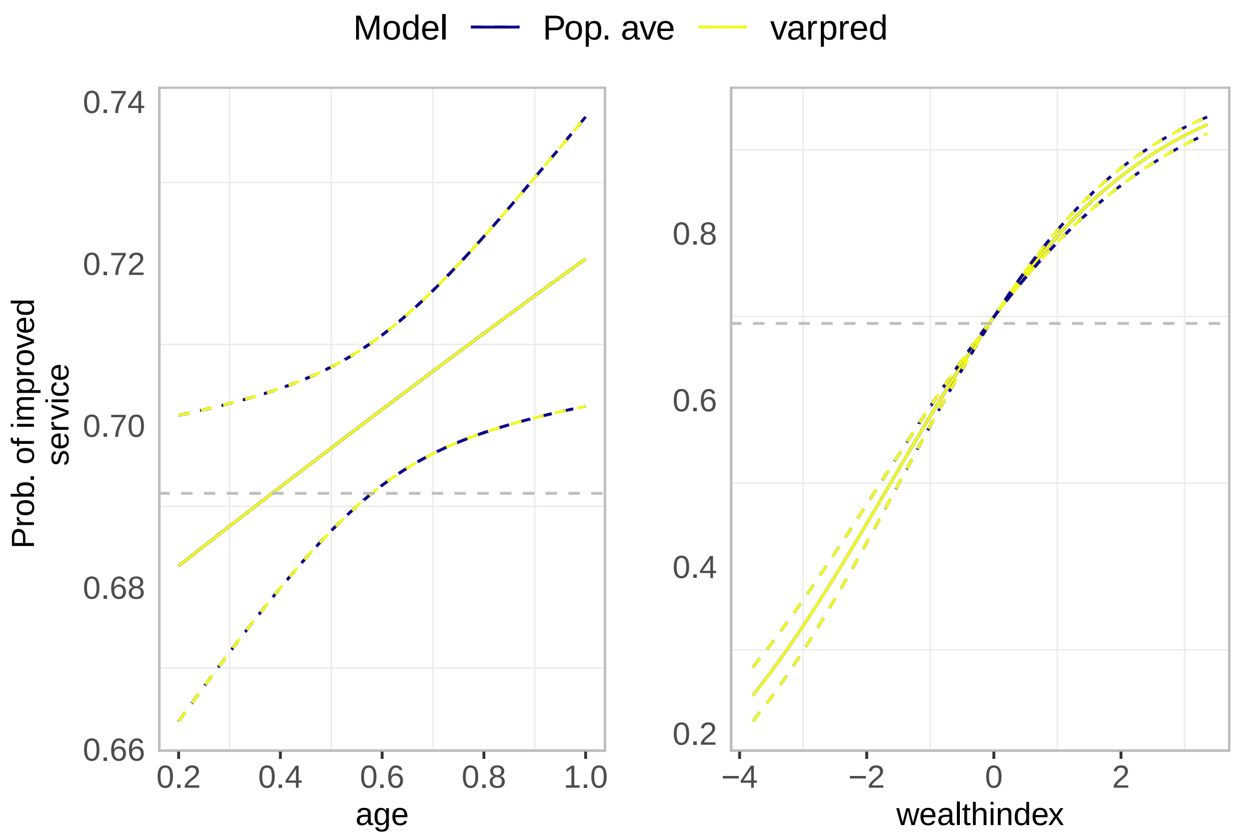


Figure 1: A comparison of population averaged and varpred-based predictions. For `age`, we implement the naive approach to compute the predictions in `popavefun` function and then implement the same in `vareffects` so as to use the centering machineries. In both cases, the population averaging and varpred gives similar estimates. The estimated population average is very close to the observed in the case of `wealthindex` but slightly higher in the case of `age`, see previous paragraph.