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

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

```r
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] 0.6916
```

```r
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

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

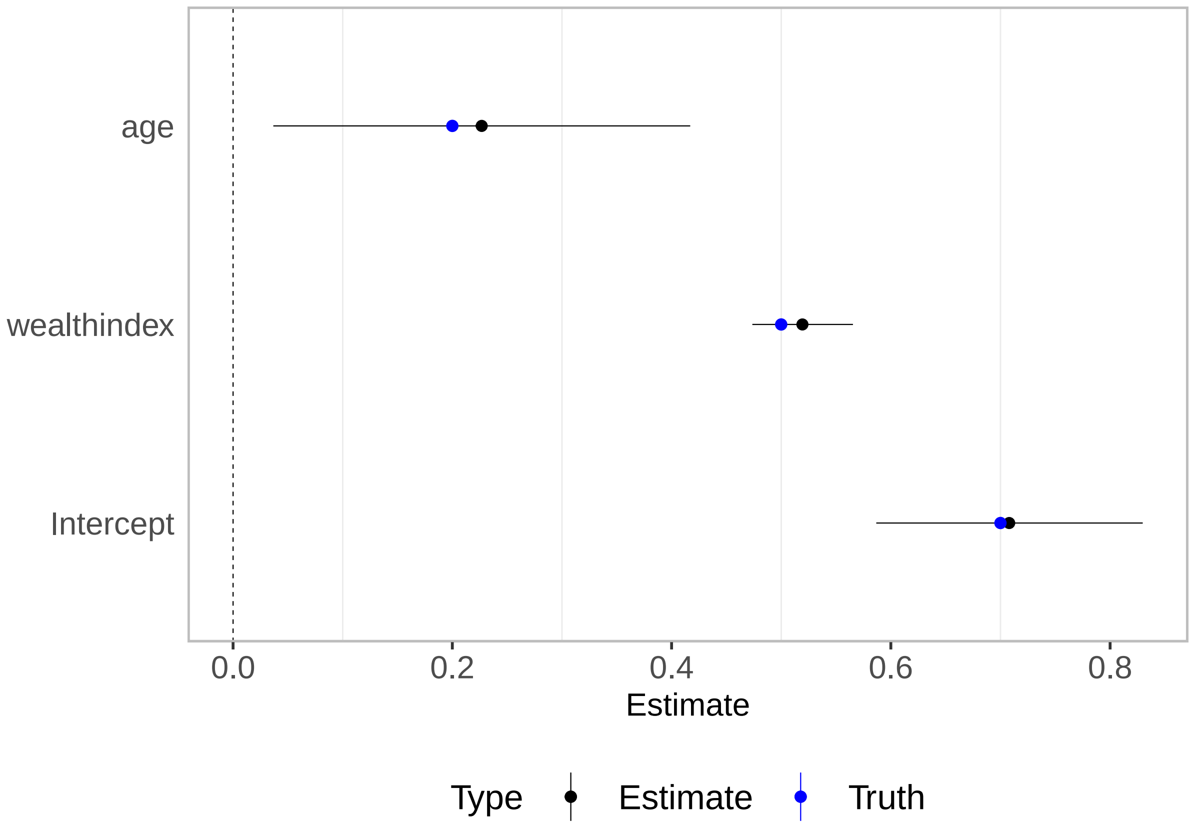
Coefficient plots

```r
## 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
```

```r
simple_coef_plot <- (plotEsize(coef_df)
	+ geom_point(data=true_beta_df, aes(x=term, y=estimate, colour="Truth"))
	+ labs(colour="Type")
	+ scale_colour_manual(values=c("black", "blue"))
)
print(simple_coef_plot)
```



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
```

Variable predictions

We consider both **varpred** and population averaging approach; and then introduce bias correction to **varpred** predictions.

```
popavefun <- function(mod, focal, non.focal, level=0.95, steps=100, bins=50, modelname="Pop. ave", ...)
    mf <- model.frame(mod)
    mm <- (mf
        %>% select_at(c(focal, non.focal))
    )
    check_df <- (mf
        %>% arrange_at(focal)
        %>% mutate(bin=ceiling(row_number()*bins/nrow(.)))
```

```
        %>% group_by(bin)
        %>% summarise_all(mean)
        %>% mutate(model="binned")
    )
    quant <- seq(0, 1, length.out=steps)
    mm <- sapply(c(focal, non.focal), function(x)quantile(mm[,x], quant), simplify = FALSE)
    mm <- do.call("expand.grid", mm)
    vc <- vcov(mod)
    mm2 <- model.matrix(formula(mod)[c(1,3)], mm)
    linpred <- as.vector(mm2 %*% coef(mod))
    pse_var <- sqrt(rowSums(mm2 * t(tcrossprod(data.matrix(vc), mm2))))
    z.val <- qnorm(1 - (1 - level)/2)
    pred_df <- (mm
        %>% select_at(focal)
        %>% mutate(fit = linpred
            , lwr = plogis(fit - z.val*pse_var)
            , upr = plogis(fit + z.val*pse_var)
            , fit = plogis(fit)
        )
        %>% group_by_at(focal)
        %>% summarise_at(c("fit", "lwr", "upr"), mean)
        %>% mutate(model = modelname, se=NA)
    )
    return(list(preds=pred_df, check=check_df))
}
```

- Age

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

## Pop. average
simple_vareff_age_pop <- varpred(simple_mod, "age", isolate=FALSE, pop.ave=TRUE, modelname="Pop. ave")
# simple_vareff_age_pop <- popavefun(simple_mod, "age", "wealthindex", modelname = "Pop. ave")
# binned_df <- simple_vareff_age_pop$check

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

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```

- Wealth index

```
# Wealth index
## varpred
simple_vareff_wealthindex <- varpred(simple_mod, "wealthindex", isolate=FALSE, modelname="varpred")

## Pop. average
simple_vareff_wealthindex_pop <- varpred(simple_mod, "wealthindex", isolate=FALSE, pop.ave=TRUE, modelna
```

```r
# binned_df <- simple_vareff_wealthindex_pop$check

vareff_wealthindex <- simple_vareff_wealthindex
vareff_wealthindex$preds <- do.call("rbind", list(vareff_wealthindex$preds, simple_vareff_wealthindex_po
wealthindex_plot <- (plot(vareff_wealthindex)
    + labs(y="", colour="Model")
    + geom_hline(yintercept=true_prop, lty=2, colour="grey")
#   + geom_point(data=binned_df, aes(x=wealthindex, y=status, color="binned"))
    + scale_colour_manual(values=c("black", "blue", "red"))
    + theme(legend.position="bottom")
)
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```

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

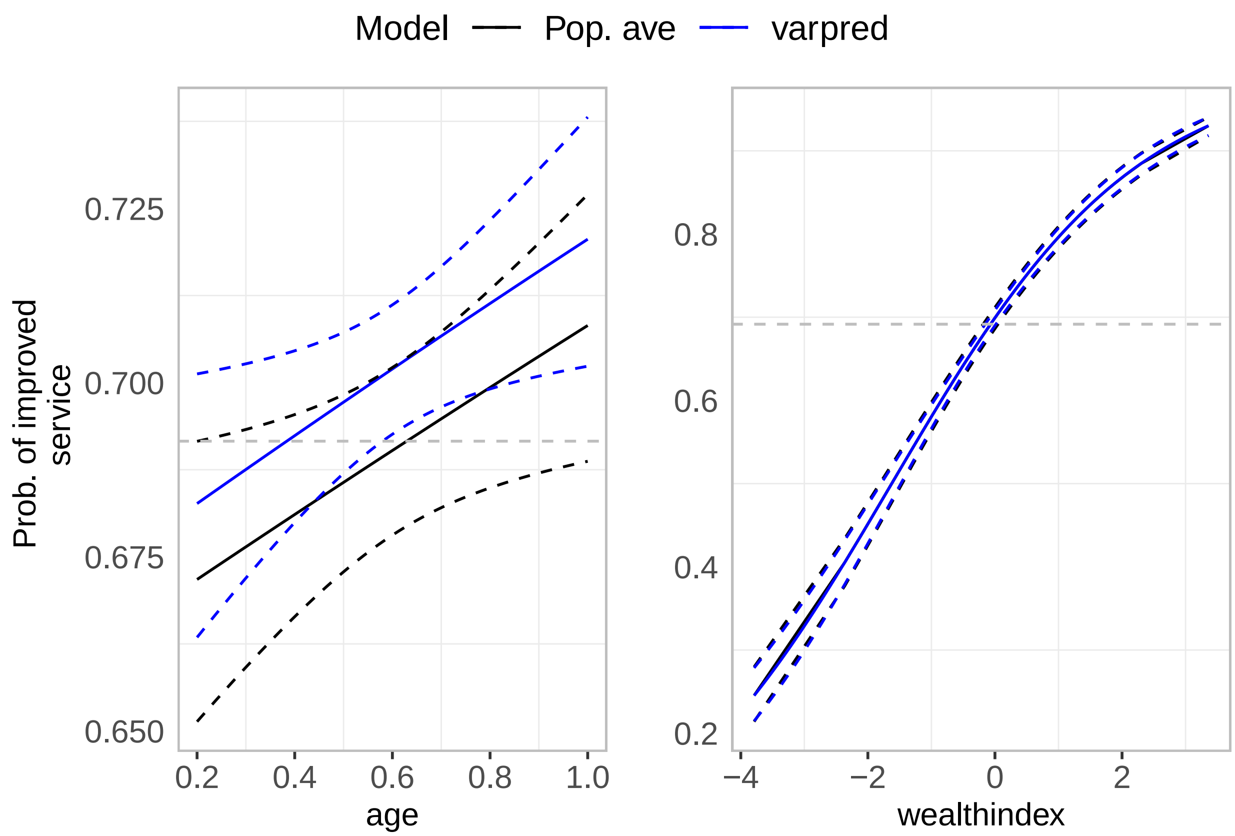


Figure 1: A comparison of population averaged and varpred-based predictions. In the case of biased predictions (age), population averaging gives better estimates. On the other hand, for wealthindex, the estimates varpred and population estimates are somehow similar.

The observed population average is 0.6916 while the estimated population averages (similar estimates with `varpred`) are:

- age: pop. average 0.690167; varpred 0.7018437
- wealthindex: pop. average 0.690167; varpred 0.6404765