All Interactions Are Wrong: Logit vs Probit

toffa @ psicostat

2025-01-14

Table of contents

1 Introduction 1

2 Scenario A: Data generating process showing a logit link function 1
- 2.1 Data simulation . . . 2
- 2.2 Correctly using a logit link function in GLM . . . 2
- 2.3 Incorrectly using a probit link function in GLM . . . 3

3 Scenario B: Data generating process showing a probit link function 4
- 3.1 Incorrectly using a logit link function in GLM . . . 4
- 3.2 Correctly using a probit link function in GLM . . . 5

1 Introduction

Under common conditions in which there are two main effects and no interaction, fitting a model with the wrong link function leads to detecting pseudointeractions; that is, to inflating the false positive rates for the interaction. This happen even if one fits a binomial GLM, but uses the "logit" link function when the data were generate using the "probit", or vice versa.

2 Scenario A: Data generating process showing a logit link function

In scenario A, the true data generating model is specified as follows:

$$\text{logit}(prob) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

Where:

- p is the probability of a positive response
- β_0 is the fixed intercept.
- β_1 and β_2 are the fixed effect coefficients for predictors X_1 and X_2, respectively.

Thus, the model incorporates fixed effects of X_1 and X_2, but not their interaction. For simplicity and computational speed we choose not to enter random effects, although they are frequently present in this scenario.

2.1 Data simulation

```
rm(list=ls())
library(ggplot2)
library(glmmTMB)
```

```
Warning in check_dep_version(dep_pkg = "TMB"): package version mismatch:
glmmTMB was built with TMB package version 1.9.16
Current TMB package version is 1.9.17
Please re-install glmmTMB from source or restore original 'TMB' package (see '?reinstalling'
```

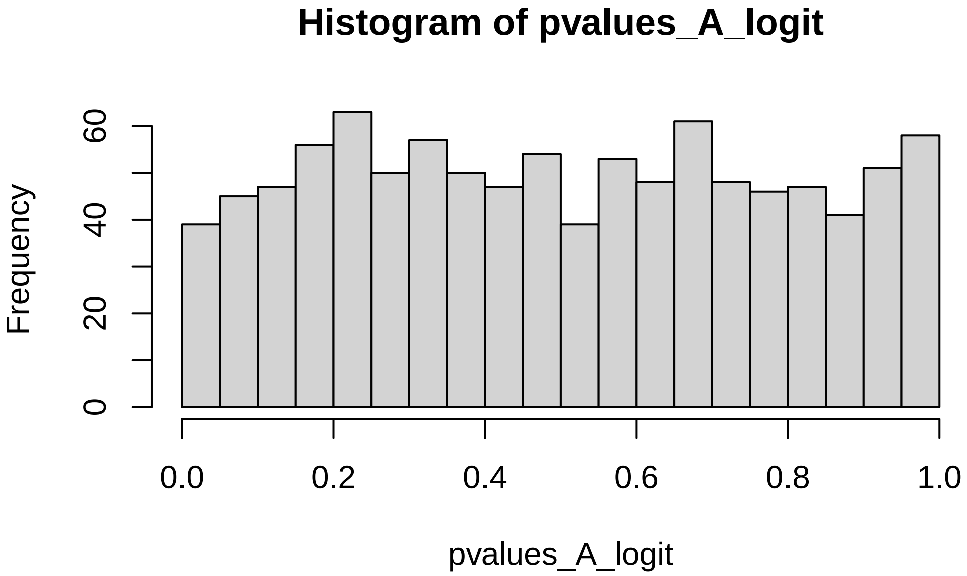
```
N = 1e3
X1 = rep(c(0,1),each=N/2)
X2 = rep(c(0,1),each=N/4,times=2)
df = data.frame(X1, X2, y=NA)
b0 = 0
b1 = 1
b2 = 1
lin_pred = b0 + b1*X1 + b2*X2
```

2.2 Correctly using a logit link function in GLM

```
niter = 1000
pvalues_A_logit = rep(NA,niter)

for(i in 1:niter){
  prob = plogis(lin_pred)
  y = rbinom(N,1,prob)
  df$y = y
  fit = glmmTMB(y ~ X1 * X2, data=df, family=binomial(link="logit"))
  pvalues_A_logit[i] = summary(fit)$coefficients$cond["X1:X2","Pr(>|z|)"]
}
```

```
hist(pvalues_A_logit,breaks=20)
```

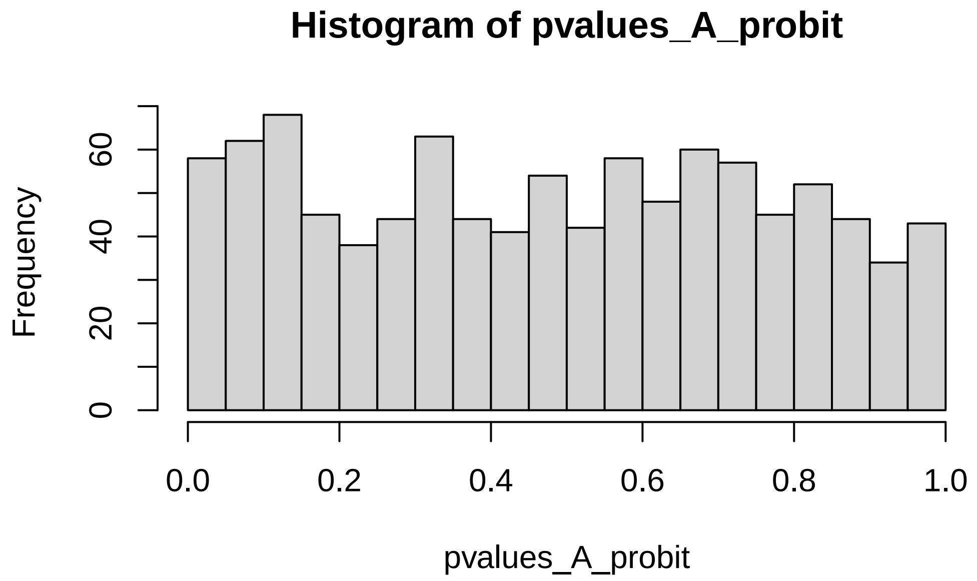


2.3 Incorrectly using a probit link function in GLM

```
niter = 1000
pvalues_A_probit = rep(NA,niter)

for(i in 1:niter){
  prob = plogis(lin_pred)
  y = rbinom(N,1,prob)
  df$y = y
  fit = glmmTMB(y ~ X1 * X2, data=df, family=binomial(link="probit"))
  pvalues_A_probit[i] = summary(fit)$coefficients$cond["X1:X2","Pr(>|z|)"]
}
```

```
hist(pvalues_A_probit,breaks=20)
```



3 Scenario B: Data generating process showing a probit link function

Scenario B is exactly identical to scenario A, except that the data generating process features a “probit” (instead of “logit”) link function.

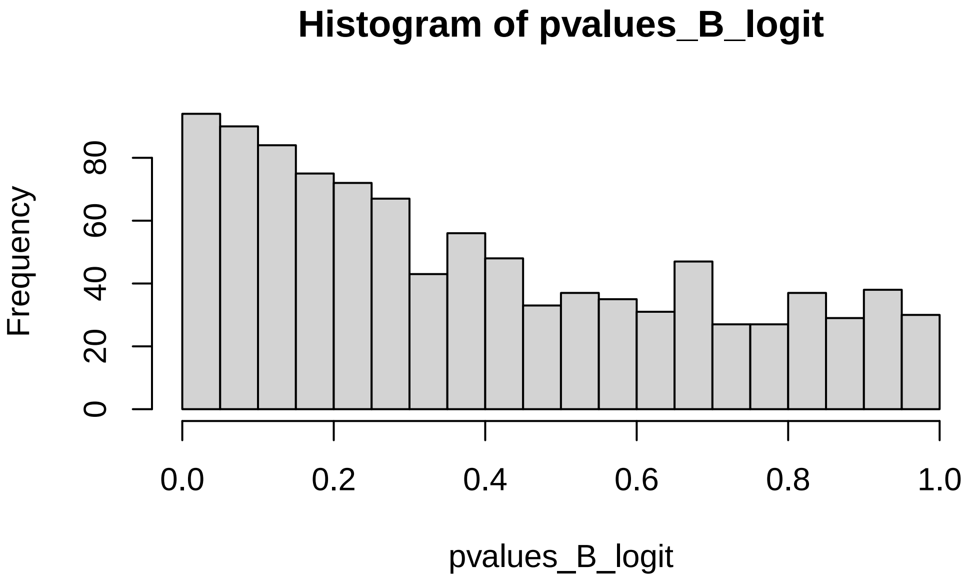
$$\text{logit}(prob) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

3.1 Incorrectly using a logit link function in GLM

```
niter = 1000
pvalues_B_logit = rep(NA,niter)

for(i in 1:niter){
  prob = pnorm(lin_pred)
  y = rbinom(N,1,prob)
  df$y = y
  fit = glmmTMB(y ~ X1 * X2, data=df, family=binomial(link="logit"))
  pvalues_B_logit[i] = summary(fit)$coefficients$cond["X1:X2","Pr(>|z|)"]
}
```

```
hist(pvalues_B_logit,breaks=20)
```

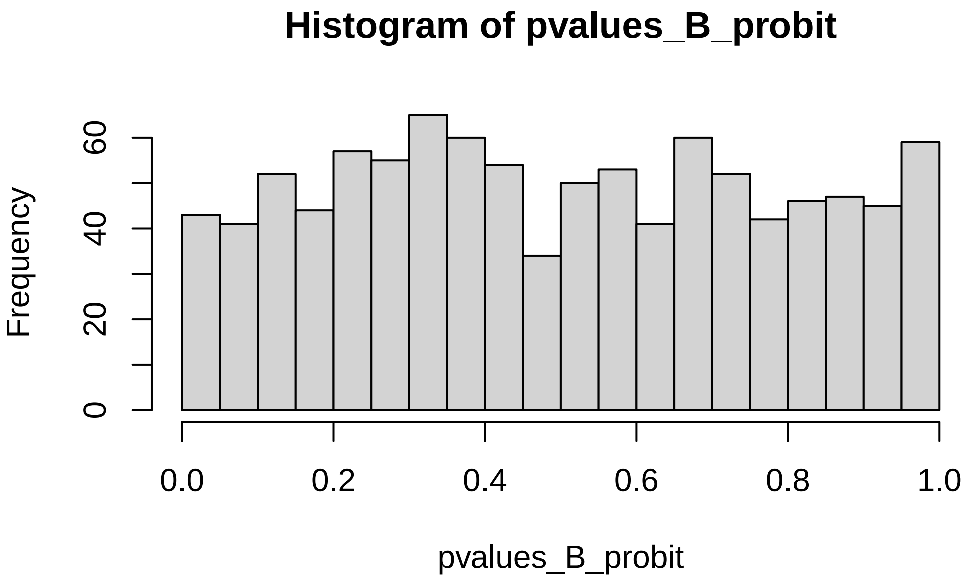


3.2 Correctly using a probit link function in GLM

```
niter = 1000
pvalues_B_probit = rep(NA,niter)

for(i in 1:niter){
  prob = pnorm(lin_pred)
  y = rbinom(N,1,prob)
  df$y = y
  fit = glmmTMB(y ~ X1 * X2, data=df, family=binomial(link="probit"))
  pvalues_B_probit[i] = summary(fit)$coefficients$cond["X1:X2","Pr(>|z|)"]
}
```

```
hist(pvalues_B_probit,breaks=20)
```


Histogram of pvalues_B_probit
Frequency
0
20
40
60
0.0
0.2
0.4
0.6
0.8
1.0
pvalues_B_probit