Methods Demonstration: Selection and Misclassification Bias Adjustment for EHR Data Analysis

Created by Dr. Lauren J Beesley. Contact: lbeesley@umich.edu

29 June, 2021

In this document, we provide a demonstration of how we can apply methods proposed in 'Statistical inference for association studies using electronic health records: handling both selection bias and outcome misclassification" by Lauren J Beesley and Bhramar Mukherjee in *Biometrics* to address selection bias and outcome misclassification. This work uses function from companion R package *SAMBA* (selection and misclassification bias adjustment) currently on CRAN. We will not include technical details about this estimation approach here, and instead we will focus on implementation. For additional details about the estimation algorithm, we refer the reader to the methods manuscript.

Model structure

Let binary D represent a patient's true disease status for a disease of interest, e.g. diabetes. Suppose we are interested in the relationship between D and and person-level information, Z. Z may contain genetic information, lab results, age, gender, or any other characteristics of interest. We call this relationship the *disease mechanism* as in **Figure 1**.

Suppose we consider a large health care system-based database with the goal of making inference about some defined general population. Let S indicate whether a particular subject in the population is sampled into our dataset (for example, by going to a particular hospital and consenting to share biosamples), where the probability of an individual being included in the current dataset may depend on the underlying lifetime disease status, D, along with additional covariates, W. W may also contain some or all adjustment factors in Z. In the EHR setting, we may often expect the sampled and non-sampled patients to have different rates of the disease, and other factors such as patient age, residence, access to care and general health state may also impact whether patients are included in the study base or not. We will call this mechanism the *selection mechanism.*

Instances of the disease are recorded in hospital or administrative records. We might expect factors such as the patient age, the length of follow-up, and the number of hospital visits to impact whether we actually *observe/record* the disease of interest for a given person. Let D^* be the *observed* disease status. D^* is a potentially misclassified version of D. We will call the mechanism generating D^* the *observation mechanism.* We will assume that misclassification is primarily through underreporting of disease. In other words, we will assume that D^* has perfect specificity and potentially imperfect sensitivity with respect to D. Let X denote patient and provider-level predictors related to the true positive rate (sensitivity).

We express the conceptual model as follows:

$$\text{Disease Model} : \text{logit}(P(D = 1|Z; \theta)) = \theta_0 + \theta_Z Z$$

$$\text{Selection Model} : P(S = 1|D, W; \phi)$$

$$\text{Sensitivity/Observation Model} : \text{logit}(P(D^* = 1|D = 1, S = 1, X; \beta)) = \beta_0 + \beta_X X$$

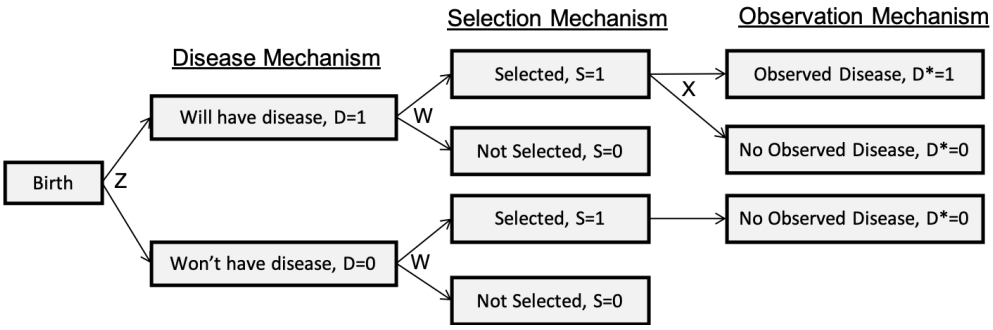


Figure 1: Figure 1: Model Structure

Simulate data

We start our exploration by simulating some binary data subject to misclassification of the outcome and selection bias. Variables related to selection are D and W. Variables related to sensitivity are X. Variables of interest are Z, which are related to W and X. In this simulation, W is also independently related to D.

```
library(SAMBA)
library(MASS)
expit <- function(x) exp(x) / (1 + exp(x))
logit <- function(x) log(x / (1 - x))

nobs <- 5000

### Generate Predictors and Follow-up Information
set.seed(1234)
cov <- mvrnorm(n = nobs, mu = rep(0, 3), Sigma = rbind(c(1,   0, 0.4),
                                                       c(0,   1,   0),
                                                       c(0.4, 0,   1)))
data <- data.frame(Z = cov[, 1], X = cov[, 2], W = cov[, 3])

# Generate random uniforms
set.seed(5678)
U1 <- runif(nobs)
set.seed(4321)
U2 <- runif(nobs)
set.seed(8765)
U3 <- runif(nobs)

# Generate Disease Status
DISEASE <- expit(-2 + 0.5 * data$Z)
data$D   <- ifelse(DISEASE > U1, 1, 0)

# Relate W and D
data$W <- data$W + 1 * data$D

# Generate Misclassification
SENS <- expit(-0.4 + 1 * data$X)
SENS[data$D == 0] = 0
data$Dstar <- ifelse(SENS > U2, 1, 0)
```

```r
# Generate Sampling Status
SELECT <- expit(-0.6 + 1 * data$D + 0.5 * data$W)
S  <- ifelse(SELECT > U3, T, F)

# Observed Data
data.samp <- data[S,]

# True marginal sampling ratio
prob1 <- expit(-0.6 + 1 * 1 + 0.5 * data$W)
prob0 <- expit(-0.6 + 1 * 0 + 0.5 * data$W)
r.marg.true <- mean(prob1[data$D == 1]) / mean(prob0[data$D == 0])

# True inverse probability of sampling weights
prob.WD <- SELECT[S]
data.samp$weights <- nrow(data.samp) * (1  / prob.WD) / (sum(1 / prob.WD))

# True associations with D in population
trueX <- glm(D ~ X, binomial(), data = data)
trueZ <- glm(D ~ Z, binomial(), data = data)

# Initial Parameter Values
fitBeta  <- glm(Dstar ~ X, binomial(), data = data.samp)
fitTheta <- glm(Dstar ~ Z, binomial(), data = data.samp)
```

Now, we generate an example external data source that will be used to estimate weights for selection bias adjustment. We generate this external data from a population with the same variable distributions as the source population for the analytical dataset used above. Unlike the analytical sample, we assume selection weights are available for the external probability sample.

```r
nobs_ext <- 5000

### Generate Predictors and Follow-up Information
set.seed(1243)
cov <- mvrnorm(n = nobs_ext, mu = rep(0, 3), Sigma = rbind(c(1,   0, 0.4),
                                                           c(0,   1,   0),
                                                           c(0.4, 0,   1)))
data_target <- data.frame(Z = cov[, 1], X = cov[, 2], W = cov[, 3])

# Generate random uniforms
set.seed(5687)
U1 <- runif(nobs_ext)
set.seed(4312)
U2 <- runif(nobs_ext)

# Generate Disease Status
DISEASE <- expit(-2 + 0.5 * data_target$Z)
data_target$D   <- ifelse(DISEASE > U1, 1, 0)

# Relate W and D
data_target$W <- data_target$W + 1 * data_target$D
data_target$Dstar = NA

# Generate Sampling Status
SELECT_EXTERNAL <- expit( data_target$D - 0.5*data_target$W)
```

```r
S_ext  <- ifelse(SELECT_EXTERNAL > U2, T, F)

# Observed Data
data.samp.external <- data_target[S_ext,]

# True inverse probability of sampling weights
prob.external <- SELECT_EXTERNAL[S_ext]
weights.external <- 1  / prob.external
data.samp.external$weights = sum(S_ext) *weights.external / (sum(weights.external))
data.samp.external$samp = prob.external
```

Estimating sensitivity

In our paper, we develop several strategies for estimating either marginal sensitivity or sensitivity as a function of covariates X. Here, we apply the proposed strategies.

```
# Using marginal sampling ratio r and P(D=1)
sens1 <- sensitivity(data.samp$Dstar, data.samp$X, prev = mean(data$D),
                     r = r.marg.true)

# Using marginal sampling ratio r and P(D=1|X)
prev  <- predict(trueX, newdata = data.samp, type = 'response')
sens2 <- sensitivity(data.samp$Dstar, data.samp$X, prev = prev, r = r.marg.true)
```

Estimating weights for selection bias adjustment

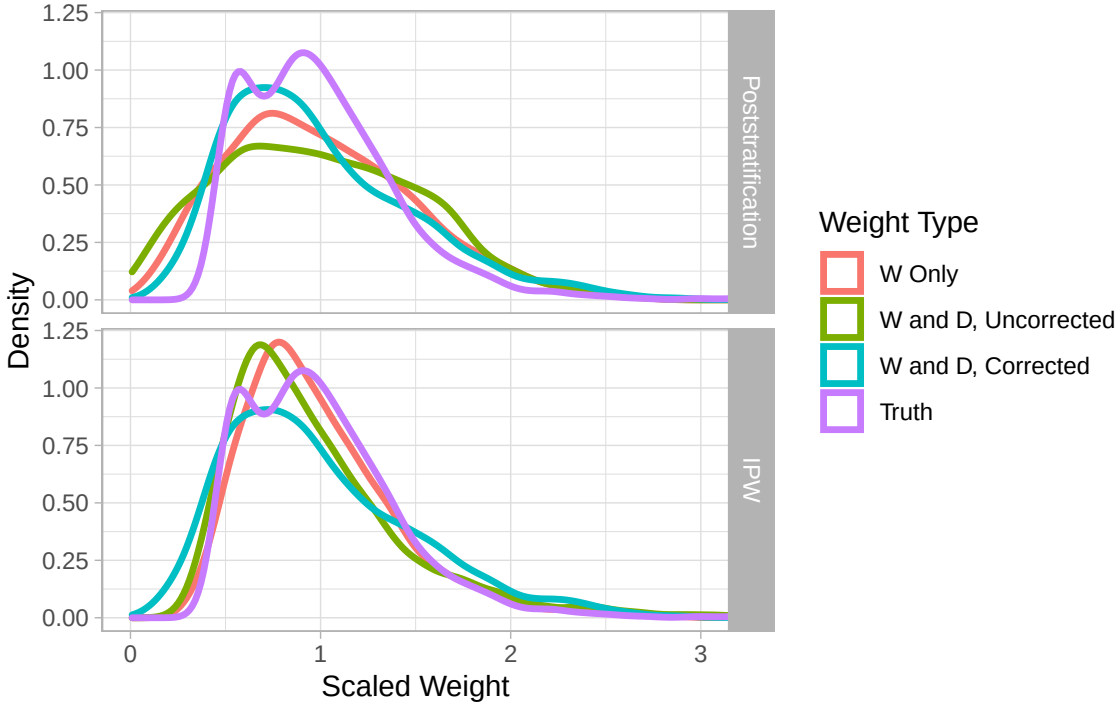
In the manuscript, we describe how summary statistics from the target population or a probability sample of individual-level data from the target population can be used to estimate weights for selection bias correction. Several of these strategies are implemented below. **Figure 2** shows the resulting weight estimates.

```
### Distribution of W in target population:
target_W = function(x){ dnorm(x, mean = 0, sd = 1)}
### Distribution of D|W in target population:
fit_target = glm(D~W, family = binomial(), data = data_target)
target_DW = function(x,w){
  preds = predict(fit_target,newdata = data.frame(W=w), type = 'response')
  return(ifelse(x==1,preds,1-preds))
}

### Distribution of W|S=1 in internal data:
internal_W = function(x){ dnorm(x, mean = mean(data.samp$W), sd = sd(data.samp$W))}
### Distribution of D|W,S=1 in internal data:
fit_internal = glm(Dstar~W, family = binomial(), data = data.samp)
internal_DW = function(x, w){
  preds = predict(fit_internal,newdata = data.frame(W=w), type = 'response')
  return(ifelse(x==1,preds,1-preds))
}

### Poststratification Weights, W Only
post_W = target_W(data.samp$W)/internal_W(data.samp$W)
post_W = sum(S)*post_W/sum(post_W)

### Poststratification Weights, D and W (No Misclassification Correction)
post_DW = target_DW(data.samp$Dstar, data.samp$W)/internal_DW(data.samp$Dstar, data.samp$W)
post_DW = post_DW*target_W(data.samp$W)/internal_W(data.samp$W)
post_DW = sum(S)*post_DW/sum(post_DW)

### Poststratification Weights, D and W (Misclassification Correction)
numerator = ifelse(data.samp$Dstar==1,
               sens2$c_X*target_DW(data.samp$Dstar, data.samp$W),
               1-sens2$c_X*target_DW(data.samp$Dstar, data.samp$W))
post_DW_CORRECTED = post_W*numerator/internal_DW(data.samp$Dstar, data.samp$W)
post_DW_CORRECTED = sum(S)*post_DW_CORRECTED/sum(post_DW_CORRECTED)

### IPW, W Only
MERGED = rbind(data.frame(dataset = 'External', data.samp.external[,c('W','D','samp')]),
               data.frame(dataset = 'Internal', W=data.samp$W, D=data.samp$Dstar,samp=NA))
selection_external = betareg::betareg(samp~W, data = data.samp.external)
selection_internal_vs_external = glm(as.numeric(dataset == 'Internal' )~W,
               data = MERGED, family = 'binomial')
p_external = predict(selection_external, newdata = MERGED[MERGED$data == 'Internal',],
               type = 'response')
p_internal_vs_external = predict(selection_internal_vs_external,
               newdata = MERGED[MERGED$data == 'Internal',], type = 'response' )
SELECT_NHANES_NOCAN = p_external*(p_internal_vs_external/(1-p_internal_vs_external))
ipw_W = (1-p_internal_vs_external)/(p_external*p_internal_vs_external)
ipw_W = sum(S)*ipw_W/sum(ipw_W)
```

```r
### IPW, D and W (No Misclassification Correction)
selection_external = betareg::betareg(samp~D+W, data = data.samp.external, link = 'probit')
selection_internal_vs_external = glm(as.numeric(dataset == 'Internal' )~D+W,
               data = MERGED, family = 'binomial')
p_external = predict(selection_external, newdata = MERGED[MERGED$data == 'Internal',],
               type = 'response')
p_internal_vs_external = predict(selection_internal_vs_external,
               newdata = MERGED[MERGED$data == 'Internal',], type = 'response' )
SELECT_NHANES_NOCAN = p_external*(p_internal_vs_external/(1-p_internal_vs_external))
ipw_DW = (1-p_internal_vs_external)/(p_external*p_internal_vs_external)
ipw_DW = sum(S)*ipw_DW/sum(ipw_DW)

### IPW, D and W (Misclassification Correction)
### Distribution of D|W,Sext=1:
fit_external = glm(D~W, family = binomial(), data = data.samp.external,
               weights = data.samp.external$weights)
external_DW = function(x,w){
  preds = predict(fit_external,newdata = data.frame(W=w), type = 'response')
  return(ifelse(x==1,preds,1-preds))
}
numerator = ifelse(data.samp$Dstar==1,
               sens2$c_X*external_DW(data.samp$Dstar, data.samp$W),
               1-sens2$c_X*external_DW(data.samp$Dstar, data.samp$W))
ipw_DW_CORRECTED = post_W*numerator/internal_DW(data.samp$Dstar, data.samp$W)
ipw_DW_CORRECTED = sum(S)*ipw_DW_CORRECTED/sum(ipw_DW_CORRECTED)
```

Below, we plot the resulting weight estimates. None of the data-generated weights exactly reproduce the true weights, but they are generally close. Weights for individuals can vary substantially between methods.

Figure 2: Distributions of Scaled Weights



Estimating log-odds ratio for D|Z

We propose several strategies for estimating the association between D and Z from a logistic regression model in the presence of different biasing factors. Below, we provide code for implementing these methods. For demonstration, we will use the "ideal" true selection weights.

```
# Approximation of D*|Z
approx1 <- approxdist(data.samp$Dstar, data.samp$Z, sens1$c_marg,
                      weights = data.samp$weights)

# Non-logistic link function method
nonlog1 <- nonlogistic(data.samp$Dstar, data.samp$Z, c_X = sens2$c_X,
                       weights = data.samp$weights)

# Direct observed data likelihood maximization without fixed intercept
start <- c(coef(fitTheta), logit(sens1$c_marg), coef(fitBeta)[2])
fit1 <- obsloglik(data.samp$Dstar, data.samp$Z, data.samp$X, start = start,
                  weights = data.samp$weights)
obsloglik1 <- list(param = fit1$param, variance = diag(fit1$variance))

# Direct observed data likelihood maximization with fixed intercept
fit2   <- obsloglik(data.samp$Dstar, data.samp$Z, data.samp$X, start = start,
                  beta0_fixed = logit(sens1$c_marg), weights = data.samp$weights)
obsloglik2 <- list(param = fit2$param, variance = diag(fit2$variance))
```

Plotting sensitivity estimates

Figure 3 shows the estimated individual-level sensitivity values when the marginal sampling ratio (r-tilde) is correctly specified. We can see that there is strong concordance with the true sensitivity values.

Figure 3: Sensitivity Estimates

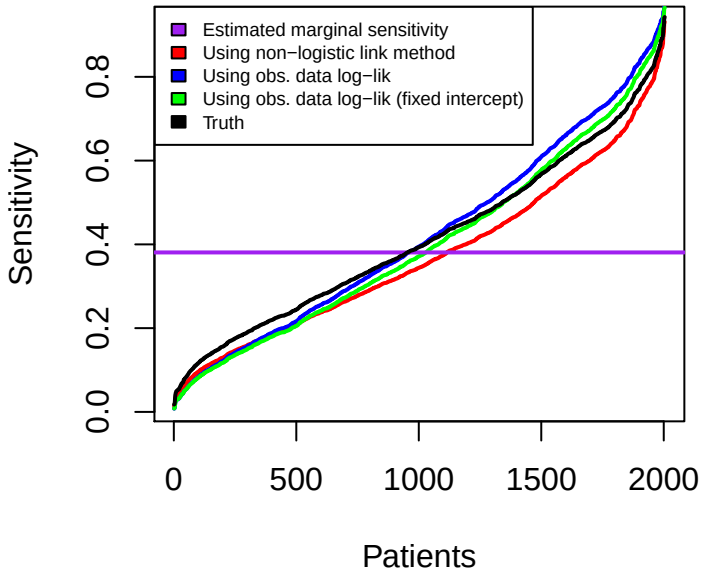
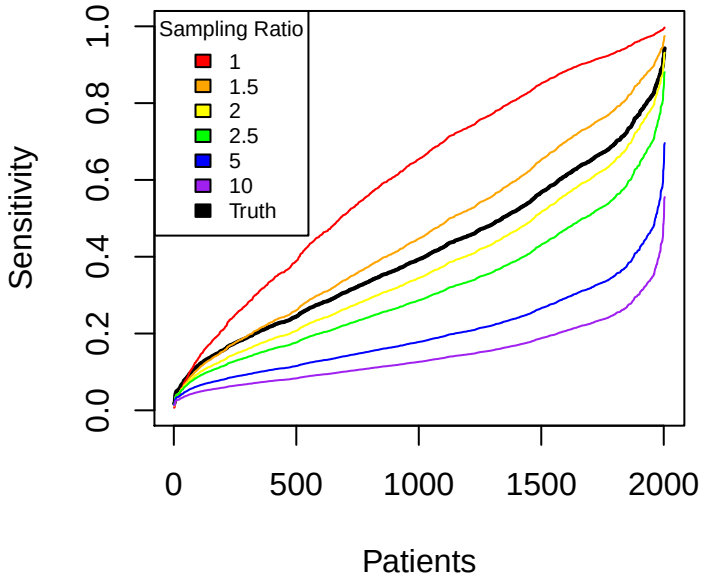


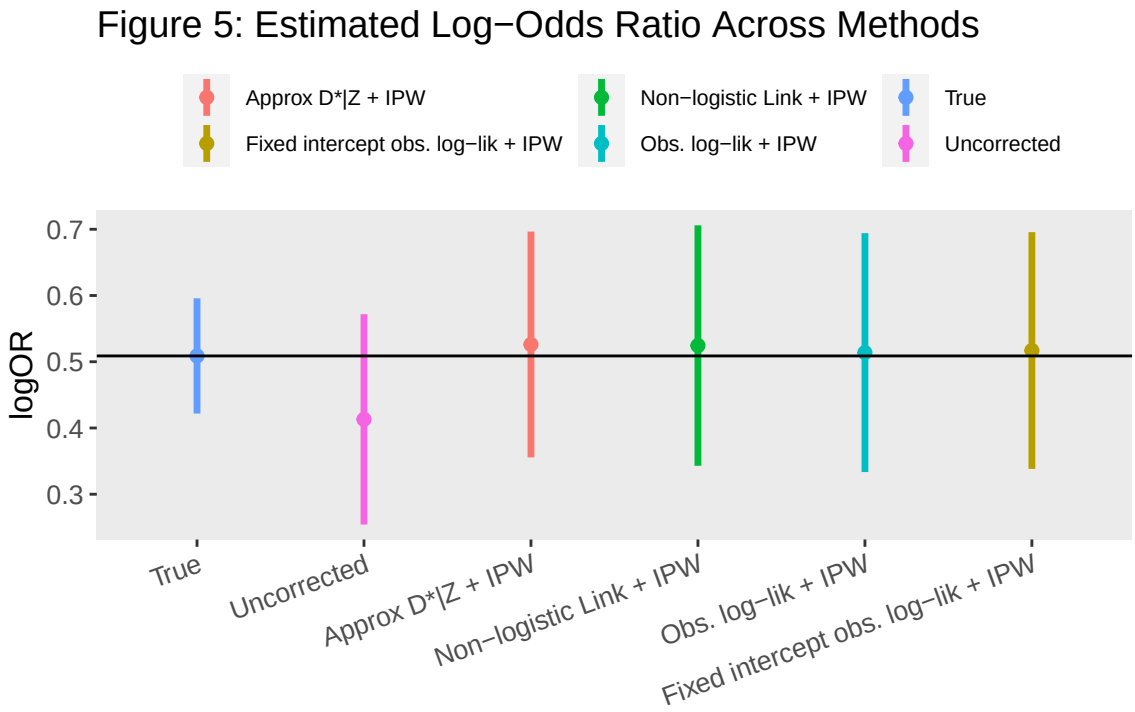
Figure 4 shows the estimated individual-level sensitivity values across different marginal sampling ratio values. In reality, we will rarely know the truth, and this strategy can help us obtain reasonable values for sensitivity across plausible sampling ratio values.

Figure 4: Estimated sensitivity across marginal sampling ratios



Plotting log-odds ratio estimates

Figure 5 shows the estimated log-odds ratio relating D and Z for the various analysis methods. Uncorrected (complete case with misclassified outcome) analysis produces bias, and some methods reduce this bias. Recall, this is a single simulated dataset, and the corrected estimators may not always equal the truth for a given simulation. When W and D are independently associated, the method using the approximated $D^*|Z$ relationship and marginal sensitivity can sometimes perform poorly.



Reference

Statistical inference for association studies using electronic health records: handling both selection bias and outcome misclassification by Lauren J Beesley and Bhramar Mukherjee, available at 10.1111/biom.13400