Misclassification Activity
JSM 2024 Short Course

Paul Gustafson

August 3, 2024

Recall this example

```
data.tbl <- matrix(c(45,94,257,945),
  dimnames = list(c("CHD+", "CHD-"),c("Resin+", "Resin-")),
  nrow = 2, byrow = TRUE)
```

```
data.tbl
```

```
##       Resin+ Resin-
## CHD+      45     94
## CHD-     257    945
```

Naive analysis presuming correct exposure classification

Inference for exposure-disease odds-ratio

```
logOR.hat <- sum(c(1,-1,-1,1)*log(as.vector(data.tbl)))

logOR.SE <- sqrt(sum(1/as.vector(data.tbl)))

exp(logOR.hat + c(0, -1.96, 1.96)*logOR.SE)
```

```
## [1] 1.76 1.20 2.58
```

Assuming nondifferential exposure misclassification with 90% sensitivity and 80% specificity

Again, recall from slides:

```
require(episensr)

ft <- misclassification(data.tbl,
 type="exposure", bias_parms=c(0.9, 0.9, 0.8, 0.8))

# point and interval estimation of OR
ft$adj.measures[2,]
```

```
##          2.5% 97.5%
## 10.67  1.64 69.55
```

Activity A

Check you can reproduce one of the **differential** classification adjustments given in the slides (i.e., one of the off-diagonal table entries on slide 151).

For instance, try presuming 90% specificity for all subjects, but sensitivity of 90% for controls, compared to 80% for cases.

Might help:

```
help(misclassification)
```

Activity B: Uncertainty about misclassification rates

Say the investigator is confident that the misclassification is nondifferential.

Has 85% sensitivity and 85% specificity as "best guesses."

But thinks either guess could be off by as much as five percentage points.

Can you look at

```
help(probsens)
```

and then provide an appropriate analysis?

HINT: First example in the help gives a template.

HINT: For simplicity, maybe "triangular" or "uniform" instead of "trapezoidal"

Activity C - Role of data

We have useful heuristics in statistics, such as the primal role of $\sqrt{n}$.

If I want interval estimates *twice* as narrow, I likely need about *four times* as much data.

Repeat Activity B, but with four times as much data. (Simplest to just keep cell *proportions* fixed in the 2 by 2 data table).

Reflect on what you find.

Activity D - Bayesian approach

Factor the joint dist. of (X, X^*, Y) in terms of (Y), $(X|Y)$, $(X^*|X, Y)$.

Leave (Y) as unmodeled [since Y observed, and given case-control design, parameter of most interest is determined by $X|Y$)].

Parameterize as $Pr(X = 1|Y = y) = r_y$, and

$$Pr(X^* = X|X = x, Y = y) = \begin{cases} Sp & \text{if } x = 0, \\ Sn & \text{if } x = 1. \end{cases}$$

Parameter of most interest: $\psi = \log OR(X, Y) = \text{logit}(r_1) - \text{logit}(r_0)$.

Priors $r_j \sim \text{Unif}(0, 1)$, $Sn \sim \text{beta}(a_{sn}, b_{sn})$,$Sp \sim \text{beta}(a_{sp}, b_{sp})$.

Activity D, continued

A bit too much overhead with trying to get JAGS/rJAGS going in a matter of minutes (unless you have experience. . .)

Here (meaning sitting in the .Rmd file generating these slides) is a bespoke R function (called **bespoke()**) to do MCMC for this model/prior only:

bespoke()

```
bespoke <- function(n.0, n.1, mstr.0, mstr.1,
                    a.sn, b.sn, a.sp, b.sp,
                    N.REP=50000, N.BURN=100) {

  ### INPUTS
  ### n.j is size of control (j=0) and case (j=1) samples
  ### mstr.j is number (out of n.j) of apparently exposed
  ### a.sn, b.sn, a.sp, b.sp are hyperparameters

  ### LATENTS
  ### m.j is number (out of n.j) actually exposed
  ### t.j is number (out of mstr.j) actually exposed
  ###     amongst the apparents

  ### OUTPUT output will be matrix, MC sample from posterior
  ans <- matrix(NA, N.REP, 8)
  colnames(ans) <- c("r0","r1","sn","sp","m0","m1","t0","t1")
```

Example

Say I am pretty sure that the exposure classification is very good (though probably not perfect). I encode this with priors $Sn \sim \text{Beta}(140, 10)$, $Sp \sim \text{Beta}(140, 10)$.

Sidenote: As a thought experiment, this would formally be the evidence had we done an external validation of 150 truly unexposed and 150 truly exposed individuals, and found 10 misclassifications in each group.

```
mc.opt <- bespoke(n.0=257+945, n.1=45+94,
                  mstr.0=257, mstr.1=45,
                  a.sn=140, b.sn=10, a.sp=140, b.sp=10)
```

Example, continued

```
### focus on target parameter
mc.trg <- logit(mc.opt[,"r1"]) - logit(mc.opt[,"r0"])
summary(mc.trg)
```

```
##     Min. 1st Qu.   Median     Mean 3rd Qu.     Max.
##  -0.513   0.570   0.736   0.739   0.905   2.150
```

```
### estimate OR
exp(mean(mc.trg))
```

```
## [1] 2.09
```

```
### corresponding 95% credible interval
exp(quantile(mc.trg, c(0.025, 0.975)))
```

```
##  2.5% 97.5%
##  1.28  3.46
```

Activity D - you try

Can you carry out a Bayesian analysis with *about* the same sort of uncertainty about misclassification parameters as in Activity B.

Activity D - further thinking/doing points

Depending on your background and interests, you could:

- Look at the code for **bespoke()** to confirm how the MCMC algorithm (in this case the Gibbs sampler) bounces back and forth between sampling complete data given parameters, and sampling parameters given complete data.
- Take a closer look at the Monte Carlo output to confirm that you do not get the luxury of *iid* draws from the posterior distribution, but rather have to live with serially autocorrelated draws.
- Take a closer look at the Monte Carlo output to consider what the posterior distribution of *Sn* and *Sp* looks like, compared to the prior.