To Bound or not to Bound : Technical Vignette

Paul Gustafson

5/7/2021

Set the thought-experiment world as we wish, by specified exposure prevalence in the control population, and specified exposure-disease odds ratio:

```
trg.tr <- log(1.30)
r.tr <- c(0.15, NA)
r.tr[2] <- expit(logit(r.tr[1]) + trg.tr)
r.tr
```

```
## [1] 0.1500000 0.1866029
```

Study 2 involves differential misclassification, with the following true values, and (valid) prior lower bounds, for sensitivity and specificity:

```
sn.tr <- c(.92, .99)
sp.tr <- c(.99, .99)

sn.lwr <- c(.90, .90)
sp.lwr <- c(.95, .95)
```

As the sample size increases, Study 2 sees this apparent effect (log OR and OR) scales

```
r.app.2 <- r.tr * sn.tr + (1-r.tr)*(1-sp.tr)
trg.app.2 <- logit(r.app.2[2]) - logit(r.app.2[1])
c(trg.app.2, exp(trg.app.2))
```

```
## [1] 0.330857 1.392161
```

Bounds (given that for fixed apparent prevalence, true prevalence increases in Sp, decreases in Sn):

```
### sp=1, sn=sn.lwr gives upper bound for both prevalences
r.upr.2 <- (r.app.2 + 1 - 1)/(sn.lwr+1-1)

### sp=sp.lwr, sn=1 gives lower bounds for both prevalences
r.lwr.2 <- (r.app.2 + sp.lwr - 1)/(1+sp.lwr-1)

rbind(r.lwr.2, r.upr.2)
```

```
##              [,1]      [,2]
## r.lwr.2 0.1015789 0.1503903
## r.upr.2 0.1627778 0.2143009
```

```
### ergo bounds on the log OR

bnd.2 <- c(
  logit(r.lwr.2[2])-logit(r.upr.2[1]),
  logit(r.upr.2[2])-logit(r.lwr.2[1]))
```

```
rbind(bnd.2, exp(bnd.2))
```

```
##              [,1]      [,2]
## bnd.2 -0.09383933 0.8806098
##        0.91042903 2.4123702
```

Proportion of Study 2 ID interval crossing null:

```
-bnd.2[1]/(bnd.2[2]-bnd.2[1])
```

```
## [1] 0.09629988
```

Study 1 is imperfect via selection bias in the case population only. Amongst those diseased, can rule out that exposure status influences participation probability, in one direction or other. This bias parameter is coded as a log risk ratio, e.g, $\gamma = \log Pr(S=1|X=1,Y=1)/\log Pr(S=1|X=0,Y=1)$.

For pedagogical purposes, we reverse-engineer both the true value and prior bounds on γ such that Study 1 and Study 2 yield the same identification interval. This is fixing three things to match two values. We fix the lower bound arbitrarily, then compute the true value and upper bound to make the intervals match.

```
gamma.lwr <- log(.7)
gamma.upr <- gamma.lwr + (bnd.2[2]-bnd.2[1])
gamma.tr <- bnd.2[1] - trg.tr + gamma.upr

c(gamma.lwr, gamma.tr, gamma.upr)
```

```
## [1] -0.3566749  0.2615706  0.6177741
```

So as then sample size increases, Study 1 reports an apparent estimate, and a bound of:

```
trg.app.1 <- trg.tr +  gamma.tr
bnd.1 <- trg.app.1 - c(gamma.upr, gamma.lwr)
c(trg.app.1,bnd.1)   ## log OR scale
```

```
## [1]  0.52393482 -0.09383933  0.88060976
```

```
exp(c(trg.app.1,bnd.1))
```

```
## [1] 1.688659 0.910429 2.412370
```

Now we turn to Bayesian analysis with uniform priors between the prior bounds, for either γ (in Study 1) or Sn_0, Sn_1, Sp_0, Sp_1 (in Study 2). In both cases r_0 and r_1 have Unif$(0,1)$ priors.

For Study 1 we reparameterize from (r_0, r_1, γ) to $(r_0, \tilde{r}_1, \gamma)$, where $\tilde{r_1} = Pr(X=1|S=1,Y=1) = \text{expit}(\text{logit}(r_1)+\gamma)$. By change-of-variables we get to the prior %

$$\pi(r_0, \tilde{r}_1, \gamma) \quad \propto \quad I_A(\gamma) I_{(0,1)}(r_0) I_{(0,1)}(\tilde{r}_1) g(\text{logit}\tilde{r}_1 - \gamma)/g(\text{logit}\tilde{r}_1)$$

where $g()$ is the standard logistic density.

This gives the limiting posterior distribution on $\psi = \text{logit}\tilde{r}_1 - \gamma - \text{logit}\tilde{r}_0$ as the logistic distribution with location $-\text{logit}(r_0^\dagger)$, truncated to the identification region.

For Study 2, we can do probabilistic bias analysis, but then, out of abundance of caution, use importance sampling to nudge (if needed) this to be the fully Bayesian posterior distribution:

The following function can be applied for either finite or infinite sample size.

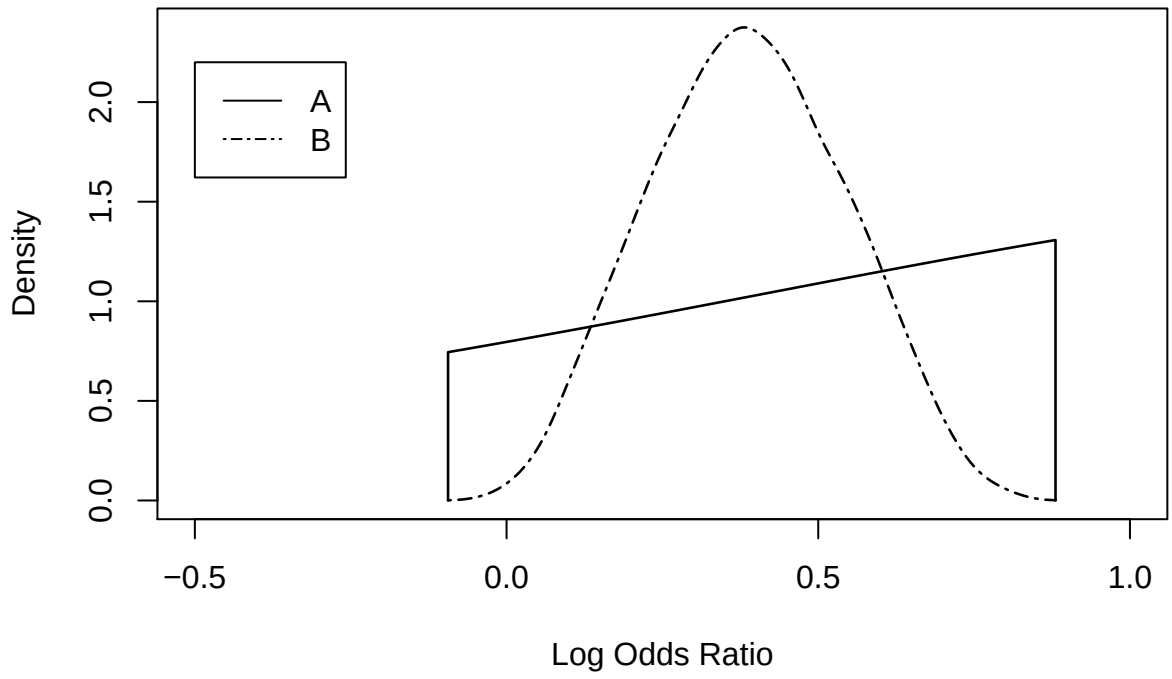
```
posterior.2 <- function(y=NA, xstr=NA, r.app.tr=NA, sn.lwr, sp.lwr, m=40000) {

  ## supply r.app.tr for large-sample limit, or
  ## supply y,xstr for actual dataset
```

```
  ### prior draws from sn, sp
  sn.drw <- t(replicate(m,runif(2, sn.lwr, rep(1,2))))
  sp.drw <- t(replicate(m,runif(2, sp.lwr, rep(1,2))))

  ### posterior draws, or limiting vals, of r.app
  if (is.na(r.app.tr[1])) {
    r.app.drw <- cbind(
      rbeta(m, 1 + sum((y==0)&(xstr==1)), 1 + sum((y==0)&(xstr==0))),
      rbeta(m, 1 + sum((y==1)&(xstr==1)), 1 + sum((y==1)&(xstr==0)))
    )
  } else {
    r.app.drw <-t(matrix(r.app.tr,2,m))
  }

  ### induces sample on actual prevalences
  r.drw <- cbind(
    (r.app.drw[,1] + sp.drw[,1] - 1)/(sn.drw[,1]+sp.drw[,1]-1),
    (r.app.drw[,2] + sp.drw[,2] - 1)/(sn.drw[,2]+sp.drw[,2]-1)
  )

  ### remove any out-of-bounds draws
  ndx <- (apply(r.drw,1,min)>0) & (apply(r.drw,1,max)<1)
  m.new <- sum(ndx)
  r.drw <- r.drw[ndx,]; sn.drw <- sn.drw[ndx,]; sp.drw <- sp.drw[ndx,]

  ### resample with importance weights to make fully Bayes
  ### weights based on Jacobian of mapping between r and r.app
  wht <- 1/((sn.drw[,1]+sp.drw[,1]-1)*(sn.drw[,2]+sp.drw[,2]-1))
  wht <- wht/sum(wht)
  ndx <- sample(1:m.new, prob=wht, replace=T)

  trg.drw <- logit(r.drw[ndx,2])-logit(r.drw[ndx,1])

  ### return posterior sample of target, two indicators of numerical precision
  list(trg.drw=trg.drw, frac.oob=1-m.new/m, ess=1/sum(wht^2))
}
```

So the limiting posterior distributions for ψ look as follows:

```
if (MYPLOT) {
  pdf.PG("Fig1.pdf",1,1)
}

### the B answer via Monte Carlo
pst <- posterior.2(r.app.tr=r.app.2, sn.lwr=sn.lwr, sp.lwr=sp.lwr)
tmp <- density(pst$trg.drw, from=bnd.2[1], to=bnd.2[2], adjust=1.5,n=512)
plot(tmp$x, tmp$y, type="l",lwd=1.3,lty=6, xlim=c(-0.5,1),
     xlab="Log Odds Ratio", ylab="Density")

### and superimpose (the closed-form) A answer
gr <- seq(from=bnd.1[1],to=bnd.1[2],length=500)
loc <- -logit(r.tr[1])
points(
  c(bnd.1[1],gr,bnd.1[2]),
```

```
  c(0,dlogis(gr,loc)/(plogis(bnd.1[2],loc)-plogis(bnd.1[1],loc)),0),
  lwd=1.3,type="l"
)

legend(-0.5, 2.2, legend=c("A","B"),lty=c(1,6))
```




```
if (MYPLOT) {
  graphics.off()
}
```

Quick due diligence on numerical computation for the B case:

```
c(pst$frac.oob, pst$ess)
```

```
## [1]     0.00 39902.55
```

How much limiting posterior probability to left of null for both studies:

```
c((plogis(0,loc) - plogis(bnd.1[1],loc))/
  (plogis(bnd.1[2],loc) - plogis(bnd.1[1],loc)),
  mean(pst$trg.drw<0))
```

```
## [1] 0.07227119 0.00180000
```

What are the 95% equal-tailed credible intervals, and what proportion of the ID interval do they occupy?

```
cred.1.95 <- c(
qlogis(plogis(bnd.1[1],loc)+0.025*(plogis(bnd.1[2],loc)-plogis(bnd.1[1],loc)),loc),
qlogis(plogis(bnd.1[1],loc)+0.975*(plogis(bnd.1[2],loc)-plogis(bnd.1[1],loc)),loc)
```

```
)

cred.2.95 <- quantile(pst$trg.drw, c(0.025, 0.975))

cred.1.95
```

```
## [1] -0.06066359  0.86141586
```

```
exp(cred.1.95)
```

```
## [1] 0.9411398 2.3665090
```

```
sum(c(-1,1)*cred.1.95)/sum(c(-1,1)*bnd.1)
```

```
## [1] 0.9462572
```

```
cred.2.95
```

```
##       2.5%      97.5%
## 0.09320451 0.68672926
```

```
exp(cred.2.95)
```

```
##     2.5%    97.5%
## 1.097686 1.987205
```

```
sum(c(-1,1)*cred.2.95)/sum(c(-1,1)*bnd.2)
```

```
## [1] 0.6090875
```

Now on to repeated sampling with finite sample size

The function above will handle Lab 2, but need a function for Team 1:

```
posterior.1 <- function(y, x, gamma.lwr, gamma.upr, m=40000) {

  r0.drw <- rbeta(m, 1+sum((y==0)&(x==1)), 1 + sum((y==0)&(x==0)))
  r1.tld.drw <- rbeta(m, 1+sum((y==1)&(x==1)), 1 + sum((y==1)&(x==0)))
  gamma.drw <- runif(m, gamma.lwr, gamma.upr)

  ### importance weights
  wht <- dlogis(logit(r1.tld.drw)-gamma.drw)/dlogis(logit(r1.tld.drw))
  wht <- wht/sum(wht)
  ndx <- sample(1:m, prob=wht, replace=T)

  trg.drw <- logit(r1.tld.drw[ndx]) - logit(r0.drw[ndx]) - gamma.drw[ndx]

  ### return posterior sample, indicator of numeric precision
  list(trg.drw=trg.drw, ess=1/sum(wht^2))
}
```

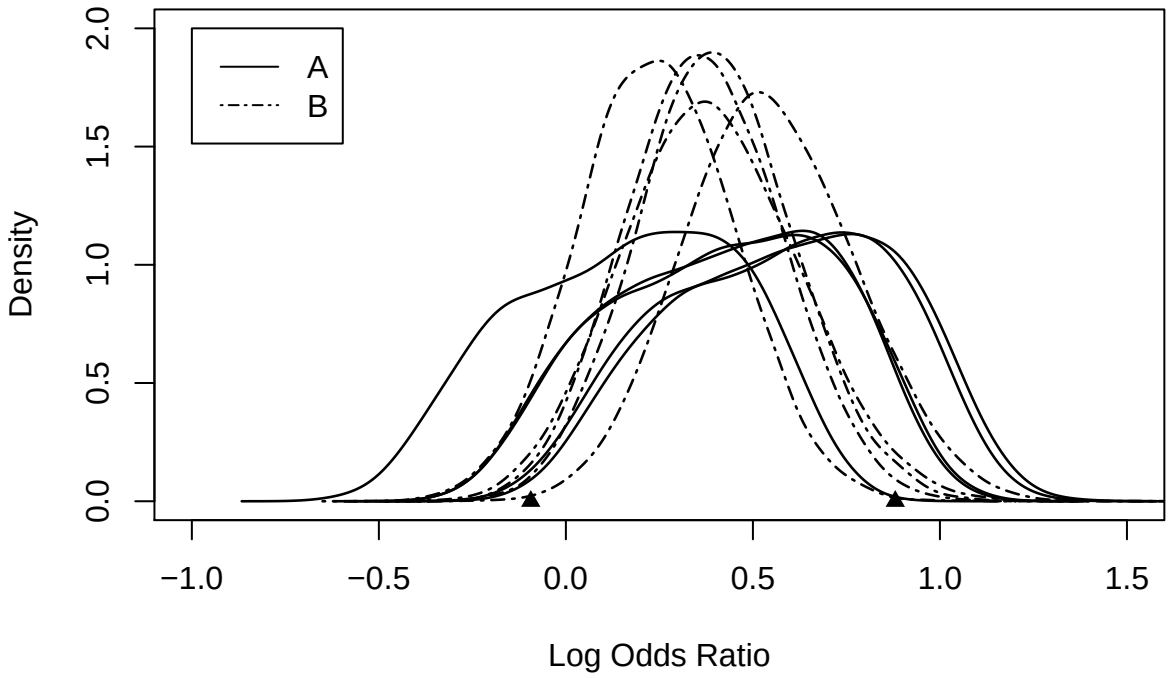
Now can draw the samples, compute and store the posteriors

```
NREP <- 5  ### number of repeated samples
n <- 2000  ### size of each sample
ans.1 <- ans.2 <- vector(mode="list", length=NREP)

### not really necessary, but will make the sample participants
### match (A versus B) as much as possible
### can't be a perfect match due to selection bias
```

```
for (i in 1:NREP) {
  y.1 <- y.2 <- c(rep(0,n/2),rep(1,n/2))    ### balanced case-control studies

  ### study B, actual and measured exposure
  x.2 <- rbinom(n, size=1, prob=(1-y.2)*r.tr[1]+y.2*r.tr[2])
  xstr.2 <- rbinom(n,size=1,
                   prob=(1-x.2)*((1-y.2)*(1-sp.tr[1]) + y.2*(1-sp.tr[2])) +
                            x.2*((1-y.2)*sn.tr[1] + y.2*sn.tr[2]))

  ### for study A, can have the same controls
  x.1 <- rep(NA, n)
  x.1[y.1==0] <- x.2[y.1==0]

  ### but for cases, will resample as per the selection bias
  x.1[y.1==1] <- sample(x.2[y.1==1], prob=exp(x.2[y.1==1]*gamma.tr), replace=T)

  ans.1[[i]] <- posterior.1(y.1, x.1, gamma.lwr, gamma.upr)
  ans.2[[i]] <- posterior.2(y=y.2, xstr=xstr.2, sn.lwr=sn.lwr, sp.lwr=sp.lwr)
}
```

Plot all the posteriors:

```
if (MYPLOT) {
  pdf.PG("fig2.pdf",1,1)
}

for (i in 1:NREP) {
  tmp <- density(ans.2[[i]]$trg.drw, adjust=1.7, n=512)
  if (i==1) {
    plot(tmp$x, tmp$y, type="l",xlim=c(-1,1.5),ylim=c(0,2),
         xlab="Log Odds Ratio", ylab="Density",lwd=1.3, lty=6)
  } else {
    points(tmp$x, tmp$y, type="l", lwd=1.3, lty=6)
  }
  tmp <- density(ans.1[[i]]$trg.drw, adjust=1.7,n=512)
  points(tmp$x, tmp$y, type="l", lwd=1.3)
}
points(bnd.1, rep(0,2), pch=17)

legend(-1, 2, legend=c("A","B"),lty=c(1,6))
```




```
if (MYPLOT) {
  graphics.off()
}
```

Then how much posterior weight to the left of the null

```
for (i in 1:NREP) {
  print(c(mean(ans.1[[i]]$trg.drw<0), mean(ans.2[[i]]$trg.drw<0)))
}
```

```
## [1] 0.09745 0.02370
## [1] 0.0943 0.0334
## [1] 0.017625 0.042800
## [1] 0.312575 0.116050
## [1] 0.026050 0.005075
```

And widths of 95% equal-tailed credible intervals

```
for (i in 1:NREP) {
  print(c(sum(c(-1,1)*quantile(ans.1[[i]]$trg.drw,c(0.025,.975))),
          sum(c(-1,1)*quantile(ans.2[[i]]$trg.drw,c(0.025,.975)))))
}
```

```
## [1] 1.0657529 0.8161476
## [1] 1.079483 0.798501
## [1] 1.0820631 0.9048237
## [1] 1.0746190 0.8029921
## [1] 1.0833184 0.8833762
```