

Parameter restrictions for the sake of identification: Example 1

Paul Gustafson

September 30, 2022

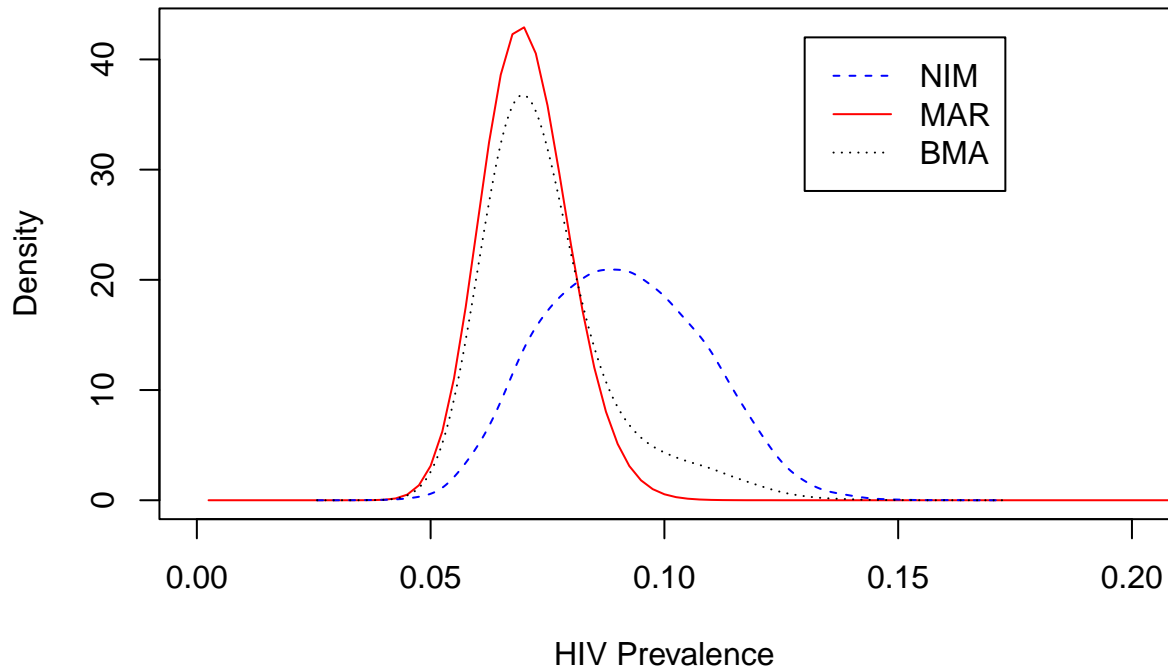
```
### Bayes factor in favor of MAR
bysfct <- function(c0, c10, c11) {
  n <- c0+c10+c11
  (n+2)*(n+3)/(6*(c0+1)*(n-c0+1))
}

### HIV data
c0 <- 36; c10 <- 699; c11 <- 52

### plot posterior under MAR
gr <- (1:399)/400
plot(gr, dbeta(gr, 1+c11, 1+c10), type="l", xlim=c(0,0.2), col="red",
      xlab="HIV Prevalence", ylab="Density")

## MC representation of NIM posterior
m <- 10000
tmp <- rdirch(m, c(2+c0, 1+c10, 1+c11))
psi.mc <- runif(m)*tmp[,1]+tmp[,3]
require(KernSmooth)
kfit <- bkde(psi.mc)
points(kfit$x, kfit$y, type="l", col="blue", lty=2)

## Model-averaged posterior
bf <- bysfct(c0,c10, c11)
tmp <- (1/(1+bf))*kfit$y +
  (bf/(1+bf))*dbeta(kfit$x, 1+c11, 1+c10)
points(kfit$x, tmp, type="l", col="black", lty=3)
legend(.13,42, legend=c("NIM","MAR","BMA"),
      lty=c(2,1,3), col=c("blue","red","black"))
```



Draw ensemble of parameter values:

```
NREP <- 100000

### cell probs, order (R,Y) = (0,0), (0,1), (1,0), (1,1)

### top-half rows generated under M0
res <- cbind(0, rdirch(NREP,rep(1,4)))

### bottom-half rows generated under M1
tmp.q <- runif(NREP)
tmp.p <- runif(NREP)
res <- rbind(res,
  cbind(1, (1-tmp.q)*(1-tmp.p), (1-tmp.q)*tmp.p, tmp.q*(1-tmp.p), tmp.q*tmp.p))
```

Compute all the limiting quantities:

```
trg <- res[,3]+res[,5]
limbf <- 1/(6*(res[,2]+res[,3])*(res[,4]+res[,5]))
limpm0 <- res[,5] + .5*(res[,2]+res[,3])
limpm1 <- res[,5]/(res[,4]+res[,5])
limmx <- (1/(1+limbf))*limpm0 + (limbf/(1+limbf))*limpm1
```

Summarize the results:

```
amse <- matrix(NA,2,2)
ndx <- res[,1]==0 ### nature = M0
#amse[1,1] <- mean((limpm0[ndx]-trg[ndx])^2)
```

```

amse[1,2] <- mean((limmx[ndx]-trg[ndx])^2)

ndx <- 1:(2*NREP) ### nature = MIX
#amse[2,1] <- mean((limpm0[ndx]-trg[ndx])^2)
amse[2,2] <- mean((limmx[ndx]-trg[ndx])^2)

### but actually have closed-forms for some elements
amse[1,1] <- 1/40
amse[2,1] <- 1/80+1/72

### output table
round(sqrt(amse),digits=3)

##      [,1] [,2]
## [1,] 0.158 0.177
## [2,] 0.162 0.136

### percentage changes
sqrt(amse[1,2]/amse[1,1])

## [1] 1.117635

sqrt(amse[2,2]/amse[2,1])

## [1] 0.8398169

### bootstrap to check stability of percentage change
bs.rep <- 200
r1.bs <- r2.bs <- rep(NA,bs.rep)
for (i in 1:bs.rep) {

  smp.bs <- sample(1:NREP, size=NREP, replace=T)
  tmp.d <- 1/40
  tmp.n <- ((limmx-trg)^2)[smp.bs]
  r1.bs[i] <- sqrt(mean(tmp.n)/mean(tmp.d))

  smp.bs <- sample(1:(2*NREP), size=2*NREP, replace=T)
  tmp.d <- 1/80+1/72
  tmp.n <- ((limmx-trg)^2)[smp.bs]
  r2.bs[i] <- sqrt(mean(tmp.n)/mean(tmp.d))
}

summary(r1.bs)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.107  1.116   1.118   1.118   1.119   1.125

sqrt(var(r1.bs))

## [1] 0.002837334

summary(r2.bs)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      0.8350  0.8385   0.8397   0.8399   0.8413   0.8448

sqrt(var(r2.bs))

```

```
## [1] 0.001994538
```