KU Leuven Summer School
Segment 3B
More Misclassification Models

Paul Gustafson

September 15, 2022

Start with a mystery dataset

Case-control again, but:

- Nobody has a measurement of true exposure X
- Everbody has a pair of measurements from two **different** surrogates (for X), X_1^*, X_2^*

E.g., think $X_1^* \sim$ self-report, $X_2^* \sim$ imperfect lab test

```
head(dta)
```

```
##   xstr1 xstr2 y
## 1     0     0 0
## 2     0     0 0
## 3     0     0 1
## 4     0     0 0
## 5     1     0 1
## 6     0     0 1
```

but still want to infer the (x, y) association

↓ n = 5000

Mystery dataset, continued

```
table(dta)
```

```
## , , y = 0
##
##      xstr2
## xstr1    0    1
##     0 1839  134
##     1  404  123
##
## , , y = 1
##
##      xstr2
## xstr1    0    1
##     0 1712  142
##     1  450  196
```

lots of agreement between x_1^* and x_2^*

Some simple analyses

```
summary(glm(y~xstr1, family=binomial))$coef
```

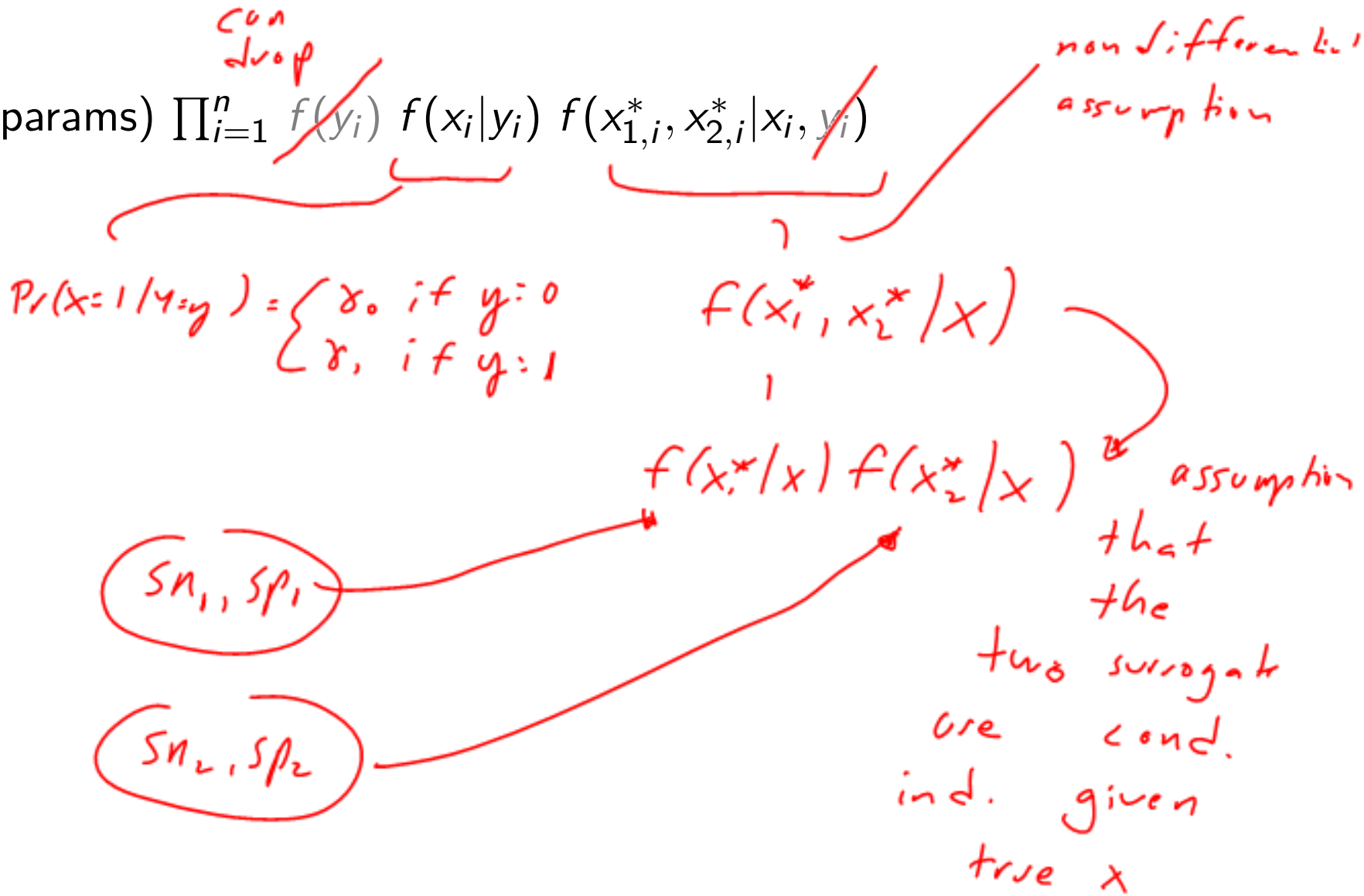
```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.0622     0.0323   -1.92 5.44e-02
## xstr1         0.2658     0.0670    3.97 7.31e-05
```

```
summary(glm(y~xstr2, family=binomial))$coef
```

```
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.0368     0.0301   -1.22 0.222329
## xstr2         0.3108     0.0881    3.53 0.000419
```

Generative model

$$f(\text{params}) \prod_{i=1}^{n} f(y_i)\, f(x_i|y_i)\, f(x^*_{1,i}, x^*_{2,i}|x_i, y_i)$$



This will run (but ess/wall-time unpleasant)

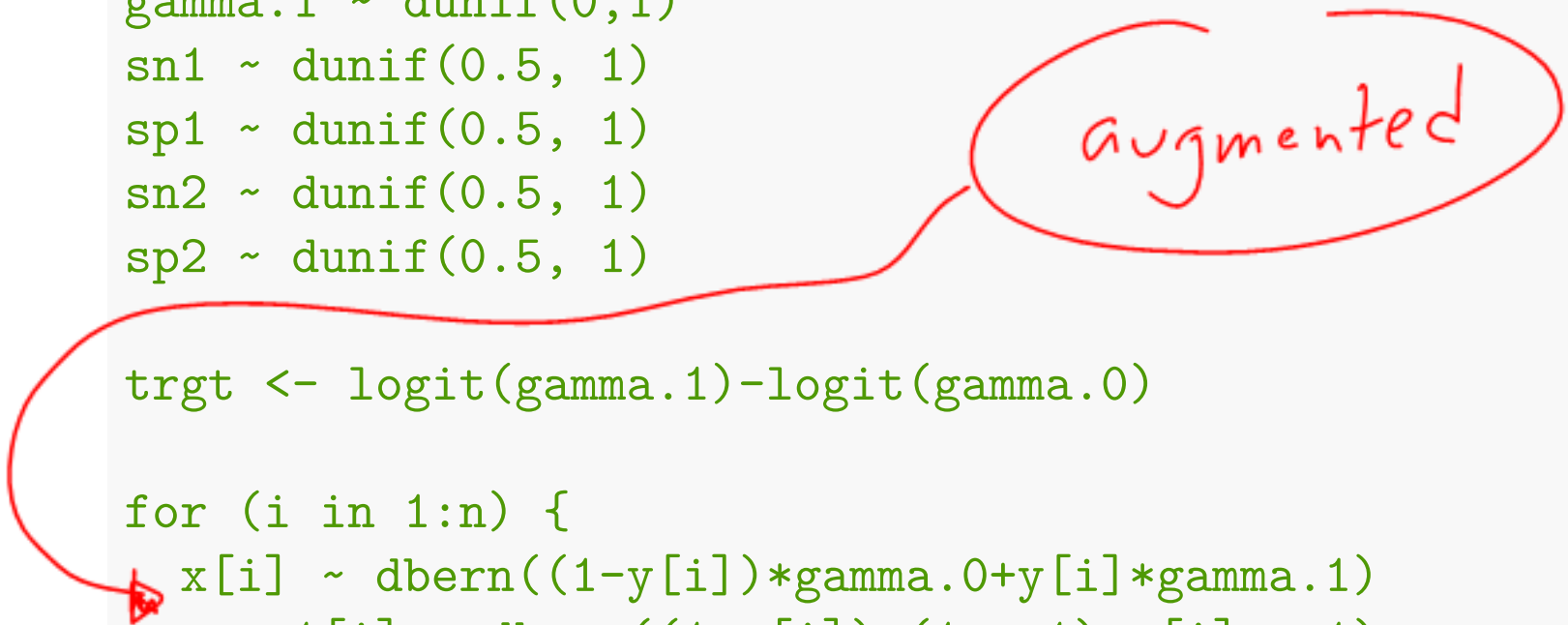
```
genmod.string <- "model{

### prior distribution
gamma.0 ~ dunif(0,1)
gamma.1 ~ dunif(0,1)
sn1 ~ dunif(0.5, 1)
sp1 ~ dunif(0.5, 1)
sn2 ~ dunif(0.5, 1)
sp2 ~ dunif(0.5, 1)

trgt <- logit(gamma.1)-logit(gamma.0)

for (i in 1:n) {
  x[i] ~ dbern((1-y[i])*gamma.0+y[i]*gamma.1)
  xstr1[i] ~ dbern((1-x[i])*(1-sp1)+x[i]*sn1)
  xstr2[i] ~ dbern((1-x[i])*(1-sp2)+x[i]*sn2)
}

}"
```



Instead consider

```
genmod.string <- "model {
gamma.0 ~ dunif(0,1); gamma.1 ~ dunif(0,1)
trg <- logit(gamma.1)-logit(gamma.0)
sn1 ~ dunif(0.5,1); sp1 ~ dunif(0.5,1)
sn2 ~ dunif(0.5,1); sp2 ~ dunif(0.5,1)

### controls: dist(xstr1, xstr2 |y=0)
q.0[1] <- (1-gamma.0)*sp1*sp2 + gamma.0*(1-sn1)*(1-sn2)
q.0[2] <- (1-gamma.0)*(1-sp1)*sp2 + gamma.0*(sn1)*(1-sn2)
q.0[3] <- (1-gamma.0)*sp1*(1-sp2) + gamma.0*(1-sn1)*sn2
q.0[4] <- (1-gamma.0)*(1-sp1)*(1-sp2) + gamma.0*sn1*sn2
dat.0 ~ dmulti(q.0[], n.0)

### cases: dist(xstr1, xstr2 |y=1)
q.1[1] <- (1-gamma.1)*sp1*sp2 + gamma.1*(1-sn1)*(1-sn2)
q.1[2] <- (1-gamma.1)*(1-sp1)*sp2 + gamma.1*(sn1)*(1-sn2)
q.1[3] <- (1-gamma.1)*sp1*(1-sp2) + gamma.1*(1-sn1)*sn2
q.1[4] <- (1-gamma.1)*(1-sp1)*(1-sp2) + gamma.1*sn1*sn2
dat.1 ~ dmulti(q.1[], n.1)
}"
```

Pause: what's going on here?

collapsed version

$f(\text{params})\, f(y)\, f(x_1^*, x_2^* \mid y, \text{params})$

e.g. 2×2 of cell counts for cases ($y=1$) say arises as multinomial

	$x_2^*=0$	$x_2^*=1$
$x_1^*=0$	q_{00}	q_{01}
$x_1^*=1$	q_{10}	q_{11}

to get these - same algebra as used in Segment 3A

Pause, continued

Turn the crank

```
### generative model, data go in
mod <- jags.model(textConnection(genmod.string),
  data=list(dat.0=as.vector(table(dta)[,,1]),
            n.0=sum(table(dta)[,,1]),
            dat.1=as.vector(table(dta)[,,2]),
            n.1=sum(table(dta)[,,2])),
  n.chains=3)

update(mod,2000) #burn-in

### MCMC output comes out
opt.JAGS <- coda.samples(mod, n.iter=60000, thin=10,
  variable.names=c("gamma.0","gamma.1","sn1","sp1",
                   "sn2","sp2","trg"))
```

$\text{logit}(\gamma_1) - \text{logit}(\gamma_0)$

Inference

```
MCMCsummary(opt.JAGS)
```

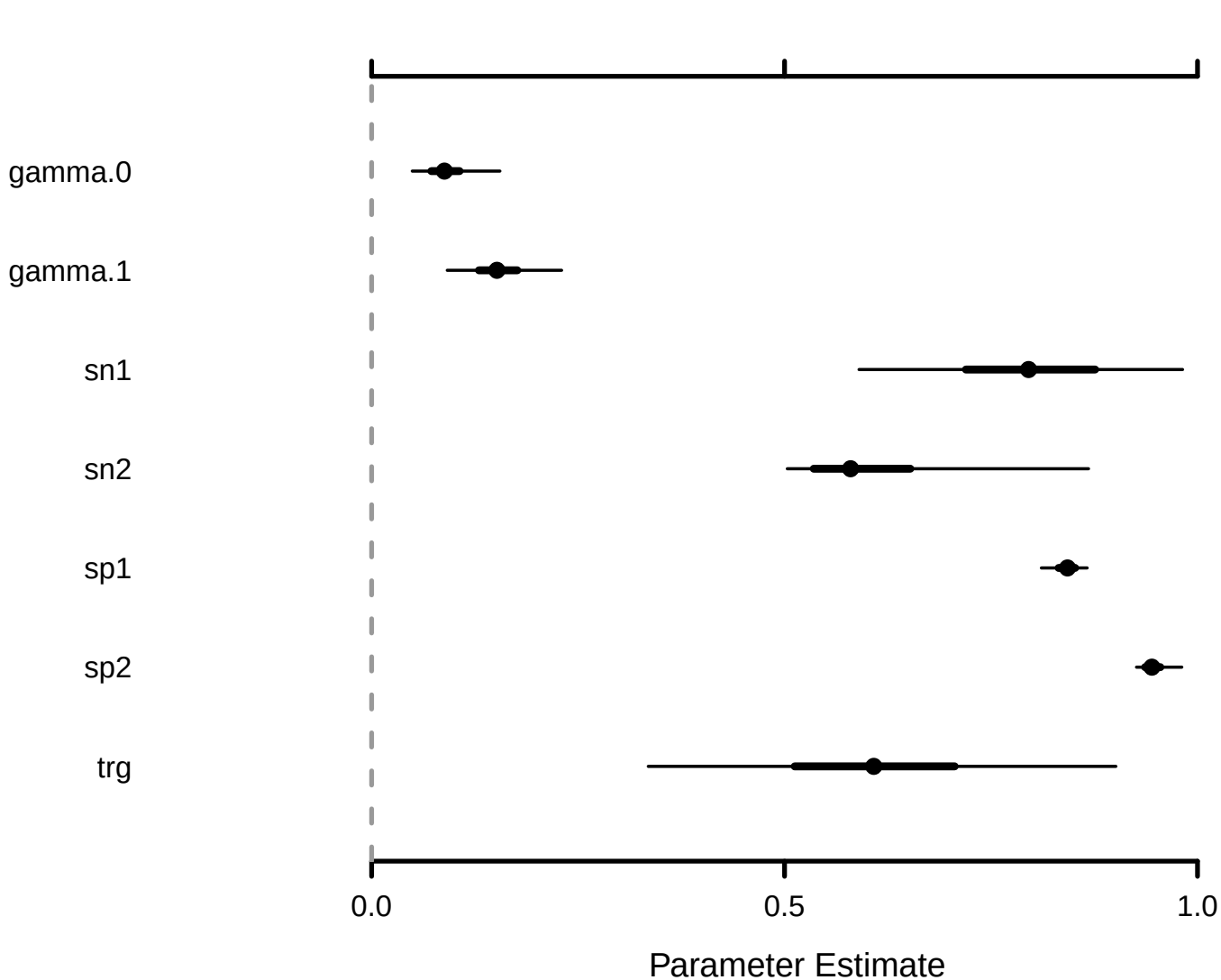
```
##              mean     sd   2.5%    50% 97.5% Rhat n.eff
## gamma.0 0.0915 0.0268 0.0492 0.0881 0.155    1  2588
## gamma.1 0.1543 0.0349 0.0915 0.1517 0.230    1  2588
## sn1     0.7952 0.1051 0.5902 0.7955 0.982    1  3190
## sn2     0.6082 0.0965 0.5033 0.5800 0.868    1  3791
## sp1     0.8414 0.0144 0.8109 0.8426 0.866    1  4418
## sp2     0.9470 0.0141 0.9261 0.9448 0.981    1  2957
## trg     0.6104 0.1440 0.3350 0.6080 0.901    1  9782
```

log OR(X,Y)

inference for (X,Y) association, without any X data!

Inference, continued

```
MCMCplot(opt.JAGS)
```



inferences compatible with truth

And a grand reveal

Mystery dataset generated as follows

$\gamma_0 = 0.1$

$\gamma_1 = \text{expit}\{\text{logit } 0.1 + \log 2\}$

$Sn_1 = .75 \quad Sp_1 = .85 \quad$ (self-report)

$Sn_2 = .60, \; Sp_2 = 0.95 \quad$ (lab test)

Pause to marvel for a moment: Asked a lot of these data, and they delivered

inference "worked" without any x data and without any obvious info about the quality of each surrogate

But parameter counting hints at why this might be possible

- $X_1^*, X_2^* | Y$ inherently a 6 df object
- we have 6 unknown parameters

An aside on parameter-counting (1 of 2)

params $\leq$ data df is necessary, but not sufficient, for identification

Need to prove that the mapping from parameters to $(x_1^*, x_2^* / y)$ cell probs is invertible

Relevant ref: Hui & Walter (1980, Biom.)

An aside on parameter-counting (2 of 2)

What if Y had 3 levels instead of 2. Data df = 9.

params: 3 for X/Y

6 for $X_1^*, X_2^*/X$, ~~Y~~

keep non-D assumption, relax cond ind. assumption

BUT param. → cell prob map NOT invertible

Relevant ref: Johnson & Hanson (2005, Stat. Sci., comment)

Pause some more: Lunch is never *completely* free

- $(X_1^*, X_2^* \perp\!\!\!\perp Y \mid X)$
- $(X_1^* \perp\!\!\!\perp X_2^* \mid X)$

both strong and empirically

unchecklable assumptions

And yet another sense in which lunch isn't free

even with those assumptions

inverting that map breaks down as

$\gamma_1 - \gamma_0 \to 0$

limit of $X \perp\!\!\!\perp Y$