Methods of Applied Stats , More INLA

Patrick Brown, University of Toronto and St Mike's

Sept to Dec 2020

Example

```
> data("bacteria", package = "MASS")
> bacteria$infected = as.numeric(bacteria$y == "y")
> bacteria$weekFac = factor(bacteria$week)
> bacteria$trt = factor(bacteria$ap, levels = c("p", "a"),
+   labels = c("placebo", "active treatment"))
> bacteria[c(1, 7:10), c("ID", "weekFac", "trt", "infected")]
    ID weekFac              trt infected
1  X01       0          placebo        1
7  X02       6 active treatment        0
8  X02      11 active treatment        1
9  X03       0 active treatment        1
10 X03       2 active treatment        1
```

Bacteria data with an individual-level random effect

$$\begin{aligned} Y_{ij} &\sim \text{Bern}(\rho_{\overset{\text{person}}{i}\underset{\text{time}}{j}}) \\ \text{logit}(\rho_{ij}) &= X_{ij}\beta + U_i \\ U_i &\sim \text{N}(0, \sigma^2) \end{aligned}$$

```
> library(INLA)
> res = inla(infected ~ weekFac + trt +
+   f(ID, model='iid', prior = 'pc.prec', param = c(2, 0.5)),
+   control.fixed = list(
+     mean = 0, mean.intercept = 0,
+     prec = 100^(-2), prec.intercept = 100^(-2)),
+   family = 'binomial', data=bacteria,
+   control.inla = list(strategy='laplace', fast=FALSE))
```

$X_{ij}\beta$

$U_i \sim N(0, \sigma^2)$, σ has a (exponential) prior median of two.

$\beta_0 \sim N(0, 100^2)$

$\begin{pmatrix} \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} \sim N(\vec{0}, 100^2 I)$

① βs are usually insensitive to the priors

② Unlike σ, β can be negative.

③ Flat priors (e.g., $N(0, 100^2 I)$) makes sense.

④ Always use default prior for the fixed effect.

```
> res$summary.fixed[, c(4, 3, 5)]

                      0.5quant 0.025quant   0.975quant
(Intercept)          3.7763208   2.327576  5.725563272
weekFac2             0.1643283  -1.255522  1.636596884
weekFac4            -1.5439286  -2.916394 -0.317738083
weekFac6            -1.6769759  -3.075238 -0.435195431
weekFac11           -1.6882410  -3.069860 -0.470155061
trtactive treatment -1.1593444  -2.549680 -0.007278442
```

Natural Scale

$$
\begin{aligned}
Y_{ij} \sim & \mathsf{Bern}[\mathsf{odds}_{ij}/(1+\mathsf{odds}_{ij})] \\
\mathsf{odds}_{ij} = & \prod_p \exp(\beta_p)^{X_{ijp}} \exp(U_i) \\
U_i \sim & \mathsf{N}(0, \sigma^2)
\end{aligned}
$$

odds ratio

- $\exp(\beta_p) = \mathsf{odds}_{ij}/\mathsf{odds}_{k\ell}$
- if $X_{ijp} = X_{k\ell p} + 1$
- and $X_{ijq} = X_{k\ell q}, q \neq p$
- and $U_i = U_k$

```
> resTable = rbind(
+   'Baseline prob' = 1/(1+exp(-res$summary.fixed[1,c(4,3,5)])),
+   exp(res$summary.fixed[-1,c(4,3,5)]),
+   '$\\sigma$'=Pmisc::priorPost(res)$summary[,c(4,3,5)])
> rownames(resTable) = gsub("Fac|trt", " ", rownames(resTable))
> knitr::kable(resTable, digits=2, caption ='Posterior medians and quantil
```

Table 1: Posterior medians and quantiles for the baseline probability, odds ratios, and standard deviation of the random effect.

(Estimation)

odds ratio

	0.5quant	0.025quant	0.975quant
Baseline prob	0.98	0.91	1.00
week 2	1.18	0.28	5.14
week 4	0.21	0.05	0.73
week 6	0.19	0.05	0.65
week 11	0.18	0.05	0.62
active treatment	0.31	0.08	0.99
σ	1.40	0.58	2.45

- Looks like the active treatment is effective
- How do we interpret σ?

Natural scale for σ?

- $\exp(\sigma) = \text{odds}_{ij}/\text{odds}_{k\ell}$
- if $U_i = U_k + \sigma$, $X_{ijq} = X_{k\ell q}$
- i.e. $U_k = 0$
    - k is typical
- and $U_i = \sigma$
    - i is 1 SD more sickly than is typical

$$
\begin{aligned}
Y_{ij} &\sim \text{Bern}[\text{odds}_{ij}/(1+\text{odds}_{ij})] \\
\text{odds}_{ij} &= \prod_p \exp(\beta_p)^{X_{ijp}} \exp(U_i) \\
U_i &\sim \text{N}(0, \sigma^2)
\end{aligned}
$$

- $\sigma = 1.4$, $\exp(\sigma) = 4.1$
- that's very big!
- Should we report $\exp(\sigma)$ instead of σ?
- nobody ever does, so it would be confusing.
- insisting on σ instead of σ^2 or $1/\sigma^2$ is as much as I'll challenge the standard practice for now.

Interperting σ

- suppose $U_k = 0$ and $U_i = \sigma$
- k is in the 50th percentile of sickliness (Typical)
- i is the $100 * \Phi(1)$ = `100*pnorm(1)` = 84th percentile

Inter-Quartile Range

- k is the 25th percentile, i is the 75th.
- $U_i = \Phi^{-1}(0.75)\sigma$ = `qnorm(0.75)` $\sigma \approx 0.67\sigma$
- $U_k = \Phi^{-1}(0.25)\sigma \approx -0.67\sigma$
- $\text{odds}_{ij}/\text{odds}_{k\ell} = \exp\{[\Phi^{-1}(0.75) - \Phi^{-1}(0.25)]\sigma\} \approx \exp(1.3\sigma)$ $\frac{\exp(U_i)}{\exp(U_k)} = \frac{\exp(0.67\sigma)}{\exp(-0.67\sigma)}$
- IQR $= \exp(1.3 \cdot 1.4) = 6.6$
- huge! $\text{IQR} \cdot \hat{\sigma}$

Person in the 75th percentile of sickness has 6.6 times the odds of being sick as the person in the 25th.

In reverse

- Suppose we don't know the results and we're setting a prior for σ
- Is 0.5 a sensible prior median for σ?
- The IQR for individual-level risk would be $\exp(1.3 \cdot 0.5) = 1.9$
- that's fairly big.
- how about $\sigma = 4$? $\exp(1.3 \cdot 4) = 180$
- how about $\sigma = 0.2$? $\exp(1.3 \cdot 0.2) = 1.3$
- $\sigma = 0.2$ is about right.
- Suppose I want my prior median for σ to correspond to an IQR of 1.5
- ... $\exp(1.3 \cdot \sigma) = 1.5$, or $\sigma = \log(1.5)/1.3 \approx 0.31$
- ... `prior = 'pc.prec', param = c(log(1.5)/1.3, 0.5)`
- or I want 1 sd to double the odds of infection
- ... so $\sigma = \log(2) \approx 0.69$
- ... `prior = 'pc.prec', param = c(log(2), 0.5)`

In summary

- we must set priors for σ
- unless you have a reason to do otherwise, use an Exponential prior (or `pc.prec`)
- set the prior median, unless you have some reason to set the mean or rate or 95% quantile.
- posteriors for σ tend to fairly insensitive to the hyperparameter of the Exponential.
- but you do need to justify the hyperparameter
- set $pr(\sigma < \log(2)) = 0.5$, prior median means 1 SD doubles odds Sensible.
- or $pr(\sigma < \log(1.5)/1.3) = 0.5$, the odds ratio of the IQR is a 50% increase
- or replace the 0.25 and 0.75 above with some other numbers (0.025 and 0.975?).

Predicted values

Let's fit a treatment by time interaction model

```
> resGlm = glm(infected ~ weekFac * trt, family = binomial(link = "logit")
+   data = bacteria)
> summary(resGlm)$coef
```

```
                                 Estimate Std. Error     z value    Pr(>|z
(Intercept)                    2.25129180  0.7433919  3.02840495 0.0024584
weekFac2                       0.69314717  1.2669470  0.54710037 0.5843097
weekFac4                      -0.99852883  0.9349118 -1.06804599 0.2854997
weekFac6                       0.52129692  1.2708682  0.41018961 0.6816668
weekFac11                     -0.86499744  0.9301245 -0.92998029 0.3523812
trtactive treatment           -0.09180755  0.9614710 -0.09548655 0.9239283
weekFac2:trtactive treatment  -0.90672127  1.5355461 -0.59048781 0.5548636
weekFac4:trtactive treatment  -0.27365222  1.2031358 -0.22744916 0.8200744
weekFac6:trtactive treatment  -2.41841691  1.4709942 -1.64406967 0.1001618
weekFac11:trtactive treatment -0.60133963  1.1934934 -0.50384832 0.6143679
```

Another model

```
> resGlm2 = glm(infected ~ weekFac:trt, family = binomial(link = "logit"),
+   data = bacteria)
> summary(resGlm2)$coef

                               Estimate Std. Error    z value   Pr(>|z|)
(Intercept)                   0.6931472  0.4330127  1.6007548 0.10943123
weekFac0:trtplacebo           1.5581446  0.8603090  1.8111453 0.07011836
weekFac2:trtplacebo           2.2512918  1.1135633  2.0217008 0.04320727
weekFac4:trtplacebo           0.5596158  0.7133923  0.7844433 0.43278006
weekFac6:trtplacebo           2.0794415  1.1180225  1.8599281 0.06289569
weekFac11:trtplacebo          0.6931472  0.7071068  0.9802581 0.32695871
weekFac0:trtactive treatment  1.4663371  0.7478602  1.9607101 0.04991284
weekFac2:trtactive treatment  1.2527630  0.7539578  1.6615823 0.09659655
weekFac4:trtactive treatment  0.1941560  0.6238435  0.3112255 0.75562918
weekFac6:trtactive treatment -0.4307829  0.6036746 -0.7136011 0.47547385
```

the two models are the same, coefficients of one are linear combinations of the other

Preditced

```
> newx = expand.grid(
+   trt = levels(bacteria$trt),
+   weekFac = levels(bacteria$weekFac))
> newx
                trt weekFac
1           placebo       0
2  active treatment       0
3           placebo       2
4  active treatment       2
5           placebo       4
6  active treatment       4
7           placebo       6
8  active treatment       6
9           placebo      11
10 active treatment      11
```

- suppose I want probability of being infected
- for each treatment and each week
- create a data frame `newx` to make predictions from

```
> myPred = as.data.frame(predict(
+   resGlm, newx, se.fit=TRUE))
> myPred[1:3,]
       fit    se.fit residual.scale
1 2.251292 0.7433919              1
2 2.159484 0.6097498              1
3 2.944439 1.0259255              1
```

$X\beta \to$ log odds.

Probabilities

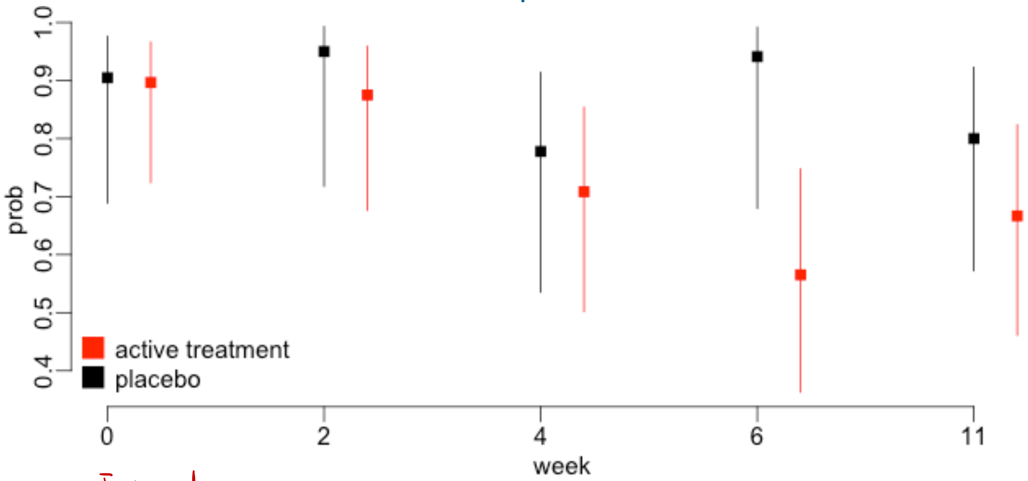
convert to confidence intervals

```
> theCiMat = Pmisc::ciMat(0.95)
> theCiMat
           est       2.5     97.5
Estimate     1  1.000000 1.000000
Std. Error   0 -1.959964 1.959964
> myPredCI =
+   as.matrix(myPred[,1:2]) %*%
+   theCiMat
> myPredCI[1:3,] 95% CI for log odds
       est       2.5     97.5
1 2.251292 0.7942704 3.708313
2 2.159484 0.9643965 3.354572
3 2.944439 0.9336619 4.955216
```

convert to probabilities

```
> myPredOdds = exp(myPredCI)
> myPredProb = myPredOdds/
+   (1+myPredOdds)
> myPredProb[1:3,]
        est       2.5      97.5
1 0.9047619 0.6887475 0.9760679
2 0.8965517 0.7240012 0.9662542
3 0.9500000 0.7178176 0.9930027
```

A plot



Estimated

Figure: ~~Predicted~~ probabilities of infection over time by treatment group, with 95pct CI.

The Code

```
> Scol = c('active treatment' = 'red', placebo = 'black')
> weekShift = as.numeric(newx$weekFac) +
+  (newx$trt=='active treatment')/5
> plot(weekShift, myPredProb[,1], pch = 15,
+   ylim = range(myPredProb), bty='n',
+   col = Scol[as.character(newx$trt)], xaxt='n',
+   xlab='week', ylab='prob')
> segments(weekShift, myPredProb[,2], y1 = myPredProb[,3],
+   col = Scol[as.character(newx$trt)])
> axis(1, 1:nlevels(newx$weekFac), levels(newx$weekFac))
> legend('bottomleft', col=Scol, pch=15, pt.cex=2,
+   legend=names(Scol), bty='n')
```

→ no axis

Predictions with INLA

This is, unfortunately, a bit of work

```
> forLincombs = do.call(INLA::inla.make.lincombs,
+   as.data.frame(model.matrix( ~ weekFac*trt,
+     data=newx)))

> res2 = inla(infected ~ weekFac * trt +
                                                  log(2)
+   f(ID, model='iid', prior = 'pc.prec', param = c(2, 0.5)),
+   control.fixed = list(
+     mean = 0, mean.intercept = 0,
+     prec = 100^(-2), prec.intercept = 100^(-2)),
+   lincomb = forLincombs,
+   family = 'binomial', data=bacteria,
+   control.compute = list(config=TRUE),
+   control.inla = list(strategy='laplace', fast=FALSE))
```

What we get

```
> cbind(newx, res2$summary.lincomb.derived[,c(5,4,6)])[1:7,]

                    trt weekFac 0.5quant 0.025quant 0.975quant
lc01            placebo       0 3.232779  1.4125734   5.445617
lc02 active treatment       0 2.898449  1.4359891   4.584478
lc03            placebo       2 4.385277  2.0110292   7.094482
lc04 active treatment       2 2.747086  1.2495438   4.474118
lc05            placebo       4 2.017296  0.5780177   3.769063
lc06 active treatment       4 1.558050  0.4001076   2.941508
lc07            placebo       6 3.895301  1.5651737   6.475668
```

- these are posterior quantiles of $\tilde{X}\beta$
- where the 10 rows of $\tilde{X}$ are the different treatment/week combinations
- If $U_i = 0$ then logit$(\rho_{ij}) = X_{ij}\beta$
- We've produced posterior quantiles for 'typical' people with $U_i = 0$

A plot

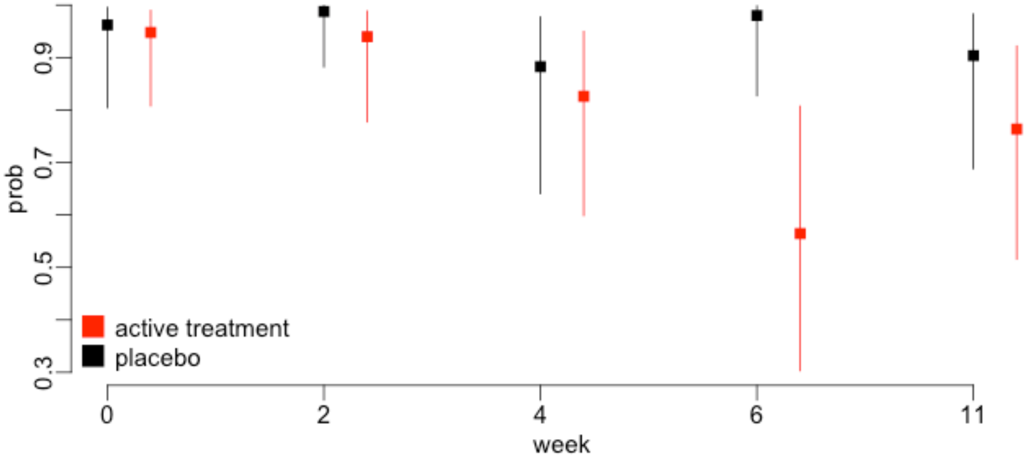


Figure: Posterior medians of probabilities of infection over time by treatment group, with 95pct CI.

The code

```
> predOddsInla = exp(res2$summary.lincomb.derived[,c(5,4,6)])
> predProbInla = predOddsInla/(1+predOddsInla)
> plot(weekShift, predProbInla[,1], pch = 15,
+   ylim = range(predProbInla), bty='n',
+   col = Scol[as.character(newx$trt)], xaxt='n', xlab='week', ylab='prob'
> segments(weekShift, predProbInla[,2], y1 = predProbInla[,3],
+   col = Scol[as.character(newx$trt)])
> axis(1, 1:nlevels(newx$weekFac), levels(newx$weekFac))
> legend('bottomleft', col=Scol, pch=15, pt.cex=2,
+   legend=names(Scol), bty='n')
```

In summary

- it doesn't matter whether you do `trt*week` or `trt:week` or `0 + trt*week`
- … if you use `predict` or `lincomb` to get the values you want
- so, get into the habit of using them!
- and make nice graphs

Posterior samples

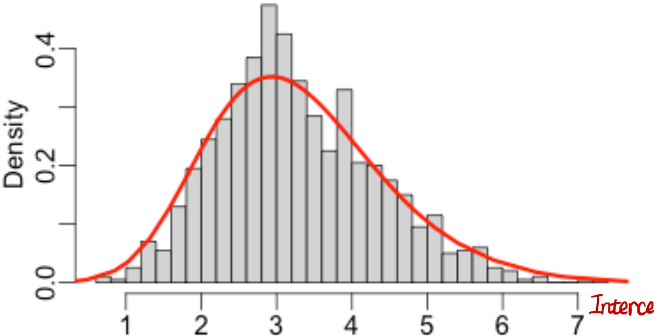
- Joint posterior

$$\pi(\beta, U, \sigma | Y)$$

- Sample from this distribution?
- useful for calculating nonlinear things

```
> mySample =
+   inla.posterior.sample(
+     1000, res2)
> length(mySample)
[1] 1000
> names(mySample[[1]])
[1] "hyperpar" "latent"    "logd
```

```
> sampleW = do.call(cbind,
+   Biobase::subListExtract(
+     mySample, 'latent'))
> hist(sampleW['(Intercept):1',],
+   prob=TRUE, main='', breaks=30, xlab='inte
> lines(
+   res2$marginals.fixed$'(Intercept)', col='
```



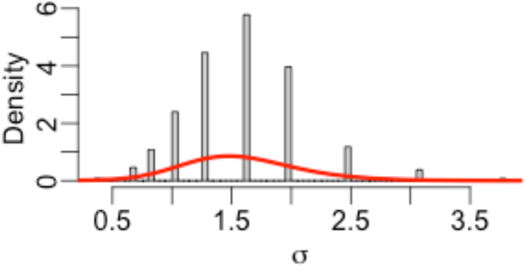
Intercept

How it works

- INLA approximates $[\sigma|Y]$ discretely
- at values $\sigma^{(1)} \ldots \sigma^{(K)}$ — configs.
- first sample $[\sigma^{(k)}|Y]$
- then sample $[W|Y, \sigma^{(k)}]$ → Normal approximation

```
> mySample[[1]]$hyperpar
Precision for ID
       0.2518642
> signif(sort(1/sqrt(exp(Biobase::subL
+      "theta", simplify = TRUE)))), 2)
 [1] 0.29 0.36 0.44 0.55 0.68
 [6] 0.85 1.00 1.30 1.60 2.00
[11] 2.50 3.10 3.80
```

```
> sampleTheta = Biobase::subListExtrac
+     mySample, 'hyperpar', simplify=T
> hist(1/sqrt(sampleTheta),
+   prob=TRUE, main='', breaks=50, xla
> lines(Pmisc::priorPost(res2)$sd$post
```



```
s,
```

some code

```
> sampleTheta = Biobase::subListExtract(
+     mySample, 'hyperpar', simplify=TRUE)
> hist(1/sqrt(sampleTheta),
+   prob=TRUE, main='', breaks=50, xlab=expression(sigma))
> lines(Pmisc::priorPost(res2)$sd$posterior, col='red', lwd=3)
```

an alternative to linear combinations

```
> sampleW = do.call(cbind, Biobase::subListExtract(mySample,
+   "latent"))
> dim(sampleW)

[1]  280 1000

> # rownames(sampleW)
> sampleBeta = t(sampleW[grep("Intercept|weekFac|trt", rownames(sampleW)),
+   ])
> colnames(sampleBeta) = gsub(":1$", "", colnames(sampleBeta))
> signif(sampleBeta[1:3, 1:4], 4)

           (Intercept) weekFac2 weekFac4 weekFac6
sample:1        3.767    1.104   -1.402  -0.1067
sample:2        4.968   -1.444   -2.801   1.1110
sample:3        4.493    1.671   -2.986  -0.3771
```

We have samples of $[\beta | Y]$ and want samples of $[\tilde{X}\beta | Y]$.

```
> newx2 = model.matrix(~weekFac * trt, newx)
> newx2[1:3, 1:4]

  (Intercept) weekFac2 weekFac4 weekFac6
1           1        0        0        0
2           1        0        0        0
3           1        1        0        0
```

$\tilde{X}$.

```
> sampleReparam = tcrossprod(sampleBeta[, colnames(newx2)],
+   newx2)
> dim(sampleReparam)

[1] 1000   10

> colnames(sampleReparam) = colnames(newx2)
> sampleNatural = exp(sampleReparam)/(1 + exp(sampleReparam))
> t(signif(apply(sampleNatural, 2, quantile, prob = c(0.5,
+   0.025, 0.975)), 3))

                                  50%  2.5% 97.5%
(Intercept)                     0.959 0.817 0.996
weekFac2                        0.948 0.819 0.989
weekFac4                        0.987 0.889 0.999
weekFac6                        0.940 0.769 0.987
weekFac11                       0.882 0.650 0.976
trtactive treatment             0.826 0.605 0.952
weekFac2:trtactive treatment    0.979 0.843 0.998
weekFac4:trtactive treatment    0.572 0.291 0.794
weekFac6:trtactive treatment    0.898 0.681 0.982
weekFac11:trtactive treatment   0.761 0.517 0.913
```

References I