Multivariate logistic distribution

June 17, 2019

1 Probit and multivariate logit model

We are interested in modeling multivariate binary outcomes (n observations and k response variables). One way to model this is to use a probit model (this is suggested in the Stan manual; it is different from the mixed model trick but I'll get to that later):

$$\begin{aligned} y_{n,k} &= 1(z_{n,k} > 0) \\ z_n &= x_n\beta + \epsilon_n \\ \epsilon_n &\sim \mathrm{N}(0, R) \end{aligned} \tag{1}$$

where R is a correlation matrix (with 1's on the diagonals). While this is a convenient way of modeling multivariate binary outcomes, its parameters are difficult to interpret. Instead, the multivariate logistic regression model proposed by O'brien and Dunson (2004) allows marginal distributions to follow a logistic distribution; the parameters of this model can be easily translated to odds ratio:

$$\begin{aligned} y_{n,k} &= 1(z_{n,k} > 0) \\ z_n &= x_n\beta + \log[F_\nu(e_n)/\{1 - F_\nu(e_n)\}] \\ e_n &\sim \mathrm{t}(0, \nu, R) \end{aligned} \tag{2}$$

where F_ν is the cumulative distribution function of a univariate t distribution with ν degrees of freedom and $t(0, \nu, R)$ is a multivariate t distribution with ν degrees of freedom, mean 0, and covariance R. Again, to avoid identifiability issues, R is constrained to have 1's on the diagonals (so it's essentially a correlation matrix). Sampling directly from this distribution is not trivial (Stan doesn't provide inverse function of CDF of a univariate t distribution); instead, we can use the approximation propsed by O'brien and Dunson (2004) and apply importance sampling to get to the correct posterior. Specifically, the distribution of z_n can be approximated by a multivariate t-distribution with mean $x_n\beta$, variance $\tilde{\sigma}^2 = \pi^2(\nu - 2)/3\nu$ ("a value chosen to make the variances of the univariate t and logistic distributions equal"), and $\tilde{\nu} = 7.3$ degrees of freedom ("a value chosen to minimize the integrated squared distance between the univariate t and univariate logistic densities").

Let's try some examples. We simulate 500 observations using (2) with intercepts only $\beta = (-1, 0, 1)$ and following correlation matrix:

$$R = \begin{pmatrix} 1 & 0.1 & 0.3 \\ 0.1 & 1 & -0.5 \\ 0.3 & -0.5 & 1 \end{pmatrix}.$$

This is what the data looks like:

```
load("../data/logit_t_intercept_only.rda")

head(simulate_dd)

##      V1 V2 V3
## [1,]  1  1  0
## [2,]  1  0  1
## [3,]  1  0  1
## [4,]  1  1  0
## [5,]  1  0  0
## [6,]  1  0  0
```

Then, I fit model (1) and model (2) to this data. First, looking at model (1):

```
load("../analysis/logit_t_intercept_only_logit_gaussian.rda")

ee <- rstan::extract(fit)
```

Looking at estimates of β (intercepts):

```
data.frame(
        median=apply(ee$beta, 2, median),
        lwr=apply(ee$beta, 2, quantile, 0.025),
        upr=apply(ee$beta, 2, quantile, 0.975)
)

##         median        lwr        upr
## 1  0.606912208  0.4954737  0.7326054
## 2  0.001469309 -0.1093642  0.1080785
## 3 -0.566471453 -0.6756364 -0.4523101
```

These are quite different from the β that I simulated with because the logit-normal doesn't preserve the marginal logit property. Interpreting each β as log-odds will be misleading. For example, these are "estimated" odds ratioes (naively assuming that these β estiamtes are log-odds):

```
data.frame(
        median=exp(apply(ee$beta, 2, median)),
        lwr=exp(apply(ee$beta, 2, quantile, 0.025)),
        upr=exp(apply(ee$beta, 2, quantile, 0.975))
)

##      median       lwr       upr
## 1 1.8347573 1.6412755 2.0804941
## 2 1.0014704 0.8964039 1.1141352
## 3 0.5675244 0.5088325 0.6361569
```

The observed odds ratios are quite different:

```
apply(simulate_dd, 2, mean)/(1-apply(simulate_dd, 2, mean))

##       V1       V2       V3
## 2.731343 1.008032 0.396648
```

Can we at least get correlation right?

```
apply(ee$Omega, 2:3, median)

##
##              [,1]       [,2]       [,3]
##   [1,] 1.0000000  0.1131334  0.2271666
##   [2,] 0.1131334  1.0000000 -0.4350445
##   [3,] 0.2271666 -0.4350445  1.0000000

apply(ee$Omega, 2:3, quantile, 0.025)

##
##                [,1]       [,2]        [,3]
##   [1,]  1.00000000 -0.0344030  0.08280362
##   [2,] -0.03440300  1.0000000 -0.55617938
##   [3,]  0.08280362 -0.5561794  1.00000000

apply(ee$Omega, 2:3, quantile, 0.975)

##
##              [,1]       [,2]       [,3]
##   [1,] 1.0000000  0.2550047  0.3808753
##   [2,] 0.2550047  1.0000000 -0.3081267
##   [3,] 0.3808753 -0.3081267  1.0000000
```

Seems like we're doing OK in terms of estimating the correlation.
Let's look at the estimates from the multivariate logisti regression model:

```r
load("../analysis/logit_t_intercept_only_logit_t.rda")

ee <- rstan::extract(fit)
```

Note that we have to take weighted quantiles this time. Look at estimates of β:

```r
## wquant from King et al.
wquant <- function (x, weights, probs = c(0.025, 0.975)) {
    which <- !is.na(weights)
    x <- x[which]
    weights <- weights[which]

    if (all(is.na(x)) || length(x) == 0) return(rep(NA, length(probs)))

    idx <- order(x)
    x <- x[idx]
    weights <- weights[idx]
    w <- cumsum(weights)/sum(weights)
    rval <- approx(w,x,probs,rule=1)
    rval$y
}

data.frame(
        median=apply(ee$beta, 2, wquant, weights=weights, probs=0.5),
        lwr=apply(ee$beta, 2, wquant, weights=weights, probs=0.025),
        upr=apply(ee$beta, 2, wquant, weights=weights, probs=0.975)
)

##         median        lwr        upr
## 1  0.995853424  0.8220403  1.2176655
## 2  0.006185303 -0.1761347  0.1640782
## 3 -0.921081770 -1.1175128 -0.7324204
```

These estimates match with the true values. Moreover, the estimates of odds ratios match with the observed odds (kind of obvious but just checking it):

```r
data.frame(
        median=exp(apply(ee$beta, 2, wquant, weights=weights, probs=0.5)),
        lwr=exp(apply(ee$beta, 2, wquant, weights=weights, probs=0.025)),
        upr=exp(apply(ee$beta, 2, wquant, weights=weights, probs=0.975))
)

##      median       lwr      upr
## 1 2.7070336 2.2751371 3.379290
## 2 1.0062045 0.8385050 1.178306
## 3 0.3980882 0.3270923 0.480744
```

Correlations look reasonable:

```
apply(ee$Omega, 2:3, wquant, weights=weights, probs=0.5)

##
##             [,1]       [,2]       [,3]
##   [1,] 1.0000000  0.1181755  0.2581493
##   [2,] 0.1181755  1.0000000 -0.4389358
##   [3,] 0.2581493 -0.4389358  1.0000000

apply(ee$Omega, 2:3, wquant, weights=weights, probs=0.025)

##
##               [,1]        [,2]       [,3]
##   [1,]  1.00000000 -0.02844068  0.1227903
##   [2,] -0.02844068  1.00000000 -0.5390483
##   [3,]  0.12279029 -0.53904828  1.0000000

apply(ee$Omega, 2:3, wquant, weights=weights, probs=0.975)

##
##             [,1]       [,2]       [,3]
##   [1,] 1.0000000  0.2657659  0.4059446
##   [2,] 0.2657659  1.0000000 -0.3141494
##   [3,] 0.4059446 -0.3141494  1.0000000
```

2 Improving probit model

O'brien and Dunson (2004) suggests using a t distribution for approximating the multivariate logit distribution but we can also use the probit model. Instead of using (1), we can rewrite it as

$$\begin{aligned} y_{n,k} &= 1(z_{n,k} > 0) \\ z_n &= x_n\beta + \epsilon_n \\ \epsilon_n &\sim \mathrm{N}(0, \sigma^2 R) \end{aligned} \tag{3}$$

where R is a correlation matrix and $\sigma^2 = \pi^2/3$. This allows the marginal distribution of z_n to have same variance as a standard logistic distribution. We fit this model to the simulated data above:

```
load("../analysis/logit_t_intercept_only_logit_gaussian2.rda")

ee <- rstan::extract(fit)
```

Looking at estimates of β:

```
data.frame(
        median=apply(ee$beta, 2, median),
        lwr=apply(ee$beta, 2, quantile, 0.025),
        upr=apply(ee$beta, 2, quantile, 0.975)
)

##        median        lwr        upr
## 1  1.12086705  0.9278208  1.3546859
## 2  0.01113124 -0.1854714  0.2043039
## 3 -1.03712738 -1.2560094 -0.8314504
```

This matches the true value much better. Converting these to odds ratios is consistent with the observed odds ratios.

```
data.frame(
        median=exp(apply(ee$beta, 2, median)),
        lwr=exp(apply(ee$beta, 2, quantile, 0.025)),
        upr=exp(apply(ee$beta, 2, quantile, 0.975))
)

##      median       lwr       upr
## 1 3.0675127 2.5289920 3.8755434
## 2 1.0111934 0.8307126 1.2266709
## 3 0.3544715 0.2847882 0.4354173
```

Correlations look good too:

```
apply(ee$Omega, 2:3, median)

##
##             [,1]       [,2]       [,3]
##   [1,] 1.0000000  0.1206049  0.2337311
##   [2,] 0.1206049  1.0000000 -0.4334244
##   [3,] 0.2337311 -0.4334244  1.0000000

apply(ee$Omega, 2:3, quantile, 0.025)

##
##               [,1]       [,2]        [,3]
##   [1,]  1.00000000 -0.0127358  0.07630079
##   [2,] -0.01273580  1.0000000 -0.55051889
##   [3,]  0.07630079 -0.5505189  1.00000000

apply(ee$Omega, 2:3, quantile, 0.975)

##
##             [,1]       [,2]       [,3]
##   [1,] 1.0000000  0.2513424  0.3790616
##   [2,] 0.2513424  1.0000000 -0.3140738
##   [3,] 0.3790616 -0.3140738  1.0000000
```

We should be able to apply importance sampling over this to make better inference but I'm not going to try it here...

3 Mixed model approach

We can also use the mixed model trick to model multivariate binary outcomes. Then, our model becomes

$$\begin{aligned} \text{logit } \Pr(Y_n = y_n) &= x_n\beta + \epsilon_n \\ \epsilon_n &\sim \mathrm{N}(0, \Sigma) \end{aligned} \tag{4}$$

I think this model is equivalent (???) to letting

$$\begin{aligned} y_{n,k} &= 1(z_{n,k} > 0) \\ z_{n,k} &\sim \mathcal{L}(x_n\beta_k + \epsilon_n) \\ \epsilon_n &\sim \mathrm{N}(0, \sigma^2 R), \end{aligned} \tag{5}$$

where $\mathcal{L}()$ represents the standard logistic distribution and $\sigma^2 = 1$. The major "problem" with this model is that the underlying distribution of z is overdispersed compared to the standard logistic distribution. This means that the estimates of β are not going to represent log-odds. So we need a way to shrink the distribution ??? or adjust the posterior using importance sampling. Or just use the first two models... Importance sampling is not going to work well because the posterior we get from this is going to be fairly different from the posterior we want.

Let's try to think of an approximation, assuming that the second model is equivalent to the first model (I'll check this later...). Note that the underlying distribution of z is a compound distribution between a logistic distribution and a normal distribution. Then, the variance of z is given by

$$\mathrm{Var}(Z) = \mathrm{E}_N(\mathrm{Var}_L(Z|\mu + \epsilon)) + \mathrm{Var}_N(\mathrm{E}_L(Z|\mu + \epsilon)) = \frac{\pi^2}{3} + 1 \approx 4.29,$$

which is the sum of variance of the logistic distribution and the underlying normal. We can confirm this using a small simulation:

```r
set.seed(101)
var(rlogis(1000000, location=rnorm(1000000)))

## [1] 4.291188
```

Then, we can approximate the marginal distribution of z with logistic distribution with scale

$$s = \sqrt{1 + \frac{3}{\pi^2}}$$

I think we can use this scale to scale the parameter estimates into the "correct" scale but I'm having some computational issues with model (4) even if I fix the variance parameter. It seems like it's harder to estimate correlation matrices... I suspect that the overdispersion makes things harder (unless I'm doing something wrong)...?

3.1 Checking mixed model "theory" for univariate cases

We can test some ideas on a univariate scale and then move to multivariate scale. First, let's show that (4) and (5) are equivalent using simulations:

```
set.seed(101)
nobs <- 100000
muvec <- (-4):4
sapply(muvec, function(x) {
        rr1 <- rbinom(nobs, 1, plogis(x + rnorm(nobs, 0, 1)))
        rr2 <- as.numeric(rlogis(nobs, location=x + rnorm(nobs, 0, 1)) > 0)
        mean(rr1) - mean(rr2)
})

## [1] -0.00005  0.00221  0.00088  0.00459  0.00258 -0.00315 -0.00073  0.00107
## [9]  0.00042
```

Looks like it works. We also want to know if we can approximately go from this logit-gaussian parameter scale to log odds scale. The idea is to approximate this compound logit-gaussian distribution with logistic distribution with different variance. I think we want to find $\hat{\beta}$ such that:

$$E[1(z > 0)] = \frac{1}{1 + \exp(-\hat{\beta})}. \tag{6}$$

given the underlying distribution of z. When z follows a standard logistic distribution, $\hat{\beta}$ corresponds to the location parameter of the logistic distribution of z. When z has a non-unit scale, we have

$$E[1(z > 0)] = \frac{1}{1 + \exp(-\mu/s)} \tag{7}$$

so $\hat{\beta} = \mu/s$, where s is a scale parameter.

In other words, if we use a mixed model trick where the underlying variance is assumed to be σ^2, we can transform the parameter estimates by dividing by

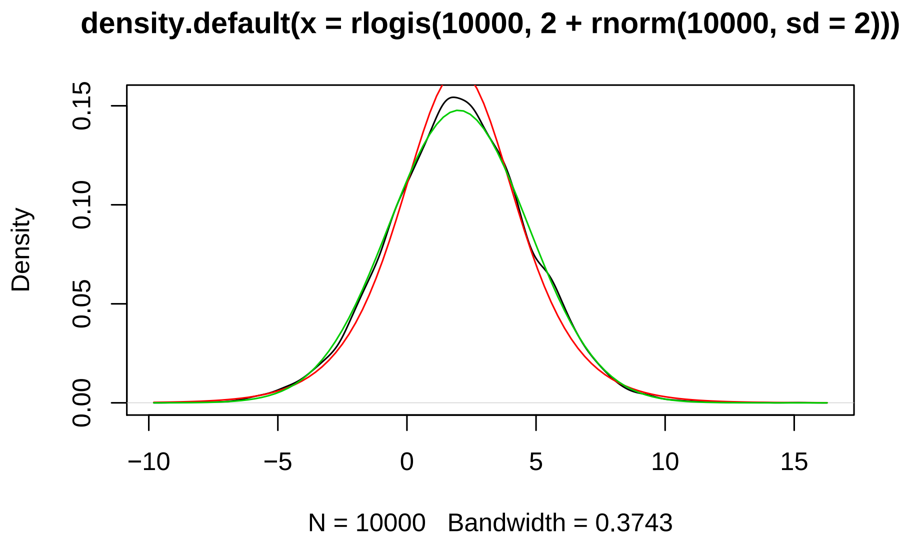
$$s = \sqrt{1 + \frac{3\sigma^2}{\pi^2}}.$$

This should (approximately) get us to the correct log-odds scale. Or we can even incorporate this value into our estimation process and apply importance sampling later (but this approximation seems good enough).

```
## mixed model mean probability
mean(rlogis(10000, 2 + rnorm(10000, sd=1)) > 0)

## [1] 0.8462

## approximation
plogis(2/sqrt(1 + 3/pi^2))

## [1] 0.8521354

## same example with different underlying variance of a normal distribution
## mixed model mean probability
mean(rlogis(10000, 2 + rnorm(10000, sd=2)) > 0)

## [1] 0.7767

## approximation
plogis(2/sqrt(1 + 4 * 3/pi^2))

## [1] 0.793076
```

These distributions are slightly different (which explains why the approximation doesn't work exactly)...

```
plot(density(rlogis(10000, 2 + rnorm(10000, sd=2))))
curve(dlogis(x, 2, scale=sqrt(1 + 4 * 3/pi^2)), add=TRUE, col=2)
curve(dnorm(x, 2, sd=sqrt(pi^2/3 + 4)), add=TRUE, col=3)
```



References

O'brien, S. M. and D. B. Dunson (2004). Bayesian multivariate logistic regression. *Biometrics 60*(3), 739–746.