negativebinomial_qc_vs_lme4_glmer

sarah

10/26/2021

I simulated a single dataset with sample size 10,000 and cluster size 5 under our NB base distribution model with betas = ones(3), single variance component Sigma = 0.5 and r = 10. Here I compare the fit of our NB regression with two most popular R packages for fitting Negative Binomial GLMM's with options to estimate "r":

1. lme4: The function glmer.nb estimates r pretty well, but very slowly. To get confidence intervals, it takes a significant amount of additional time, but no inference on r.
2. glmmTMB: The function glmmTMB estimates r pretty well and a little faster than lme4 but still slow. We can get the CI for all parameters in much less time than lme4.

Comparing Parameter Estimates

Parameter	Truth	Ours (CI)	lme4 (CI)	glmmTMB (CI)
Beta_1	1.000	0.972 (0.967, 0.977)	0.996 (0.989, 1.004)	0.998 (0.990, 1.005)
Beta_2	1.000	1.015 (1.012, 1.017)	0.991 (0.985, 0.997)	0.992 (0.986, 0.998)
Beta_3	1.000	1.016 (1.013, 1.019)	1.009 (1.003,1.015)	1.010 (1.004, 1.016)
Sigma_1	0.500	0.496 (0.472, 0.520)	0.013 (0.011, 0.016)	0.013 (0.011, 0.016)
r	10.000	8.926 (8.886, 8.966)	9.199 (NA, NA)	9.183 (8.844, 9.534)

Comparing Fit Times (CI times) in Seconds

Sample Size	Cluster Size	Ours (CI)	lme4 (CI)	glmmTMB (CI)
10000	5	6.28 (0.08)	126.18 (137.82)	104.59 (0.52)

lme4: glmer.nb

```
# Start the clock!
ptm <- proc.time()
m.nb <- glmer.nb(Y ~ 1 + X2 + X3 + (1|group), data = data, verbose = TRUE)
```

```
## theta.ml: iter 0 'theta = 0.327668'

## theta.ml: iter1 theta =0.570858

## theta.ml: iter2 theta =0.971684

## theta.ml: iter3 theta =1.61844

## theta.ml: iter4 theta =2.63462

## theta.ml: iter5 theta =4.17686

## theta.ml: iter6 theta =6.42055

## theta.ml: iter7 theta =9.52455

## theta.ml: iter8 theta =13.5576

## theta.ml: iter9 theta =18.3607

## theta.ml: iter10 theta =23.3398

## theta.ml: iter11 theta =27.3493

## theta.ml: iter12 theta =29.3015

## theta.ml: iter13 theta =29.6503

## theta.ml: iter14 theta =29.6596

## theta.ml: iter15 theta =29.6596

## th := est_theta(glmer(..)) = 29.65963 --> dev.= -2*logLik(.) = 207708
##  1: th=    14.60819974, dev=205748.84894280, beta[1]=    0.98745631
##  2: th=    60.21917735, dev=209949.06489691, beta[1]=    0.97124942
##  3: th=    6.087240107, dev=205777.35935810, beta[1]=    1.00649609
##  4: th=    9.548055584, dev=205251.75790778, beta[1]=    0.99552972
##  5: th=    9.487638797, dev=205250.60033162, beta[1]=    0.99568012
##  6: th=    9.145629923, dev=205248.13784438, beta[1]=    0.99651845
##  7: th=    7.828609557, dev=205322.49376410, beta[1]=    1.00042846
##  8: th=    9.199108881, dev=205248.03697117, beta[1]=    0.99638170
##  9: th=    9.201419099, dev=205248.04416087, beta[1]=    0.99638277
## 10: th=    9.182841738, dev=205248.05299758, beta[1]=    0.99642394
## 11: th=    9.192891987, dev=205248.04547195, beta[1]=    0.99640000
## 12: th=    9.197300869, dev=205248.04420570, beta[1]=    0.99638926
## 13: th=    9.199362905, dev=205248.04404626, beta[1]=    0.99638445
## 14: th=    9.198418240, dev=205248.04408267, beta[1]=    0.99638365
## 15: th=    9.198891038, dev=205248.04405486, beta[1]=    0.99638365
## 16: th=    9.199108881, dev=205248.04404807, beta[1]=    0.99638365
```

```r
# Stop the clock
proc.time() - ptm
```

```
##    user  system elapsed 
## 125.536   2.015 127.607
```

Show estimated r from lme4: glmer.nb

```r
getME(m.nb, "glmer.nb.theta")
```

```
## [1] 9.199109
```

Show confidence intervals from lme4: glmer.nb

```r
ptm <- proc.time()
confint(m.nb)
```

```
## Computing profile confidence intervals ...
```

```
##                 2.5 %    97.5 %
## .sig01      0.1043302 0.1248652
## (Intercept) 0.9887601 1.0040061
## X2          0.9854442 0.9974201
## X3          1.0033062 1.0154295
```

```r
# Stop the clock
proc.time() - ptm
```

```
##    user  system elapsed 
## 134.407   2.728 137.188
```

glmmTMB: glmmTMB

```
# Start the clock!
ptm <- proc.time()
m.glmmtmb_nb <- glmmTMB(Y ~ 1 + X2 + X3 + (1|group), data = data, family=nbinom2)
# Stop the clock
proc.time() - ptm
```

```
##    user  system elapsed
## 105.409   0.690 106.140
```

Show estimated r from glmmTMB

```
sigma(m.glmmtmb_nb)
```

```
## [1] 9.182583
```

Show confidence intervals from glmmTMB

```
ptm <- proc.time()
confint(m.glmmtmb_nb, full = TRUE)
```

```
##                                   2.5 %    97.5 %  Estimate
## cond.(Intercept)              0.9903703 1.0056084 0.9979893
## cond.X2                       0.9864608 0.9983947 0.9924278
## cond.X3                       1.0043137 1.0164268 1.0103703
## sigma                         8.8442912 9.5338144 9.1825830
## group.cond.Std.Dev.(Intercept) 0.1026876 0.1264207 0.1139378
```

```
# Stop the clock
proc.time() - ptm
```

```
##    user  system elapsed
##   0.513   0.028   0.541
```