Poisson GLMM (1)

Chris Parrish

August 19, 2016

Contents

4. Introduction to random effects: Conventional Poisson GLMM for count data **1**

4.2. Accounting for overdispersion by random effects-modeling in R and WinBUGS **1**

4.2.1. Generation and analysis of simulated data **1**

data . . . 1
model . . . 2
stan model . . . 2
call Stan from R . . . 3
results . . . 3

Poisson GLMM (1), BPA, chap 04

reference:
- Kéry and Schaub, 2012, Bayesian Populaton Analysis Using WinBUGS, Academic Press

```
library(rstan)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
```

4. Introduction to random effects: Conventional Poisson GLMM for count data

4.2. Accounting for overdispersion by random effects-modeling in R and WinBUGS

4.2.1. Generation and analysis of simulated data

data

```
## Read data
## The data generation code is in bpa-code.txt, available at
## http://www.vogelwarte.ch/de/projekte/publikationen/bpa/complete-code-and-data-files-of-the-book.html
stan_data <- read_rdump("GLMM_Poisson.data.R")
```

model

stan model

GLMM_Poisson.stan

```
data {
  int<lower=0> n;       // Number of years
  int<lower=0> C[n];    // Counts
  vector[n] year;       // Year
}

parameters {
  real<lower=-20,upper=20> alpha;
  real<lower=-10,upper=10> beta1;
  real<lower=-10,upper=20> beta2;
  real<lower=-10,upper=10> beta3;
  vector[n] eps;        // Year effects
  real<lower=0,upper=5> sigma;
}

transformed parameters {
  vector[n] log_lambda;

  // Linear predictor incl. random year effect
 log_lambda = alpha +
               beta1 * year +
               beta2 * year .* year +
               beta3 * year .* year .* year +
               eps;
}

model {
  // Priors
  alpha ~ uniform(-20, 20);
  beta1 ~ uniform(-10, 10);
  beta2 ~ uniform(-10, 10);
  beta3 ~ uniform(-10, 10);
  sigma ~ uniform(0, 5);

  // Likelihood
  C ~ poisson_log(log_lambda);
  eps ~ normal(0, sigma);
}

generated quantities {
  vector<lower=0>[n] lambda;

  lambda = exp(log_lambda);
}
```

```
## Initial values
inits <- function() list(alpha = runif(1, -2, 2),
                         beta1 = runif(1, -3, 3),
                         sd = runif(1, 0, 1))
```

```r
## Parameters monitored
params <- c("alpha", "beta1", "beta2", "beta3", "lambda", "sigma",
            "eps")

# MCMC settings
ni <- 15000
nt <- 10
nb <- 5000
nc <- 4
```

call Stan from R

```r
## Call Stan from R
fit <- stan("GLMM_Poisson.stan", data = stan_data,
            init = inits, pars = params,
            chains = nc, iter = ni, warmup = nb, thin = nt,
            seed = 1,
            open_progress = FALSE)
```

results

```r
## Summarize posteriors
print(fit)
```

```
## Inference for Stan model: GLMM_Poisson.
## 4 chains, each with iter=15000; warmup=5000; thin=10;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##                 mean se_mean   sd   2.5%    25%    50%    75%
## alpha           4.30    0.00 0.03   4.24   4.28   4.30   4.32
## beta1           1.25    0.00 0.05   1.15   1.21   1.25   1.28
## beta2           0.07    0.00 0.02   0.03   0.06   0.07   0.09
## beta3          -0.23    0.00 0.03  -0.28  -0.24  -0.23  -0.21
## lambda[1]      32.07    0.06 3.41  25.54  29.72  32.02  34.25
## lambda[2]      30.62    0.05 2.87  25.11  28.65  30.50  32.42
## lambda[3]      29.12    0.04 2.46  24.48  27.41  29.07  30.71
## lambda[4]      28.32    0.04 2.22  24.08  26.84  28.20  29.72
## lambda[5]      26.98    0.03 2.00  23.04  25.67  26.96  28.27
## lambda[6]      27.40    0.03 1.88  23.82  26.15  27.33  28.60
## lambda[7]      27.22    0.03 1.82  23.72  26.06  27.17  28.37
## lambda[8]      27.71    0.03 1.80  24.21  26.54  27.75  28.82
## lambda[9]      28.86    0.03 1.87  25.26  27.64  28.81  30.01
## lambda[10]     30.02    0.03 1.91  26.34  28.77  29.97  31.23
## lambda[11]     32.18    0.03 2.07  28.40  30.81  32.09  33.43
## lambda[12]     34.64    0.04 2.31  30.62  33.11  34.47  35.96
## lambda[13]     36.44    0.04 2.28  32.25  34.91  36.37  37.85
## lambda[14]     39.07    0.04 2.40  34.44  37.54  39.05  40.60
## lambda[15]     42.92    0.04 2.58  38.20  41.19  42.80  44.51
## lambda[16]     47.99    0.05 2.99  42.73  45.96  47.81  49.79
## lambda[17]     52.09    0.05 3.01  46.54  50.06  51.91  53.92
## lambda[18]     56.10    0.06 3.16  49.79  54.10  56.13  58.18
```

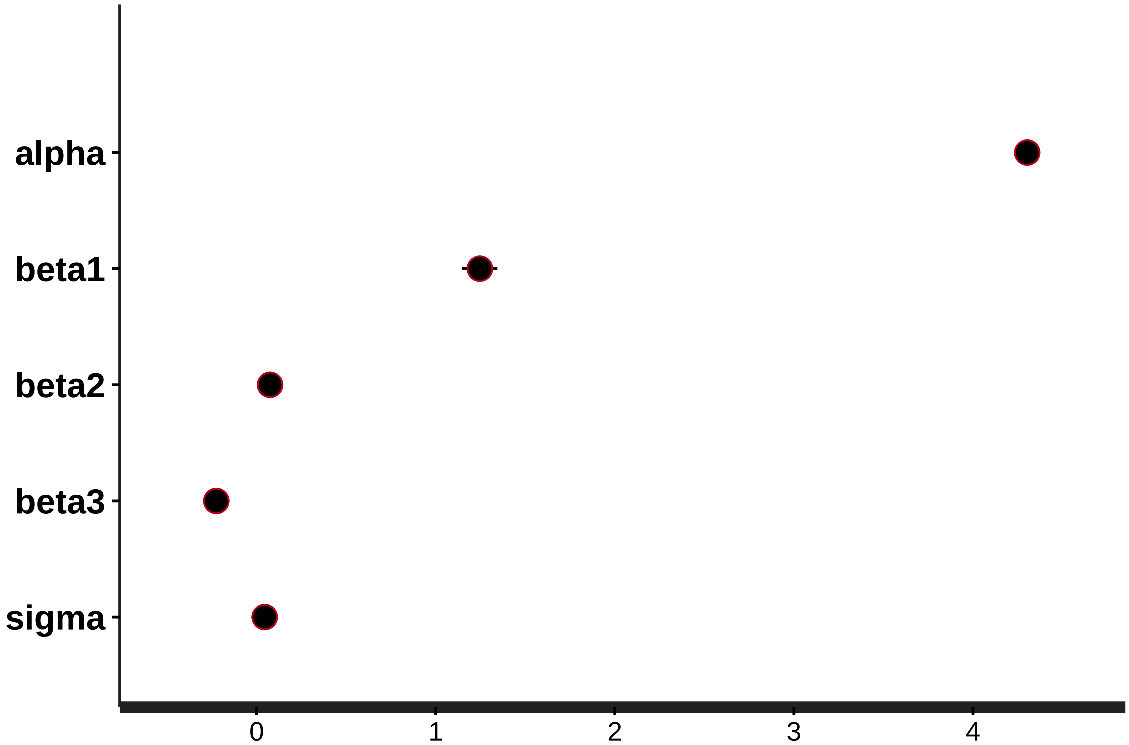
```
## lambda[19]    63.06    0.06  3.41    56.51    60.81    63.02    65.21
## lambda[20]    72.16    0.07  4.12    64.91    69.46    71.78    74.48
## lambda[21]    75.07    0.08  4.29    65.66    72.51    75.40    77.93
## lambda[22]    86.85    0.07  4.36    78.20    84.08    86.79    89.53
## lambda[23]    95.60    0.07  4.68    86.17    92.66    95.64    98.63
## lambda[24]   106.07    0.09  5.33    95.30   102.81   106.21   109.41
## lambda[25]   119.73    0.09  5.54   109.20   116.18   119.58   123.08
## lambda[26]   132.08    0.10  6.18   119.97   128.20   131.91   135.79
## lambda[27]   146.77    0.11  6.81   134.19   142.38   146.44   150.78
## lambda[28]   156.97    0.12  7.17   141.95   152.50   157.17   161.74
## lambda[29]   166.33    0.16  8.73   146.83   160.90   167.13   172.41
## lambda[30]   198.74    0.19  9.90   182.04   192.14   197.58   204.52
## lambda[31]   205.68    0.14  8.74   188.63   200.13   205.61   211.15
## lambda[32]   211.95    0.17  9.67   191.50   205.82   212.56   218.54
## lambda[33]   238.33    0.18 10.28   220.23   231.47   237.44   244.43
## lambda[34]   248.32    0.16 10.13   229.63   241.69   247.80   254.30
## lambda[35]   255.79    0.17 10.41   236.08   248.86   255.46   262.26
## lambda[36]   261.70    0.17 10.55   241.29   254.94   261.40   268.08
## lambda[37]   262.24    0.18 10.71   240.73   255.25   262.06   269.10
## lambda[38]   263.51    0.17 11.05   242.81   255.99   263.33   270.51
## lambda[39]   262.03    0.18 11.62   240.20   254.02   261.81   269.93
## lambda[40]   251.25    0.20 12.49   227.08   242.83   251.05   259.77
## sigma          0.05    0.00  0.02     0.02     0.03     0.04     0.06
## eps[1]        -0.02    0.00  0.05    -0.13    -0.04    -0.01     0.01
## eps[2]         0.01    0.00  0.05    -0.09    -0.02     0.00     0.03
## eps[3]         0.01    0.00  0.05    -0.09    -0.02     0.00     0.03
## eps[4]         0.01    0.00  0.05    -0.08    -0.01     0.01     0.04
## eps[5]        -0.01    0.00  0.05    -0.13    -0.04    -0.01     0.02
## eps[6]         0.01    0.00  0.05    -0.09    -0.02     0.00     0.03
## eps[7]        -0.01    0.00  0.05    -0.11    -0.03    -0.01     0.02
## eps[8]        -0.01    0.00  0.05    -0.12    -0.04    -0.01     0.02
## eps[9]         0.00    0.00  0.05    -0.11    -0.03     0.00     0.02
## eps[10]       -0.01    0.00  0.05    -0.11    -0.03     0.00     0.02
## eps[11]        0.01    0.00  0.05    -0.08    -0.02     0.01     0.04
## eps[12]        0.02    0.00  0.05    -0.07    -0.01     0.01     0.05
## eps[13]        0.00    0.00  0.05    -0.10    -0.03     0.00     0.03
## eps[14]       -0.01    0.00  0.05    -0.11    -0.03    -0.01     0.02
## eps[15]        0.00    0.00  0.05    -0.09    -0.03     0.00     0.03
## eps[16]        0.02    0.00  0.05    -0.07    -0.01     0.02     0.05
## eps[17]        0.01    0.00  0.05    -0.08    -0.02     0.01     0.03
## eps[18]       -0.02    0.00  0.05    -0.12    -0.04    -0.01     0.01
## eps[19]        0.00    0.00  0.05    -0.10    -0.03     0.00     0.02
## eps[20]        0.03    0.00  0.05    -0.06     0.00     0.02     0.06
## eps[21]       -0.04    0.00  0.05    -0.17    -0.07    -0.03     0.00
## eps[22]        0.00    0.00  0.05    -0.09    -0.03     0.00     0.03
## eps[23]       -0.01    0.00  0.04    -0.11    -0.04    -0.01     0.02
## eps[24]       -0.01    0.00  0.05    -0.11    -0.04    -0.01     0.01
## eps[25]        0.00    0.00  0.04    -0.08    -0.02     0.00     0.03
## eps[26]        0.00    0.00  0.04    -0.09    -0.02     0.00     0.03
## eps[27]        0.01    0.00  0.04    -0.07    -0.02     0.01     0.03
## eps[28]       -0.02    0.00  0.04    -0.11    -0.04    -0.01     0.01
## eps[29]       -0.05    0.00  0.05    -0.17    -0.08    -0.04    -0.01
## eps[30]        0.05    0.00  0.05    -0.03     0.01     0.04     0.07
## eps[31]        0.01    0.00  0.04    -0.07    -0.02     0.01     0.03
```

```
## eps[32]       -0.03    0.00  0.04    -0.12    -0.05    -0.02     0.00
## eps[33]        0.03    0.00  0.04    -0.04     0.00     0.02     0.05
## eps[34]        0.02    0.00  0.04    -0.05     0.00     0.02     0.04
## eps[35]        0.01    0.00  0.04    -0.06    -0.01     0.01     0.03
## eps[36]        0.01    0.00  0.04    -0.07    -0.02     0.01     0.03
## eps[37]       -0.01    0.00  0.04    -0.09    -0.03     0.00     0.02
## eps[38]        0.00    0.00  0.04    -0.09    -0.03     0.00     0.02
## eps[39]        0.00    0.00  0.04    -0.08    -0.02     0.00     0.03
## eps[40]       -0.02    0.00  0.04    -0.11    -0.04    -0.01     0.01
## lp__       18278.67    0.41 16.49 18250.98 18266.71 18277.17 18288.99
##               97.5% n_eff Rhat
## alpha          4.36  3653    1
## beta1          1.34  3062    1
## beta2          0.12  3586    1
## beta3         -0.17  3225    1
## lambda[1]     39.32  3509    1
## lambda[2]     36.69  3909    1
## lambda[3]     34.09  3664    1
## lambda[4]     33.06  3964    1
## lambda[5]     30.85  4000    1
## lambda[6]     31.28  3946    1
## lambda[7]     30.87  3936    1
## lambda[8]     31.34  3631    1
## lambda[9]     32.73  3941    1
## lambda[10]    33.97  3641    1
## lambda[11]    36.73  3572    1
## lambda[12]    39.99  3644    1
## lambda[13]    41.24  3584    1
## lambda[14]    43.81  3592    1
## lambda[15]    48.45  4000    1
## lambda[16]    54.34  3515    1
## lambda[17]    58.65  3481    1
## lambda[18]    62.46  2890    1
## lambda[19]    70.09  3767    1
## lambda[20]    81.39  3617    1
## lambda[21]    82.76  3034    1
## lambda[22]    95.70  4000    1
## lambda[23]   104.70  4000    1
## lambda[24]   116.26  3797    1
## lambda[25]   131.49  4000    1
## lambda[26]   144.96  4000    1
## lambda[27]   161.30  3661    1
## lambda[28]   170.82  3768    1
## lambda[29]   181.40  2840    1
## lambda[30]   221.30  2852    1
## lambda[31]   223.53  4000    1
## lambda[32]   229.70  3431    1
## lambda[33]   261.81  3421    1
## lambda[34]   269.95  3787    1
## lambda[35]   277.24  3966    1
## lambda[36]   283.84  4000    1
## lambda[37]   283.48  3701    1
## lambda[38]   286.92  4000    1
## lambda[39]   285.03  4000    1
```

```
## lambda[40]   276.15  3763    1
## sigma          0.09  1836    1
## eps[1]         0.08  3590    1
## eps[2]         0.11  3860    1
## eps[3]         0.12  3991    1
## eps[4]         0.12  3585    1
## eps[5]         0.08  4000    1
## eps[6]         0.11  4000    1
## eps[7]         0.09  3545    1
## eps[8]         0.08  4000    1
## eps[9]         0.09  3815    1
## eps[10]        0.09  4000    1
## eps[11]        0.11  3881    1
## eps[12]        0.14  4000    1
## eps[13]        0.10  4000    1
## eps[14]        0.08  3582    1
## eps[15]        0.10  4000    1
## eps[16]        0.14  3722    1
## eps[17]        0.11  3750    1
## eps[18]        0.08  3382    1
## eps[19]        0.09  3935    1
## eps[20]        0.14  3868    1
## eps[21]        0.04  3010    1
## eps[22]        0.09  4000    1
## eps[23]        0.07  3971    1
## eps[24]        0.07  3588    1
## eps[25]        0.09  4000    1
## eps[26]        0.09  4000    1
## eps[27]        0.10  3893    1
## eps[28]        0.06  3810    1
## eps[29]        0.03  2788    1
## eps[30]        0.16  2898    1
## eps[31]        0.09  4000    1
## eps[32]        0.04  3463    1
## eps[33]        0.12  3178    1
## eps[34]        0.11  3657    1
## eps[35]        0.09  4000    1
## eps[36]        0.09  3912    1
## eps[37]        0.07  3947    1
## eps[38]        0.08  3893    1
## eps[39]        0.09  4000    1
## eps[40]        0.07  3754    1
## lp__       18315.48  1595    1
##
## Samples were drawn using NUTS(diag_e) at Fri Aug 19 14:53:34 2016.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
##  The estimated Bayesian Fraction of Missing Information is a measure of
##  the efficiency of the sampler with values close to 1 being ideal.
##  For each chain, these estimates are
##  0.2 0.2 0.1 0.2
```

```r
plot(fit, pars=c("alpha", "beta1", "beta2", "beta3", "sigma"))
```

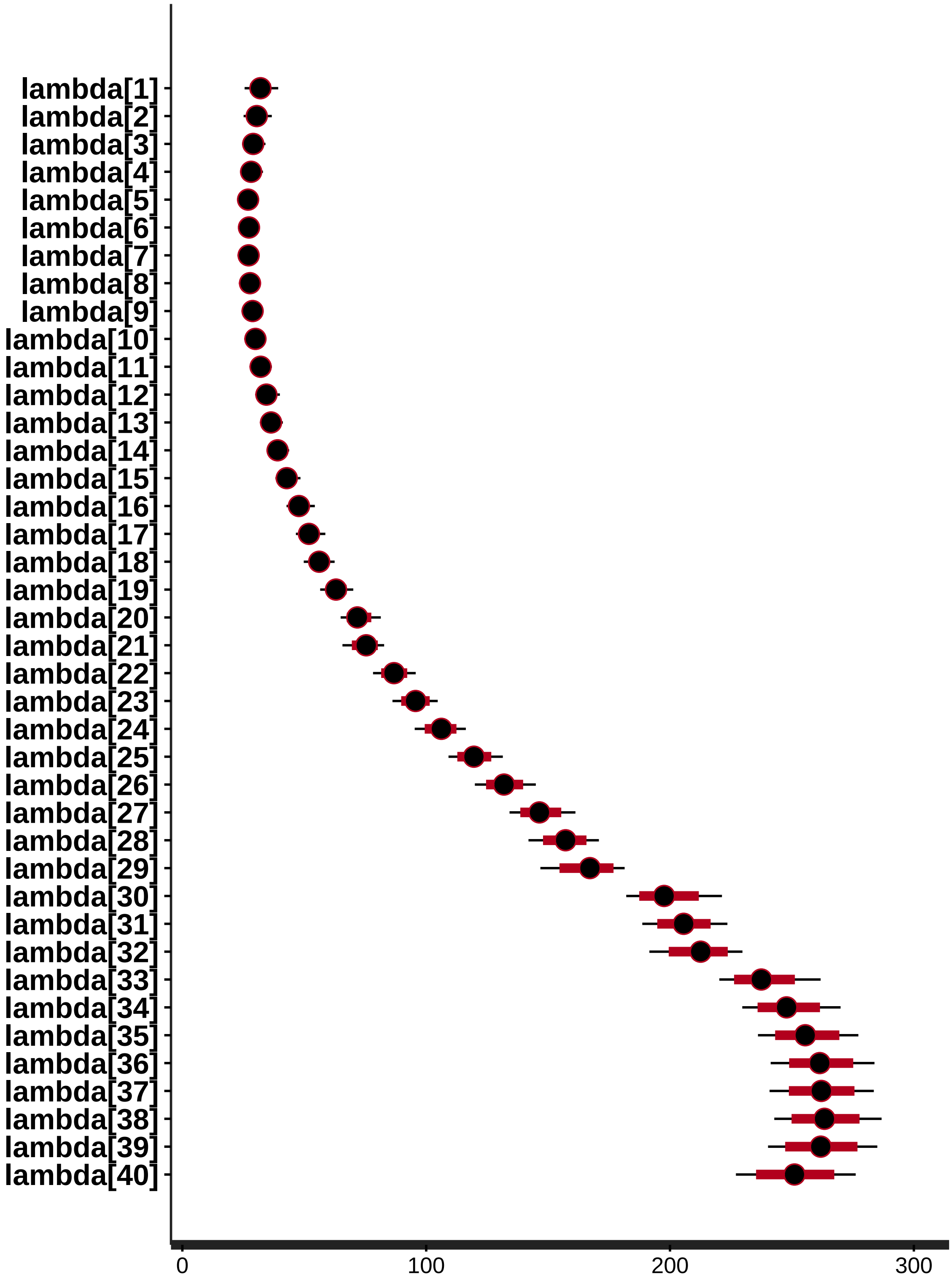
```
## ci_level: 0.8 (80% intervals)

## outer_level: 0.95 (95% intervals)
```



```
plot(fit, pars=c("lambda"))
```

```
## ci_level: 0.8 (80% intervals)
```

```
## outer_level: 0.95 (95% intervals)
```



```r
plot(fit, pars=c("eps"))
```

```
## ci_level: 0.8 (80% intervals)
```

```
## outer_level: 0.95 (95% intervals)
```