Steel_MultilevelGroup

Zack Steel

October 27, 2016

Trends of fire patterns over time

Synopsis and questions for UCD Multilevel modeling group

Highlights:

– Model types: Regression with random slopes and intercepts; gaussian, binomial, and potentially zero-inflated error structures

– Modeling Packages/approach: Richard's rethinking and Rstan

– Feedback sought:

1) Best practices for model checking?

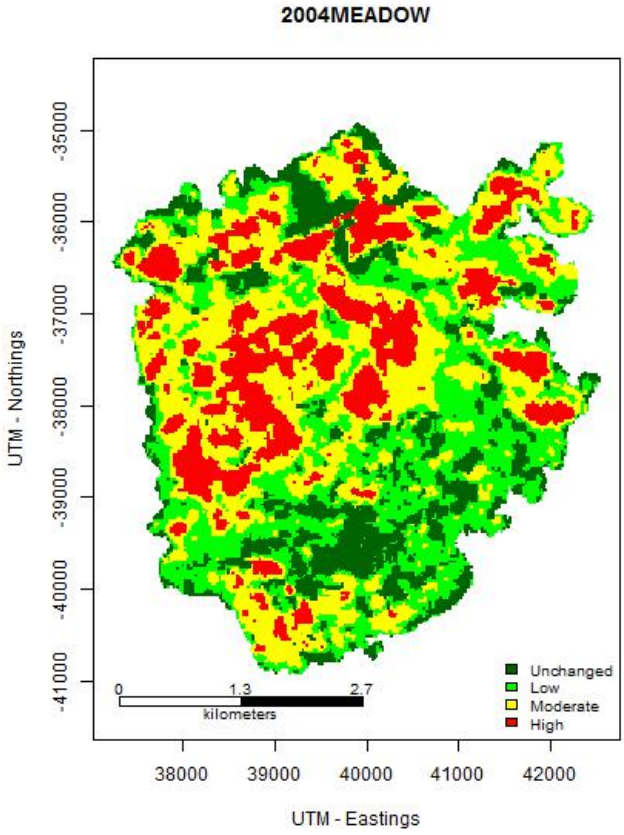
2) How to deal with sorta zero-inflated data?

3) Confirmation that I don't have temporal auto-correlation. I.e. Time-series always contain time data but time data are not always a time series.

4) Generally, do my methods/code look sound? What issues might a reviewer bring up?

Project summary:

We are testing whether the landscape patterns created by fire in California's conifer forests are changing over time. The data this analysis is based on is Landsat imagery - before and after fire, which is used to estimate changes in vegetation (i.e Fire severity). For example, this is what one of these fires looks like as classified by burn severity:



For the project I am focusing on both "unchanged" and "high severity" patches, but for now we'll just look at high severity, which essentially represents stand-replacing fire where all vegetation is killed.

The models I'm running are conceptually pretty simple. Our sample unit is the individual fire. We have an outcome variable such as proportion area burned at high severity, which are calculated for each fire, and I have year as the sole fixed effect. Year is a continuous variable ranging from 1984 to 2014. We expect fire patterns and trends to vary somewhat by forest type and by region within California, so we also allow intercepts and slopes to vary by these two categorical variables (i.e. two random effects).

There are three flavors of this basic model depending on the outcome variable of interest. I will present one of each:

1) Outcome - Edge density (log m/m^2); Error structure: Gaussian
2) Outcome - Proportion area; Error structure: binomial where the outcome is the number of pixels that burned at high severity out of the number of pixels in each fire.
3) Outcome - Core area (log m^2); Error structure: Gaussian or zero-inflated?

Model code

Below are how I've run these three models using Richard McElreath's rethinking package and Stan. Subsequently I'll show some model diagnostis and a couple example results plots.

But first what the data look like: We have fire name, fire year, fire year - centered, dominant forest type wihtin the fire perimeter, forest type id, bioregion, bioregion id, log of edge density, severity proportion (this is broken out into successes and failures for the actual model), and log of core area within high severity patches ($>$ 100m from the patch edge).

```
## First I'll bring in the data and the pre-run models
load("Scripts/Learning/multilevgrp.RData")

## Give you a taste of what the data look like
dplyr::select(dd,VB_ID, FIRE_YEAR, fy_c, dom_pfr, pfr_id,
              bioregion, br_id, logedgearea, sev_prop,
              logcore) %>% head()
```

```
##           VB_ID FIRE_YEAR      fy_c       dom_pfr pfr_id
## 1    1984BADGER      1984 -16.77183 Mixed conifer      2
## 2      1984RAIL      1984 -16.77183 Mixed conifer      2
## 3  1985BARTLETT      1985 -15.77183 Mixed conifer      2
## 4     1985DELTA      1985 -15.77183 Mixed conifer      2
## 5 1985DOUGHERTY      1985 -15.77183 Mixed conifer      2
## 6  1985KENDRICK      1985 -15.77183 Mixed conifer      2
##              bioregion br_id logedgearea   sev_prop  logcore
## 1    Southern Cascades     6    5.442268 0.17565754 11.74021
## 2 Northeastern Plateau     3    4.474274 0.64713870 14.95272
## 3        Sierra Nevada     4    5.456922 0.20553994 11.04402
## 4    Klamath Mountains     1    4.187476 0.46925243 14.61554
## 5        Sierra Nevada     4    5.701798 0.07777688 10.25857
## 6        Sierra Nevada     4    5.982030 0.06012873  0.00000
```

```
## Edge density of high severity patches
x <- select(dd, logedgearea, pfr_id, br_id, fy_c) #Pulling out variables needed
x <- x[complete.cases(x),] #there are few fires that have no high severity component so edge density do
logedgearea.m <- map2stan(
  alist(
    # likelihood
    logedgearea ~ dnorm(mu, sigma),

    # linear models
    mu <- A + By*fy_c, #split this out into submodels for clarity
    A <- a + a_pfr[pfr_id] + a_br[br_id],
    By <- by + by_pfr[pfr_id] + by_br[br_id],

    # adaptive priors
    c(a_pfr,by_pfr)[pfr_id] ~ dmvnormNC(sigma_pfr, Rho_pfr),
    c(a_br,by_br)[br_id] ~ dmvnormNC(sigma_br, Rho_br),

    # fixed priors
    a ~ dnorm(0, 10),
    by ~ dnorm(0, 1),
    c(sigma_pfr,sigma_br,sigma) ~ dcauchy(0,2),
    c(Rho_pfr,Rho_br) ~ dlkjcorr(2)),
  data <- x,iter=3000 , warmup=1500 , chains=2,
  control=list(adapt_delta=0.95))



## Proportion area in high severity
x <- select(dd, Shape_Area, fire_area, fire_id, pfr_id, br_id, fy_c)
## Binomials need integers
```

```
x$Shape_Area <- as.integer(x$Shape_Area/900) #Pixel scale
x$fire_area <- as.integer(x$fire_area/900)

## Get start values to help stan out
resp <- cbind(x$Shape_Area, (x$fire_area - x$Shape_Area)) #cbind(successes, failures)
m0 <- glm(resp ~ x$fy_c, family='binomial')

## Run the model
prop_area.m <- map2stan(
  alist(
    # likelihood
    Shape_Area ~ dbinom(fire_area, p), #Shape_area = "successful" pixels"; fire_area = n pixels

    # linear models
    logit(p) <- A + By*fy_c, #split this out into submodels for clarity
    A <- a + a_fire[fire_id] + a_pfr[pfr_id] + a_br[br_id], #add fire id, needed because we have repeat
    By <- by + by_pfr[pfr_id] + by_br[br_id],

    # adaptive priors
    c(a_pfr,by_pfr)[pfr_id] ~ dmvnormNC(sigma_pfr, Rho_pfr),
    c(a_br,by_br)[br_id] ~ dmvnormNC(sigma_br, Rho_br),
    a_fire[fire_id] ~ dnorm(0, sigma_fire),

    # fixed priors
    a ~ dnorm(0, 1),
    by ~ dnorm(0, 0.5),
    c(sigma_pfr,sigma_br,sigma_fire) ~ dcauchy(0,2),
    c(Rho_pfr,Rho_br) ~ dlkjcorr(2)),
  data <- x,iter=3000 , warmup=1500 , chains=2,
  start=list(a=coef(m0)[1], by=coef(m0)[2]), cores = 2,
  control=list(adapt_delta=0.95,max_treedepth=20), WAIC=F)


## core area of HS; first a simple gaussian model
x <- select(dd, logcore, pfr_id, br_id, fy_c)
logcore.m <- map2stan(
  alist(
    # likelihood
    logcore ~ dnorm(mu, sigma),

    # linear models
    mu <- A + By*fy_c, #split this out into submodels for clarity
    A <- a + a_pfr[pfr_id] + a_br[br_id],
    By <- by + by_pfr[pfr_id] + by_br[br_id],

    # adaptive priors
    c(a_pfr,by_pfr)[pfr_id] ~ dmvnormNC(sigma_pfr, Rho_pfr),
    c(a_br,by_br)[br_id] ~ dmvnormNC(sigma_br, Rho_br),

    # fixed priors
    a ~ dnorm(0, 10),
    by ~ dnorm(0, 1),
    c(sigma_pfr,sigma_br,sigma) ~ dcauchy(0,2),
```

```
    c(Rho_pfr,Rho_br) ~ dlkjcorr(2)),
  data <- x,iter=3000 , warmup=1500 , chains=2,
  control=list(adapt_delta=0.95))
## zero-inflated normal distribution?
```

Model diagnostics

Note in the precis outputs it warns of divergent iterations. Reading rethinking it doesn't sound like this is necessarily a problem if your Rhat and traceplots look ok, but I'm not exactly sure what this means. I just show a few traceplots. The rest look pretty good to me.

Question: Is there an easy way to look at residual plots for guassian models? Do you need to do something similar for binomial models?

```
## looking at first level of precis tables
precis(ms$logedgearea.m)
```

```
## Warning in precis(ms$logedgearea.m): There were 77 divergent iterations during sampling.
## Check the chains (trace plots, n_eff, Rhat) carefully to ensure they are valid.

## 60 vector or matrix parameters omitted in display. Use depth=2 to show them.

##       Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## a     5.20   0.45       4.53       6.00   101 1.02
## by    0.00   0.02      -0.02       0.02   563 1.00
## sigma 0.56   0.02       0.53       0.59   244 1.01
```

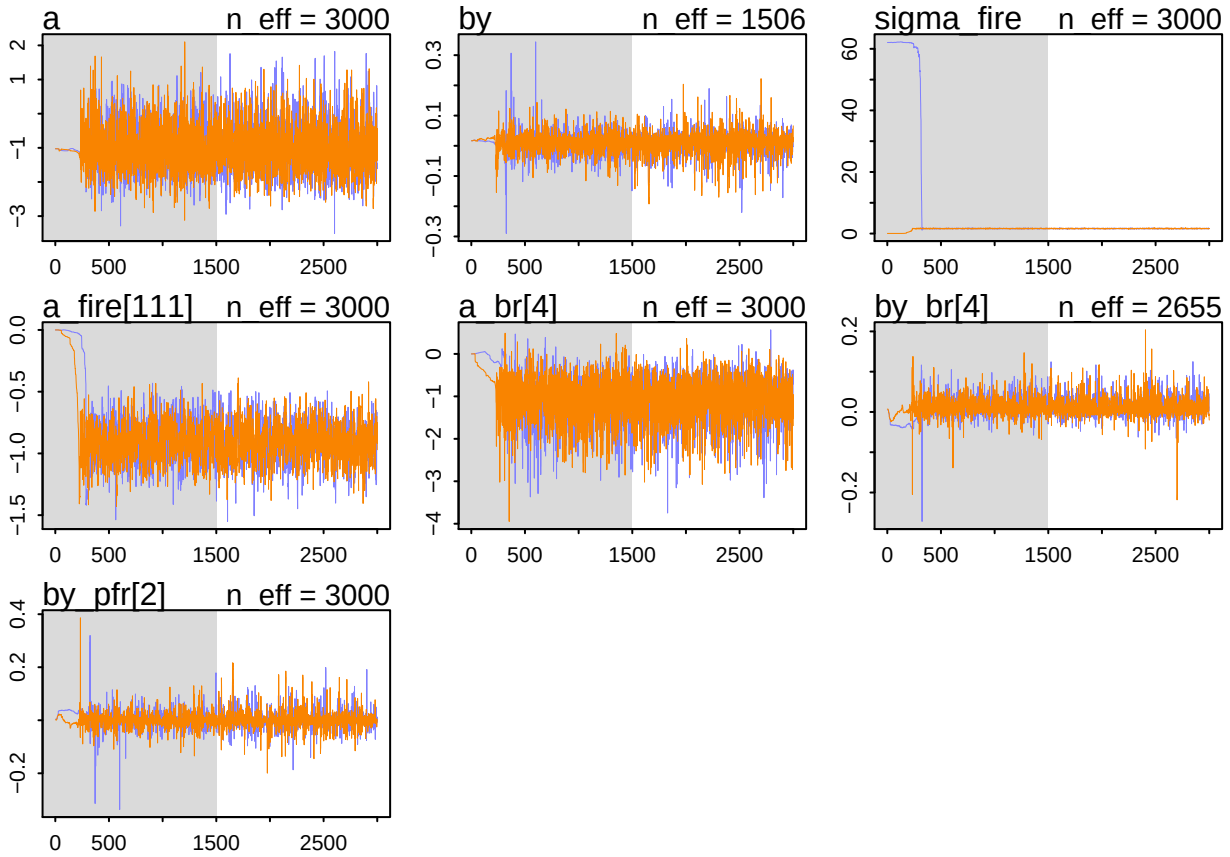
```
precis(ms$prop_area.m)
```

```
## Warning in precis(ms$prop_area.m): There were 91 divergent iterations during sampling.
## Check the chains (trace plots, n_eff, Rhat) carefully to ensure they are valid.

## 415 vector or matrix parameters omitted in display. Use depth=2 to show them.

##             Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## a          -0.99   0.71      -2.09       0.13  3000    1
## by          0.01   0.04      -0.05       0.06  1506    1
## sigma_fire  1.63   0.07       1.53       1.74  2719    1
```

```
precis(ms$logcore.m)
```

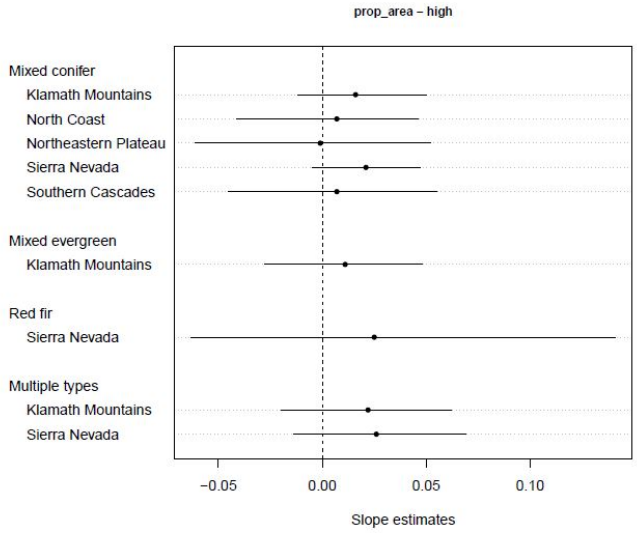
```
## Warning in precis(ms$logcore.m): There were 19 divergent iterations during sampling.
## Check the chains (trace plots, n_eff, Rhat) carefully to ensure they are valid.

## 60 vector or matrix parameters omitted in display. Use depth=2 to show them.

##        Mean StdDev lower 0.89 upper 0.89 n_eff Rhat
## a     10.05   0.83       8.83      11.36   902    1
## by     0.00   0.12      -0.19       0.15   667    1
## sigma  4.89   0.09       4.74       5.03  2383    1
```

```
## Looking at some traceplots, fixed effects and some arbitratily chosen random intercepts/slopes
plot(ms$prop_area.m, pars=c("a","by","sigma_fire", "a_fire[111]", "a_br[4]", "by_br[4]", "by_pfr[2]"))
```

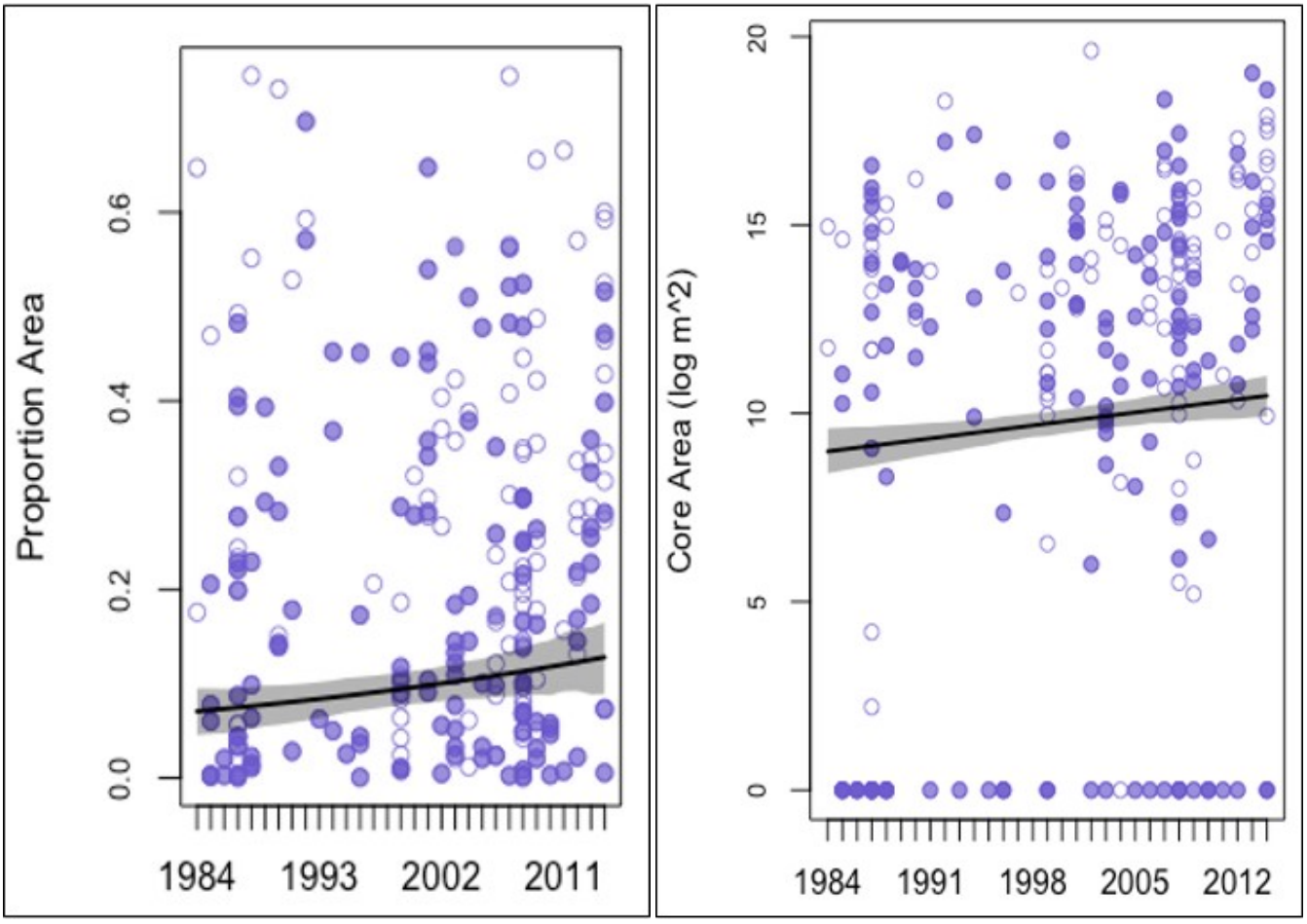


Model predictions

Here are some example figures that show model estimates and predicitons. First here is a dotplot for the proportion high severity model. Estimates are of slopes (i.e. change over time, positive means increasing) for the different forest and bioregion groups, with 95% HPDI error bars.



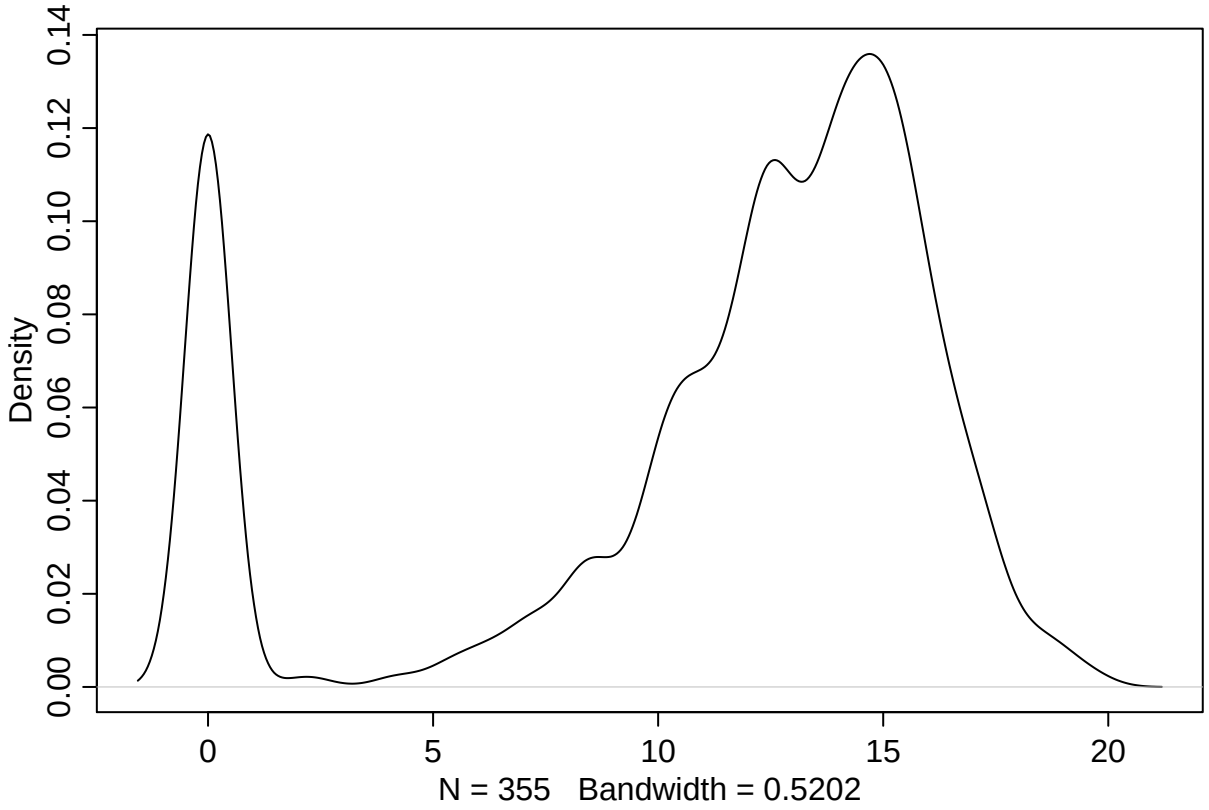
Now here are some predictions. So basically we are zooming in on one bioregion (Sierra) and one forest type (mixed conifer). Circles are individual fires, filled are for the Sierra, open are mixed conifer fires, but in another bioregion. The first figure shows proportion HS change over time. The second shows change in amount of core area changing over time. The latter looks a little weird because of the "zero-inflation". Specifically we have a number of fires without any core area so our mean falls somewhere between those without any core and those with some core.



Looking a bit more at the core area issue. Here's a density plot of these data. You can see that these data are essentially log-normal but that we are running up against zero on the low side, which may cause problems in the modeling.

Question: How do we address the quasi-zero-inflation of the core area data?

```
dens(dd$logcore)
```


Density
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0
5
10
15
20
N = 355 Bandwidth = 0.5202