

fitting1

Yunfeng Yang

2021/3/30

```
load("fittingwithrestrict.RData")
require(rstan)
require(MASS)
require(loo)
```

Model 1, including all the alpha

```
dmg_mod <- stan_model("damage.stan")

init_dmg <- function() {
  list(mu = c(35,8), sigma = c(10,1), rho = .5, alpha_R20 = 1,
       alpha_R40 = 1, alpha_R60 = 1, alpha_T20 = 1, alpha_T40 = 1, alpha_T60 = 1)
}
set.seed(2020)
dmg_fit <- sampling(object = dmg_mod,
                      data = list(N_R20 = nrow(R20_data), N_R40 = nrow(R40_data), N_R60 = nrow(R60_data),
                                  N_T20 = nrow(T20_data), N_T40 = nrow(T40_data), N_T60 = nrow(T60_data),
                                  N_x = length(T100_data), N_y = length(R100_data),
                                  X_R20 = R20_data, X_R40 = R40_data, X_R60 = R60_data,
                                  X_T20 = T20_data, X_T40 = T40_data, X_T60 = T60_data,
                                  t_x = R100_data, t_y = T100_data,
                                  l_R20=R_pf[1], l_R40=R_pf[2], l_R60=R_pf[3],
                                  l_T20=T_pf[1], l_T40=T_pf[2], l_T60=T_pf[3]),
                      control = list(adapt_delta = 0.8), init = init_dmg)

print(dmg_fit, pars = c('mu', 'sigma', 'rho', 'alpha_R20', 'alpha_R40',
                       'alpha_R60', 'alpha_T20', 'alpha_T40', 'alpha_T60'))

## Inference for Stan model: damage.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean   se_mean     sd    2.5%    25%    50%    75%   97.5% n_eff Rhat
## mu[1]      45.15     0.01   0.59  43.99  44.74  45.15  45.55  46.30   3043     1
## mu[2]       5.51     0.00   0.05   5.41   5.48   5.51   5.55   5.61   2839     1
## sigma[1]    12.91     0.01   0.42  12.11  12.63  12.90  13.19  13.76   3382     1
## sigma[2]     1.08     0.00   0.04   1.01   1.05   1.08   1.11   1.16   2849     1
## rho         0.80     0.00   0.05   0.70   0.77   0.80   0.83   0.88   2233     1
```

```

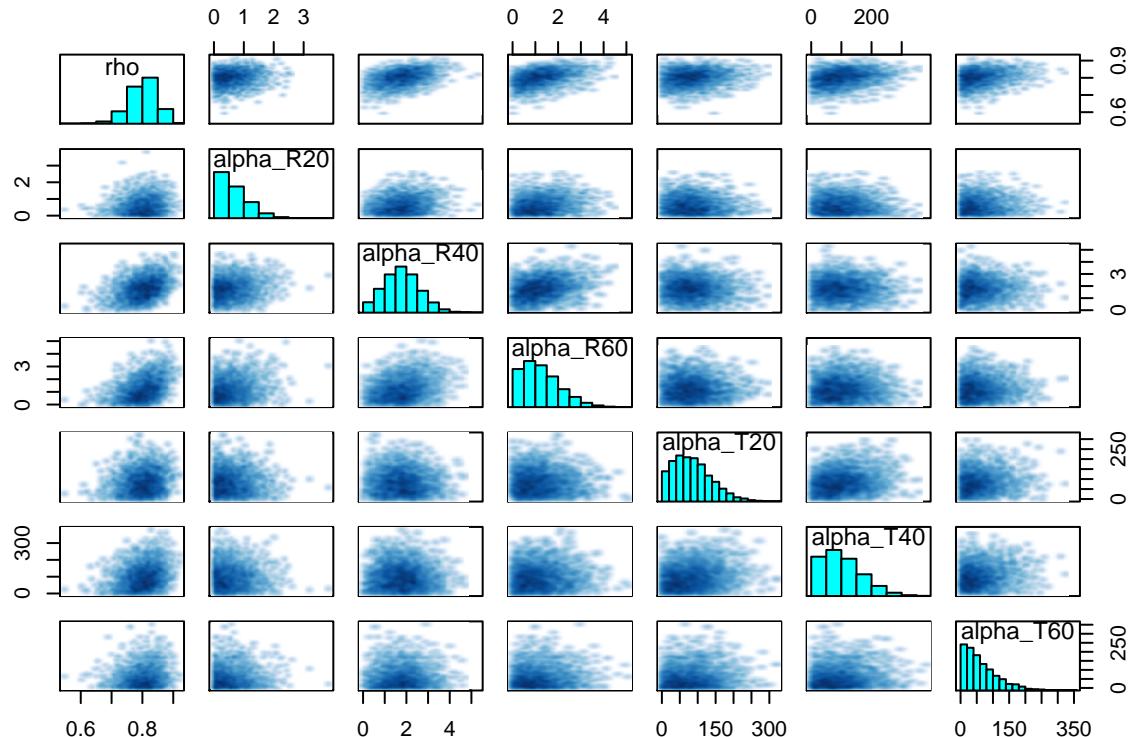
## alpha_R20 0.65 0.01 0.48 0.03 0.27 0.55 0.95 1.78 3229 1
## alpha_R40 1.79 0.02 0.82 0.31 1.19 1.78 2.33 3.47 2014 1
## alpha_R60 1.26 0.02 0.83 0.07 0.61 1.13 1.76 3.15 2247 1
## alpha_T20 85.69 1.03 54.08 5.14 43.71 79.33 118.49 208.21 2782 1
## alpha_T40 102.71 1.30 66.36 6.40 50.53 93.47 143.47 253.22 2602 1
## alpha_T60 64.76 0.91 53.35 2.19 23.62 51.42 92.40 196.12 3415 1
##
## Samples were drawn using NUTS(diag_e) at Tue Mar 30 12:04:11 2021.
## 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).

```

```

pairs(dmg_fit,pars = c('rho','alpha_R20','alpha_R40',
'alpha_R60','alpha_T20','alpha_T40','alpha_T60'))

```



```

loo_dmg <- loo(dmg_fit)

```

Model 2, without any alpha

```

nondmg_mod <- stan_model("nondamage.stan")
init_nondmg <- function() {
  list(mu = c(35,8), sigma = c(10,1), rho = .5)
}

```

```

}

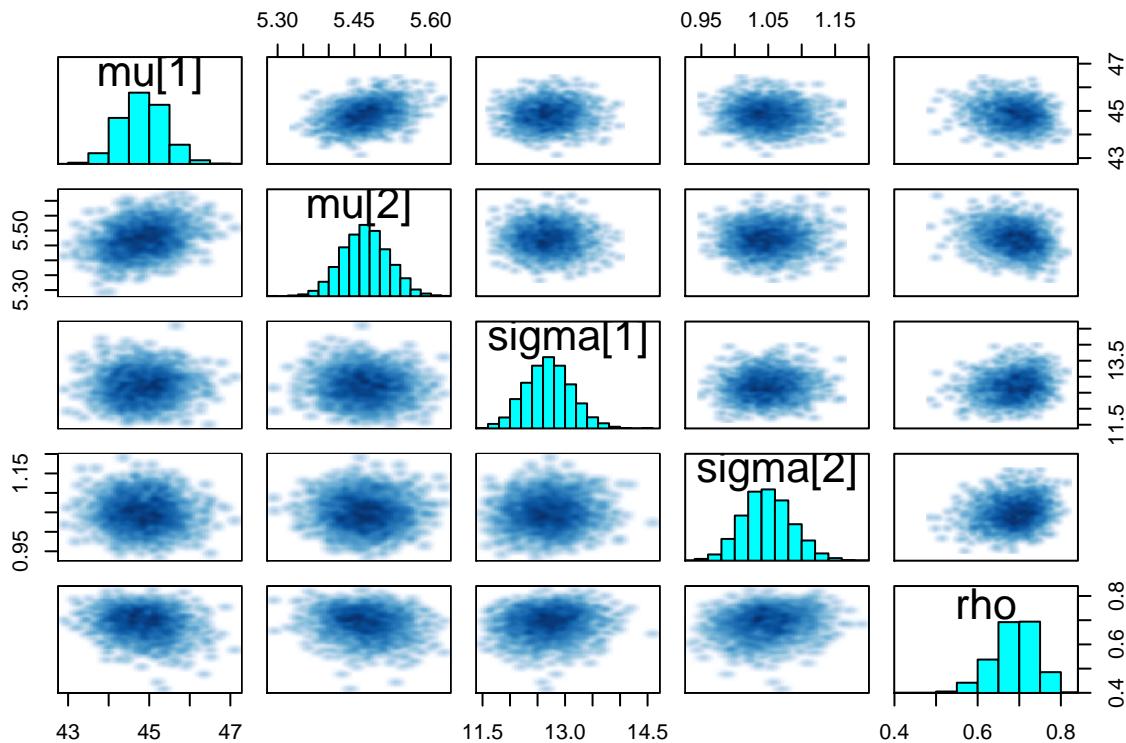
nondmg_fit <- sampling(object = nondmg_mod,
                        data = list(N_R20 = nrow(R20_data), N_R40 = nrow(R40_data), N_R60 = nrow(R60_da
                                         N_T20 = nrow(T20_data), N_T40 = nrow(T40_data), N_T60 = nrow(T60_da
                                         N_x = length(T100_data), N_y = length(R100_data),
                                         X_R20 = R20_data, X_R40 = R40_data, X_R60 = R60_data,
                                         X_T20 = T20_data, X_T40 = T40_data, X_T60 = T60_data,
                                         t_x = R100_data, t_y = T100_data,
                                         l_R20=R_pf[1],l_R40=R_pf[2],l_R60=R_pf[3],
                                         l_T20=T_pf[1],l_T40=T_pf[2],l_T60=T_pf[3]),
                                         control = list(adapt_delta = 0.8),init = init_nondmg)

print(nondmg_fit,pars = c('mu','sigma','rho'))

## Inference for Stan model: nondamage.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean    sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## mu[1]    44.84    0.01 0.55 43.82 44.46 44.84 45.21 45.92  4053     1
## mu[2]     5.47    0.00 0.05  5.38  5.44  5.47  5.50  5.56  4036     1
## sigma[1] 12.70    0.01 0.42 11.91 12.41 12.69 12.98 13.55  4082     1
## sigma[2]   1.05    0.00 0.04  0.98  1.02  1.05  1.07  1.12  3766     1
## rho       0.69    0.00 0.05  0.58  0.66  0.69  0.72  0.77  4333     1
##
## Samples were drawn using NUTS(diag_e) at Tue Mar 30 12:14:18 2021.
## 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).

pairs(nondmg_fit,pars = c('mu','sigma','rho'))

```



```
# LOOIC

loo_nondamage <- loo(nondmg_fit)

## LOOIC comparison between damage model and nondamage model
## The preferred model will be at the first row.
loo_compare(loo_dmg, loo_nondamage)
```

```
##          elpd_diff se_diff
## model2    0.0      0.0
## model1 -3.1     2.2
```

Model 3, only including alpha_R40

```
R40dmg_mod <- stan_model("only_alphaR40.stan")
init_R40dmg <- function() {
  list(mu = c(35,8), sigma = c(10,1), rho = .5, alpha_R40 = 1)
}

R40dmg_fit <- sampling(object = R40dmg_mod,
                         data = list(N_R20 = nrow(R20_data), N_R40 = nrow(R40_data), N_R60 = nrow(R60_data))
```

```

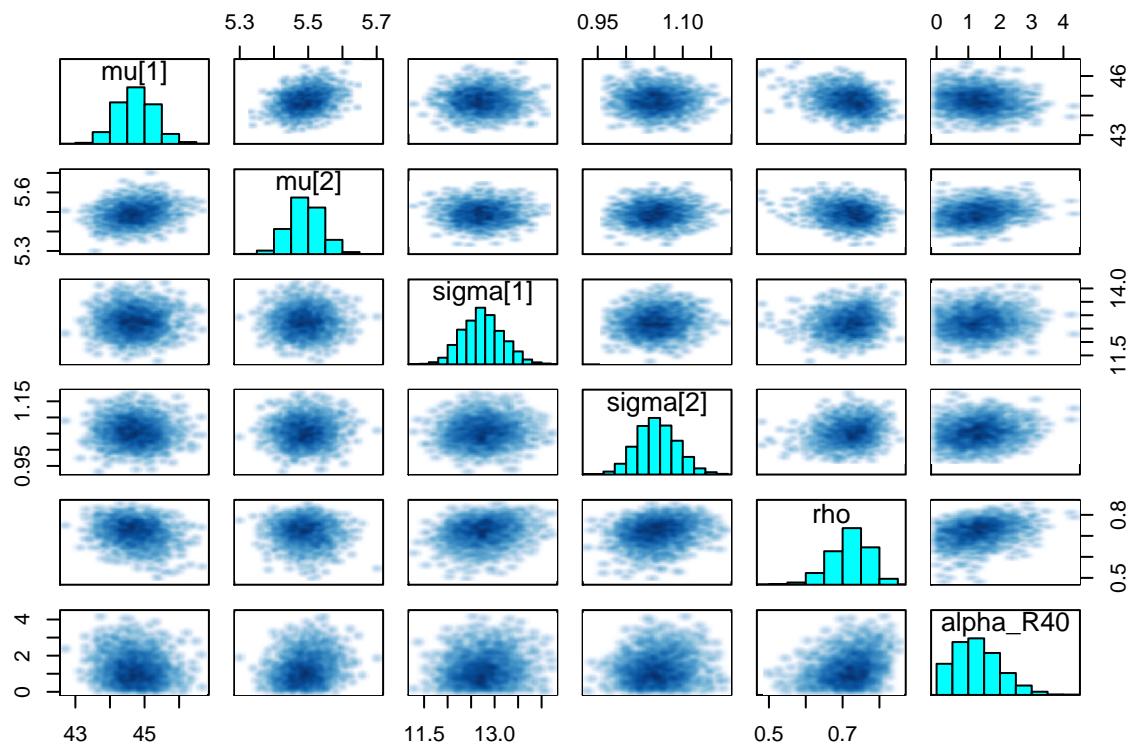
N_T20 = nrow(T20_data), N_T40 = nrow(T40_data), N_T60 = nrow(T60_data),
N_x = length(T100_data), N_y = length(R100_data),
X_R20 = R20_data, X_R40 = R40_data, X_R60 = R60_data,
X_T20 = T20_data, X_T40 = T40_data, X_T60 = T60_data,
t_x = R100_data, t_y = T100_data,
l_R20=R_pf[1], l_R40=R_pf[2], l_R60=R_pf[3],
l_T20=T_pf[1], l_T40=T_pf[2], l_T60=T_pf[3]),
control = list(adapt_delta = 0.8), init = init_R40dmg)

print(R40dmg_fit, pars = c('mu', 'sigma', 'rho', 'alpha_R40'))

## Inference for Stan model: only_alphaR40.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean    sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
## mu[1]    44.74    0.01 0.53 43.69 44.38 44.72 45.10 45.82  3877     1
## mu[2]     5.49    0.00 0.05  5.40  5.46  5.49  5.52  5.59  3900     1
## sigma[1]  12.74    0.01 0.43 11.94 12.45 12.73 13.03 13.61  3966     1
## sigma[2]   1.05    0.00 0.04  0.99  1.03  1.05  1.08  1.13  4146     1
## rho       0.72    0.00 0.05  0.61  0.69  0.72  0.76  0.81  3622     1
## alpha_R40 1.28    0.01 0.72  0.13  0.74  1.23  1.74  2.89  3313     1
##
## Samples were drawn using NUTS(diag_e) at Tue Mar 30 12:17:25 2021.
## 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).

pairs(R40dmg_fit, pars = c('mu', 'sigma', 'rho', 'alpha_R40'))

```



```
# LOOIC

loo_R40dmg <- loo(R40dmg_fit)
extract<- extract(R40dmg_fit)$'alpha_R40'
##
loo_compare(loo_R40dmg, loo_nondamage)

##          elpd_diff se_diff
## model1    0.0      0.0
## model2   -0.2     1.7
```