Meta Analysis_Two Approaches

Subbiah

17 January 2018

AIM

To explain Meta analytic approach in random effects model (REM) framework using 1. summary statistics (SSM) and 2. Fully Bayesian (FBM) method.

Ref: Meta-analysis: formulating, evaluating, combining, and reporting Normand,ST Statist. Med. 1999

REM here indicates combining multiple (k) studies each with a study effect y_i where $i = 1, 2, 3.....k$

AIM 1 - SSM:

Underlying model is $y_i \sim N(\mu_i, v_i^2)$ $\mu_i \sim N(d, \sigma_\mu^2)$. Assumption is v_i^2 are known (estimated from data, most of the cases asymptotic variance) quantities of interest are μ and σ_μ^2.

Step 1:Summary Statistics

We use metafor pacakge to estimate the summary measure y_i and asymptotic variance v_i^2

In this note y_i is log odds ratio of study i $\forall 1 \leq i \leq k$ and hence $v_i^2 = a^{-1} + b^{-1} + c^{-1} + d^{-1}$ where a, b, c, d are cell counts of given 2×2 table

```
#Summary Statistics Approach
#Step 1
ami=c(0,    0,  0,  0,  1,  0,  0,  2,  0,  0,  0,  0,  0,  1,  0,  0,  1,  1,  0,  0,  0)
bmi=c(100,  100,    28, 14, 19, 80, 54, 148,    60, 15, 10, 15, 88, 141,    136,    204,    97, 23, 25,
cmi=c(0,    0,  1,  0,  1,  0,  0,  0,  2,  0,  0,  1,  2,  2,  0,  2,  2,  1,  0,  0,  0)
dmi=c(100,  101,    36, 20, 18, 80, 49, 150,    58, 16, 10, 19, 86, 137,    131,    182,    97, 25, 25,
require(metafor)
y=escalc(ai=ami,bi=bmi,ci=cmi,di=dmi,measure = "OR",add = 10^(-8), to="only0" ,  drop00=FALSE)
#End of step 1-------------------------------------------------------------
```

Step 2: Stan code for REM Normal-Normal Model

```
data {
  int<lower=0> N;
  real y[N];
  real<lower=0> v1[N];
}
parameters {
  real d;
  real<lower=0> sigma_mu;
  vector[N] mu;
}
transformed parameters {
```

```
  real<lower=0> sigmasq_mu;
  sigmasq_mu=sigma_mu*sigma_mu;
}
model {
  y ~ normal(mu, v1);
  mu ~ normal(d, sigma_mu);
  d ~ normal(0, 1E3);
  sigmasq_mu ~ inv_gamma(1E-3, 1E-3);
}
```

```
require(rstan)
```

```
## Loading required package: rstan
```

```
## Loading required package: ggplot2
```

```
## Loading required package: StanHeaders
```

```
## rstan (Version 2.17.2, GitRev: 2e1f913d3ca3)
```

```
## For execution on a local, multicore CPU with excess RAM we recommend calling
## options(mc.cores = parallel::detectCores()).
## To avoid recompilation of unchanged Stan programs, we recommend calling
## rstan_options(auto_write = TRUE)
```

```
yi=as.vector(y$yi);v1=sqrt(y$vi) #sq.root is for SD parameter in Normal in Stan
N=length(ami)
MA_SS_data <- list(y=yi,v=v1,N=N)
MA_SS_init <- function(){list(
  d = 0,
  mu = rnorm(N,0,1),
  sigma_mu = 1
)}
mod1 <- sampling(MA_SS_tbyt,data=MA_SS_data, init = MA_SS_init,
                 control = list(adapt_delta=0.99,max_treedepth=15),iter=2000,chains=4)
```

```
##
## SAMPLING FOR MODEL '9619ccf3b9cf03e6a4a776ff2a33354f' NOW (CHAIN 1).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
```

```
##  Elapsed Time: 0.737 seconds (Warm-up)
##                1.048 seconds (Sampling)
##                1.785 seconds (Total)
##
##
## SAMPLING FOR MODEL '9619ccf3b9cf03e6a4a776ff2a33354f' NOW (CHAIN 2).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.501 seconds (Warm-up)
##                1.129 seconds (Sampling)
##                1.63 seconds (Total)
##
##
## SAMPLING FOR MODEL '9619ccf3b9cf03e6a4a776ff2a33354f' NOW (CHAIN 3).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.846 seconds (Warm-up)
##                0.531 seconds (Sampling)
##                1.377 seconds (Total)
##
```

```
##
## SAMPLING FOR MODEL '9619ccf3b9cf03e6a4a776ff2a33354f' NOW (CHAIN 4).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 0.768 seconds (Warm-up)
##                0.484 seconds (Sampling)
##                1.252 seconds (Total)

## Warning: There were 54 divergent transitions after warmup. Increasing adapt_delta above 0.99 may hel
## http://mc-stan.org/misc/warnings.html#divergent-transitions-after-warmup

## Warning: There were 4 chains where the estimated Bayesian Fraction of Missing Information was low. S
## http://mc-stan.org/misc/warnings.html#bfmi-low

## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
fitNor_Nor_summary<-as.data.frame(summary(mod1, pars = c("d","sigmasq_mu"), probs = c(0.025, 0.975))$su
ss_re=round(fitNor_Nor_summary,4)
```

AIM 2 - FBM:

Additional Reference:Smith et al 1995 Bayesian Approaches to Random-Effects Meta analysis: A Comparative Study

Underlying model is

$r_i^C \sim B(p_i^C, n_i^C)$

$r_i^T \sim B(p_i^T, n_i^T)$

$\text{logit}(p_i^C)$=μ_i-$\delta_i/2$

$\text{logit}(p_i^T)$=$\mu_i + \delta_i/2$

$\delta_i \sim N(d, \sigma^2)$

$\mu_i \sim N(d_0, \sigma_0^2)$

Here, r_i^C indicates the number of successes in Control group arising from n_i^C cases assumed to have probability of p_i^C Similarly r_i^T can be defined for treatment group Quantities of interest are d and σ^2.

Fully Bayesian Method

Stan Model

```
data {
  int<lower=0> N;
  int<lower=0> nt[N];
  int<lower=0> rt[N];
  int<lower=0> nc[N];
  int<lower=0> rc[N];
}
parameters {
  real d;
  real<lower=0> sigma_delta;
  vector[N] mu;
  vector[N] delta;
}
transformed parameters {
  real<lower=0> sigmasq_delta;
  sigmasq_delta = sigma_delta*sigma_delta;

}
model {
  rt ~ binomial_logit(nt, mu+delta/2);
  rc ~ binomial_logit(nc, mu-delta/2);
  delta  ~ normal(d, sigma_delta);
  mu ~ normal(0, 2);
  d ~ normal(0, sqrt(10));
  sigmasq_delta ~ inv_gamma(3, 1);
}
```

```
require(rstan)
nt=ami+bmi;nc=cmi+dmi
N=length(ami)

MA_FB_data <- list(nc=nc,rc=cmi,nt=nt,rt=ami,N=N)
MA_FB_init <- function(){list(
  d = 0,
  mu = rnorm(N,0,1),
  delta= rnorm(N,0,1),
  sigma_delta = 1
)}
fitMA_FB <- stan(file="MA_FB_tbyt.stan",
                 data=MA_FB_data, init = MA_FB_init,
                 control = list(adapt_delta=0.99,max_treedepth=15),iter=2000,chains=4)
```

```
## In file included from C:/Users/Lapilluz2/Documents/R/win-library/3.4/BH/include/boost/config.hpp:39:0
##                  from C:/Users/Lapilluz2/Documents/R/win-library/3.4/BH/include/boost/math/tools/config.hpp:13,
##                  from C:/Users/Lapilluz2/Documents/R/win-library/3.4/StanHeaders/include/stan/math/rev/core.hpp:12,
##                  from C:/Users/Lapilluz2/Documents/R/win-library/3.4/StanHeaders/include/stan/math/rev/mat.hpp:4,
##                  from C:/Users/Lapilluz2/Documents/R/win-library/3.4/StanHeaders/include/stan/math/rev/scal.hpp:4,
##                  from C:/Users/Lapilluz2/Documents/R/win-library/3.4/StanHeaders/include/stan/math.hpp:19,
##                  from C:/Users/Lapilluz2/Documents/R/win-library/3.4/StanHeaders/include/src/stan/model/model_header.hpp:4,
##                  from file13b827153ee7.cpp:8:
```

```
## C:/Users/Lapilluz2/Documents/R/win-library/3.4/BH/include/boost/config/compiler/gcc.hpp:186:0: warnin
##  #  define BOOST_NO_CXX11_RVALUE_REFERENCES
##  ^
## <command-line>:0:0: note: this is the location of the previous definition
## cc1plus.exe: warning: unrecognized command line option "-Wno-ignored-attributes"
##
## SAMPLING FOR MODEL 'MA_FB_tbyt' NOW (CHAIN 1).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 11.718 seconds (Warm-up)
##                11.513 seconds (Sampling)
##                23.231 seconds (Total)
##
##
## SAMPLING FOR MODEL 'MA_FB_tbyt' NOW (CHAIN 2).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 11.349 seconds (Warm-up)
##                12.962 seconds (Sampling)
##                24.311 seconds (Total)
```

```
##
##
## SAMPLING FOR MODEL 'MA_FB_tbyt' NOW (CHAIN 3).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 14.504 seconds (Warm-up)
##                9.143 seconds (Sampling)
##                23.647 seconds (Total)
##
##
## SAMPLING FOR MODEL 'MA_FB_tbyt' NOW (CHAIN 4).
##
## Gradient evaluation took 0 seconds
## 1000 transitions using 10 leapfrog steps per transition would take 0 seconds.
## Adjust your expectations accordingly!
##
##
## Iteration:    1 / 2000 [  0%]  (Warmup)
## Iteration:  200 / 2000 [ 10%]  (Warmup)
## Iteration:  400 / 2000 [ 20%]  (Warmup)
## Iteration:  600 / 2000 [ 30%]  (Warmup)
## Iteration:  800 / 2000 [ 40%]  (Warmup)
## Iteration: 1000 / 2000 [ 50%]  (Warmup)
## Iteration: 1001 / 2000 [ 50%]  (Sampling)
## Iteration: 1200 / 2000 [ 60%]  (Sampling)
## Iteration: 1400 / 2000 [ 70%]  (Sampling)
## Iteration: 1600 / 2000 [ 80%]  (Sampling)
## Iteration: 1800 / 2000 [ 90%]  (Sampling)
## Iteration: 2000 / 2000 [100%]  (Sampling)
##
##  Elapsed Time: 10.414 seconds (Warm-up)
##                11 seconds (Sampling)
##                21.414 seconds (Total)

## Warning: There were 4 chains where the estimated Bayesian Fraction of Missing Information was low. Se
## http://mc-stan.org/misc/warnings.html#bfmi-low

## Warning: Examine the pairs() plot to diagnose sampling problems
```

```
fitMA_FB_summary<-as.data.frame(summary(fitMA_FB, pars = c("d","sigmasq_delta"), probs = c(0.025, 0.975)
fb_re=round(fitMA_FB_summary,4)
```

```
library(knitr)
kable(ss_re,caption = "Summary Statistics Method")
```

Table 1: Summary Statistics Method

	mean	se_mean	sd	2.5%	97.5%	n_eff	Rhat
d	-0.2680	0.0844	0.6492	-1.5606	0.8637	59.2360	1.0275
sigmasq_mu	0.0891	0.0740	0.4406	0.0006	0.7186	35.4513	1.0772

```
kable(fb_re,caption = "Fully Bayesian Method")
```

Table 2: Fully Bayesian Method

	mean	se_mean	sd	2.5%	97.5%	n_eff	Rhat
d	-0.6331	0.0256	0.4015	-1.4392	0.1176	245.5072	1.0197
sigmasq_delta	0.0394	0.0121	0.1691	0.0004	0.3188	195.5637	1.0146

This demonstrates the two MA approaches. Summary statistics requires continuity corrections for zero cells, which requires a careful investigation; where as this is completely alleviated when FBM is used.

Still a careful specification of priors (especially for between variance) is always a concern in Bayesian analysis