Bayesian Thurstonian models for ranking data using JAGS

Eric Scuccimarra (eas7@illinois.edu)

11/04/2020

Bayesian Thurstonian models for ranking data using JAGS

Implementation of examples in paper with updated code.

Example 1

The first example is for data which compares ranked compensation plans against some cultural attitudes.

We are using three chains instead of the one as in the original paper.

Note that function make inits has been updated in order to create overdispersed starting points.

```
source("data.R")
data <- with(compensation, {
    rankings <- cbind(y1, y2, y3, y4)
    data <- list(y = rankings, x = cbind(1, hi, hc, vi, vc), N = 116, P = 4, K = 4)
})

init <- list(list(z = makeinits(data$y, loc = 0)),
         list(z = makeinits(data$y, loc = 5)),
         list(z = makeinits(data$y, loc = -5)))

jags <- jags.model("thurstone-b.txt", data, init, n.chains = 3, n.adapt = 1000)
```

```
## Compiling data graph
##    Resolving undeclared variables
##    Allocating nodes
##    Initializing
##    Reading data back into data table
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 348
##    Unobserved stochastic nodes: 253
##    Total graph size: 4575
##
## Initializing model
```

```
## Warning in jags.model("thurstone-b.txt", data, init, n.chains = 3, n.adapt
## = 1000): Adaptation incomplete
```

```
update(jags, 15000)
samp <- coda.samples(jags, c("beta"), 1e+4)
```

```
## NOTE: Stopping adaptation
```

Note that adaptation was not complete. We are ignoring this as it may be related to the changes we have made to the model code.

Check for convergence - note that some values in beta are artificially forced to 0 making it difficult to know how much these results can be relied on. Also note that sigma (not monitored here) has a high $\hat{R}$:

```
gelman.diag(samp, multivariate = FALSE)
```

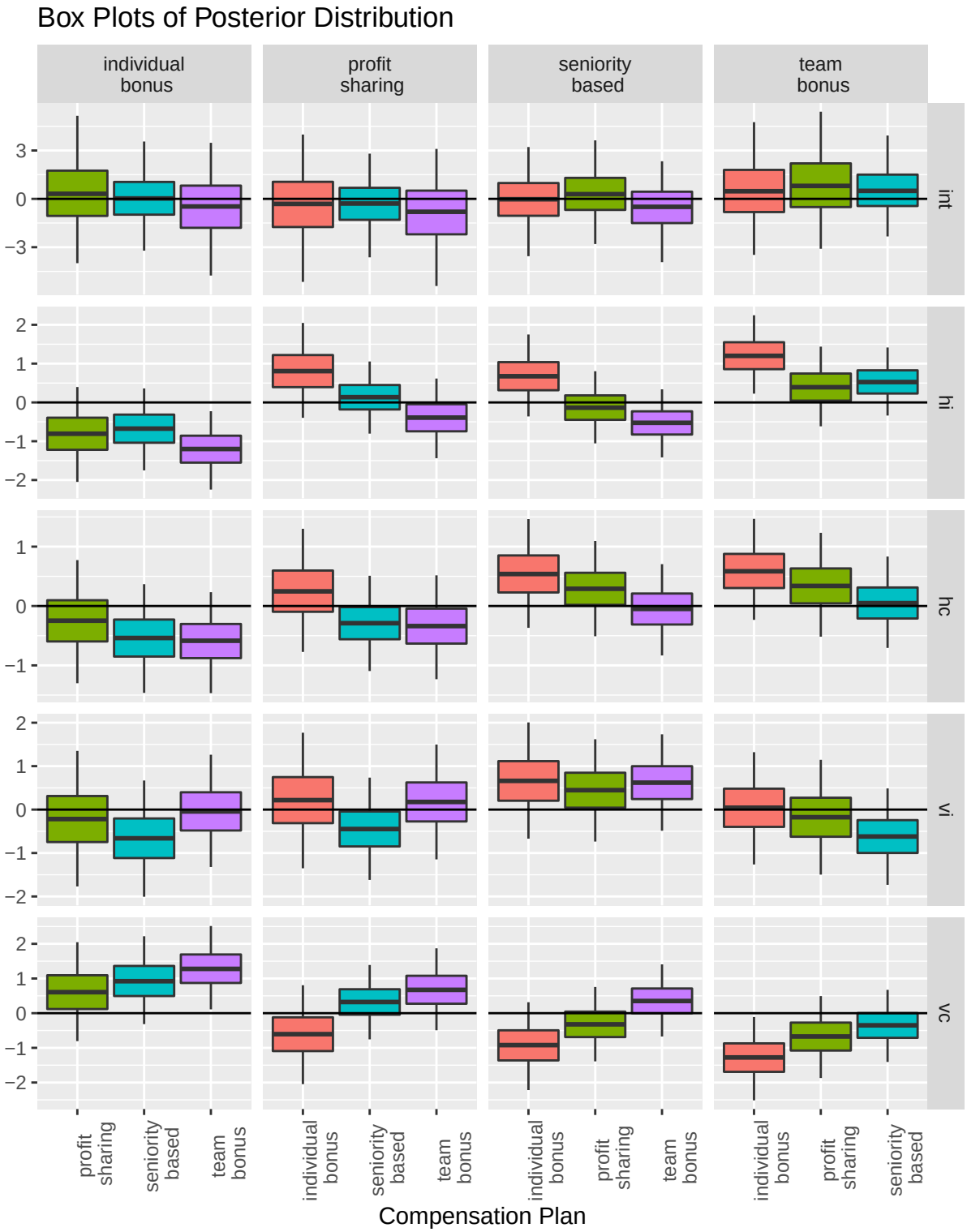
```
## Potential scale reduction factors:
##
##           Point est. Upper C.I.
## beta[1,1]       1.01       1.02
## beta[2,1]       1.25       1.78
## beta[3,1]       1.12       1.34
## beta[4,1]       1.05       1.08
## beta[5,1]       1.10       1.27
## beta[1,2]       1.01       1.02
## beta[2,2]       1.17       1.51
## beta[3,2]       1.07       1.10
## beta[4,2]       1.07       1.16
## beta[5,2]       1.14       1.41
## beta[1,3]       1.01       1.01
## beta[2,3]       1.06       1.07
## beta[3,3]       1.08       1.12
## beta[4,3]       1.08       1.16
## beta[5,3]       1.06       1.08
## beta[1,4]       1.00       1.00
## beta[2,4]       1.00       1.00
## beta[3,4]       1.00       1.00
## beta[4,4]       1.00       1.00
## beta[5,4]       1.00       1.00
```

```
samp <- coda.samples(jags, c("beta.identified", "sigma.identified"), 5e+4)
```

Plot the box plots for `beta.identified` :

```
m <- as.matrix(samp)

df <- as.data.frame(m[, 'beta.identified[1,1,2]'])
names(df) <- c('beta.identified[1,1,2]')
for(i in 1:5){
  for(j in 1:4){

    for(k in 1:4){
      if(j != k){
        col.name <- paste0('beta.identified[', i, ',', j, ',', k,']')
        df[, col.name] <- m[, col.name]
      }
    }
  }
}

df_m = melt(df, id.vars = NULL)
ks = c('individual\n bonus', 'team\n bonus', 'profit\n sharing', 'seniority\n based')
ps = c("int", "hi", "hc", "vi", "vc")

df_m[, 'Compensation_Plan'] = NA
df_m[, 'k1'] = NA
df_m[, 'p'] = NA
```

```r
for(i in 1:5){
  for(j in 1:4){
    for(k in 1:4){
      col.name <- paste0('beta.identified[', i, ',', j, ',', k,']')
      df_m[df_m$variable == col.name, 'Compensation_Plan'] = ks[j]
      df_m[df_m$variable == col.name, 'k1'] = ks[k]
      df_m[df_m$variable == col.name, 'p'] = ps[i]
    }
  }
}

df_m$p_f = factor(df_m$p, levels = ps)

f <- function(x) {
  r <- quantile(x, probs = c(0.025, 0.25, 0.5, 0.75, 0.975))
  names(r) <- c("ymin", "lower", "middle", "upper", "ymax")
  r
}

ggplot(df_m, aes(x=Compensation_Plan, y=value, fill=Compensation_Plan)) + stat_summary(fun.data = f, ge
```

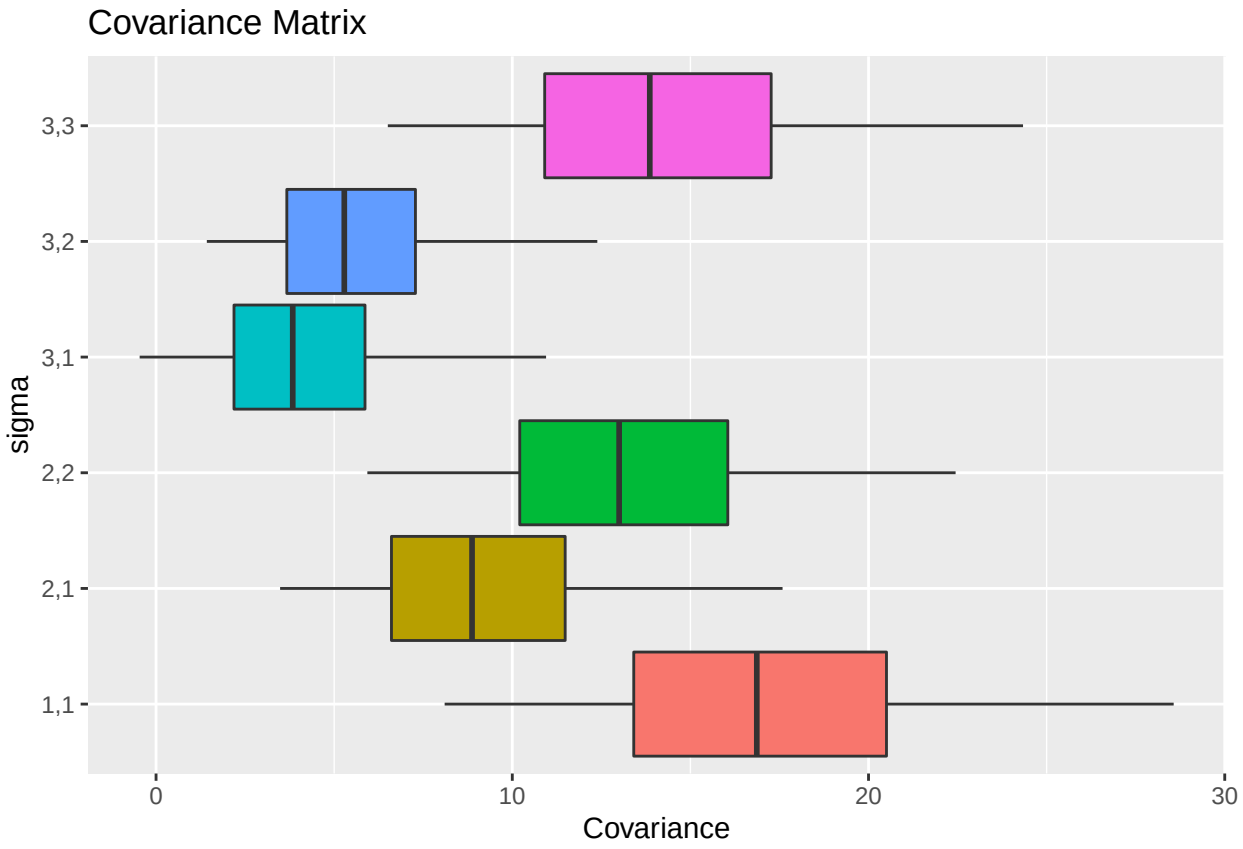


Here we are plotting the posterior distributions for $\hat{\beta}_{jk} - \hat{\beta}_{jk'}$ with each column corresponding to the k'th compensation plan and the box plots within each column corresponding to the kth compensation plan.

Note that the intercepts appear markedly different from the original paper while the rest of the plots seem to be similar.

Plot the box plots for `sigma.identified`. Note that this appears to be on a different scale than in the original paper :

```
df3 <- as.data.frame(m[, paste0("sigma.identified[1,1]")])
df3[, paste0("sigma.identified[", 2, ",", 1:2, "]")] <- m[, paste0("sigma.identified[", 2, ",", 1:2, "]
df3[, paste0("sigma.identified[", 3, ",", 1:3, "]")] <- m[, paste0("sigma.identified[", 3, ",", 1:3, "]
names(df3) <- c("1,1", "2,1", "2,2", "3,1", "3,2", "3,3")

df_m2 <- melt(df3, id.vars = NULL)
names(df_m2) <- c("sigma", "Covariance")

f <- function(x) {
  r <- quantile(x, probs = c(0.025, 0.25, 0.5, 0.75, 0.975))
  names(r) <- c("ymin", "lower", "middle", "upper", "ymax")
  r
}

ggplot(df_m2, aes(x=sigma, y=Covariance, fill=sigma)) + stat_summary(fun.data = f, geom="boxplot") + gg
```



Example 2

This example uses data from a study that examines the effect of experimental manipulation of apparent deception on the rank ordering of hypothetical job applicants. Note that this example runs without modification of the model.

```
data2 <- with(resume, {
  rankings <- cbind(mc.rank, as.rank, sw.rank)
```

```
  xmat <- array(NA, c(73, 4, 3))
  x.mc <- cbind(1, 0, dates.mc, degree.mc); xmat[,,1] <- x.mc
  x.as <- cbind(0, 0, dates.as, degree.as); xmat[,,2] <- x.as
  x.sw <- cbind(0, 1, dates.sw, degree.sw); xmat[,,3] <- x.sw
  data <- list(y = rankings, x = xmat, N = 73, P = 4, K = 3)
})


init <- list(list(z = makeinits(data2$y, loc = 0)),
             list(z = makeinits(data2$y, loc = 2)),
             list(z = makeinits(data2$y, loc = -2)))
jags <- jags.model("thurstone-w.txt", data2, init, n.chains = 3)
```

```
## Compiling data graph
##    Resolving undeclared variables
##    Allocating nodes
##    Initializing
##    Reading data back into data table
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 146
##    Unobserved stochastic nodes: 151
##    Total graph size: 2614
##
## Initializing model
```

```
## Warning in jags.model("thurstone-w.txt", data2, init, n.chains = 3):
## Adaptation incomplete
```

```
update(jags, 5000)
samp <- coda.samples(jags, c("beta"), 3e+4)
```

```
## NOTE: Stopping adaptation
```

Check for convergence :

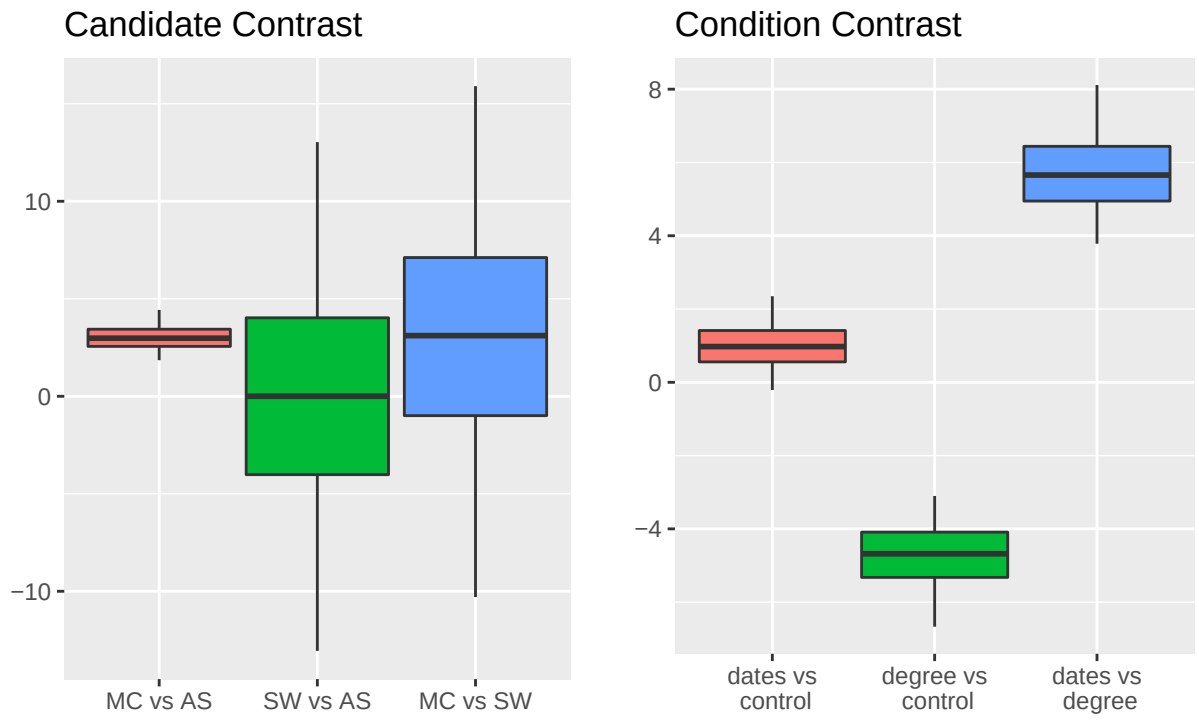
```
gelman.diag(samp)
```

```
## Potential scale reduction factors:
##
##         Point est. Upper C.I.
## beta[1]       1.05       1.16
## beta[2]       1.00       1.00
## beta[3]       1.02       1.05
## beta[4]       1.05       1.16
##
## Multivariate psrf
##
## 1.04
```

Collect sample for posterior :

```
samp <- coda.samples(jags, c("beta.identified", "sigma.identified"), 4e+5)
```

```r
m <- as.matrix(samp)
df <- as.data.frame(m[, c("beta.identified[1]", "beta.identified[2]", "beta.identified[3]", "beta.identi
names(df) <- c("MC vs AS", "SW vs AS", "dates vs\n control", "degree vs\n control")
df[, 'MC vs SW'] <- df[, "MC vs AS"] - df[, 'SW vs AS']
df[, "dates vs\n degree"] <- df[, "dates vs\n control"] - df[, "degree vs\n control"]
df_m <- melt(df, id.vars = NULL)

p1 <- ggplot(df_m[df_m$variable == 'MC vs AS' | df_m$variable == 'SW vs AS' | df_m$variable == 'MC vs S

p2 <- ggplot(df_m[df_m$variable == 'dates vs\n control' | df_m$variable == 'degree vs\n control' | df_m

ggarrange(p1, p2, ncol = 2)
```



Note that the Candidate Contrast differs significantly from the original paper while the Condition Contrast is similar.