Compare Parameterisations

James Totterdell

31/10/2021

Contents


```
library(rstan)
library(posterior)
```

1 Purpose

The simulations use Laplace approximations for approximating the posterior when calculating predictive probability of success.

Recall that the MAP is not invariant to the model parameterisation. Therefore, it is best to parameterise the model such that the parameter posteriors are as close to normal as possible.

This document compares sampling and optimisation when parameterising in terms of outcome level probabilities versus the cut-points in a cumulative logistic model.

2 Model

In what follows, parameterisation 1 refers to specifying the model in terms of outcome level probabilities Parameterisation 2 refers to specifying the model in terms of the intercepts.

For both models, the prior and likelihood is the same, but the parameterisation used for sampling and optimisation differs.

```
tmp1 <- capture.output(
  mod1 <- rstan::stan_model(
    "../clarity2sims/inst/stan/cumulative_logistic_agg_rev.stan"))
tmp2 <- capture.output(
  mod2 <- rstan::stan_model(
    "../clarity2sims/inst/stan/cumulative_logistic_agg_rev_altpar.stan"))
dat <- list(
  N = 3,
  K = 8,
  P = 2,
  y = t(rmultinom(3, 200, c(16, 28, 32, 12, 2, 2, 2, 6) / 100)),
  X = rbind(0, c(1, 0), c(0, 1)),
  prior_counts = 2*rep(1/8, 8),
  prior_sd = rep(1, 2)
)
```

```
ts1 <- system.time(
  fit1 <- sampling(mod1, data = dat, warmup = 500, iter = 5000, refresh = 0))
ts2 <- system.time(
  fit2 <- sampling(mod2, data = dat, warmup = 500, iter = 5000, refresh = 0))
to1 <- system.time(
  opt1 <- optimizing(mod1, data = dat, hessian = TRUE))
to2 <- system.time(
  opt2 <- optimizing(mod2, data = dat, hessian = TRUE))

c("Sampling time param 1" = ts1[3], "Sampling time param 2" = ts2[3])
```

```
## Sampling time param 1.elapsed Sampling time param 2.elapsed
##                         2.923                         5.114
```

```
c("Optimising time param 1" = to1[3], "Optimising time param 2" = to2[3])
```

```
## Optimising time param 1.elapsed Optimising time param 2.elapsed
##                           0.004                           0.053
```

```
M1 <- opt1$par
V1 <- solve(-opt1$hessian)
M2 <- opt2$par
V2 <- solve(-opt2$hessian)

drw1 <- as_draws_matrix(fit1)
drw2 <- as_draws_matrix(fit2)

alpha1 <- drw1[, grepl("alpha", colnames(drw1))]
alpha2 <- drw2[, grepl("alpha", colnames(drw2))]

beta1 <- drw1[, grepl("beta\\[", colnames(drw1))]
beta2 <- drw2[, grepl("beta\\[", colnames(drw2))]
```

```r
par(mfrow = c(3, 3), mar = c(4,4,1,1), cex = 0.75)
for(i in 1:7) {
  plot(density(alpha1[, i]), main = "", xlab = bquote(alpha[.(i)]))
  lines(density(alpha2[,i]), col = "red")
  abline(v = M1[grepl("alpha", names(M1))][i], col = "blue", lty = 2)
  abline(v = M2[grepl("alpha", names(M2))][i], col = "green", lty = 2)
}
```

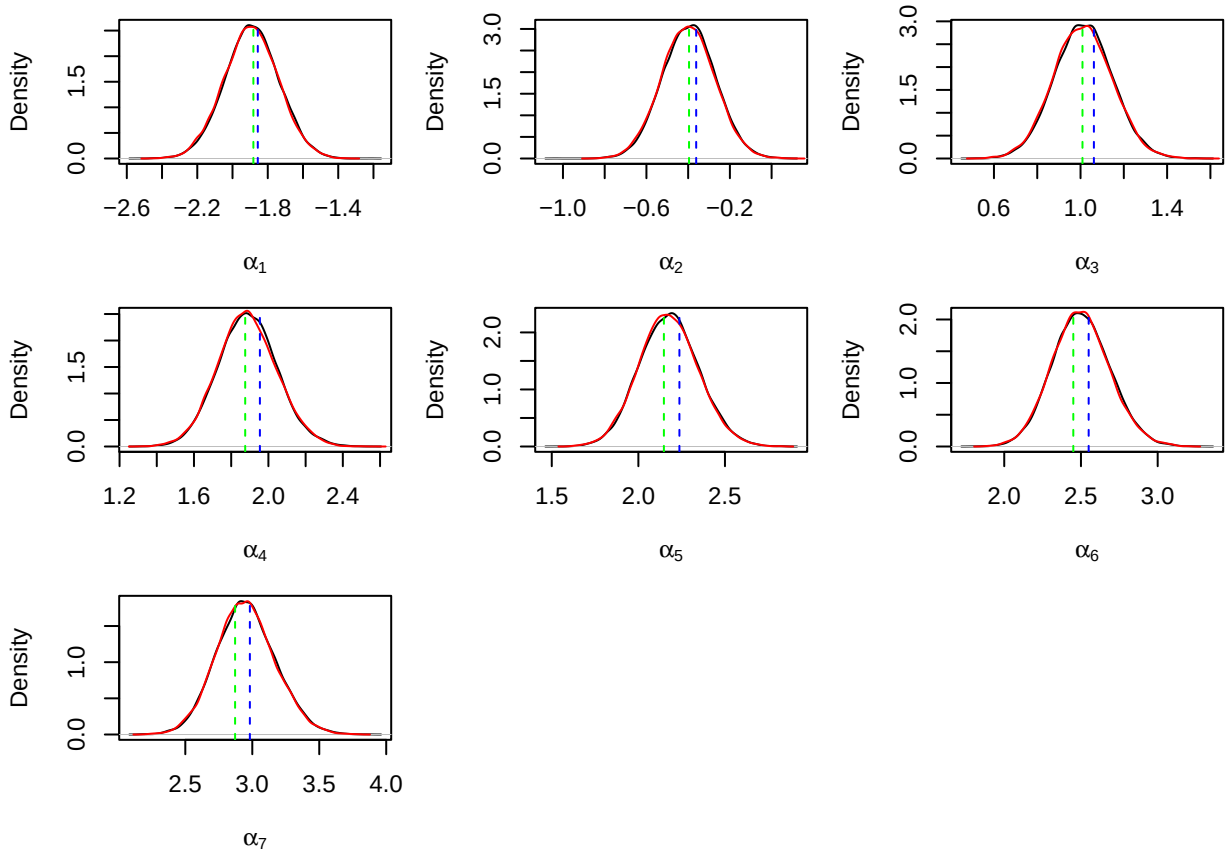

Figure 1: Comparison of posterior for α sampled and computed MAP from both parameterisations.

```r
par(mfrow = c(1, 2), mar = c(4,4,1,1), cex = 0.75)
for(i in 1:2) {
  plot(density(beta1[,i]), main = "", xlab = bquote(beta[.(i)]))
  lines(density(beta2[,i]), col = "red")
  abline(v = M1[grepl("beta\\[", names(M1))][i], col = "blue", lty = 2)
  abline(v = M2[grepl("beta\\[", names(M2))][i], col = "green", lty = 2)
}
```

```r
par(mfrow = c(1, 2), mar = c(4,4,1,1), cex = 0.75)
# Note, don't need to scale to prior as they are Normal(0,1)
for(i in 1:2) {
  plot(density(beta1[, i]), main = "", xlab = bquote(beta[.(i)]))
  curve(
    dnorm(x,
          M1[grepl("beta\\[", names(M1))][i],
          sqrt(V1[grepl("beta", rownames(V1)), grepl("beta", rownames(V1))][i,i])),
```

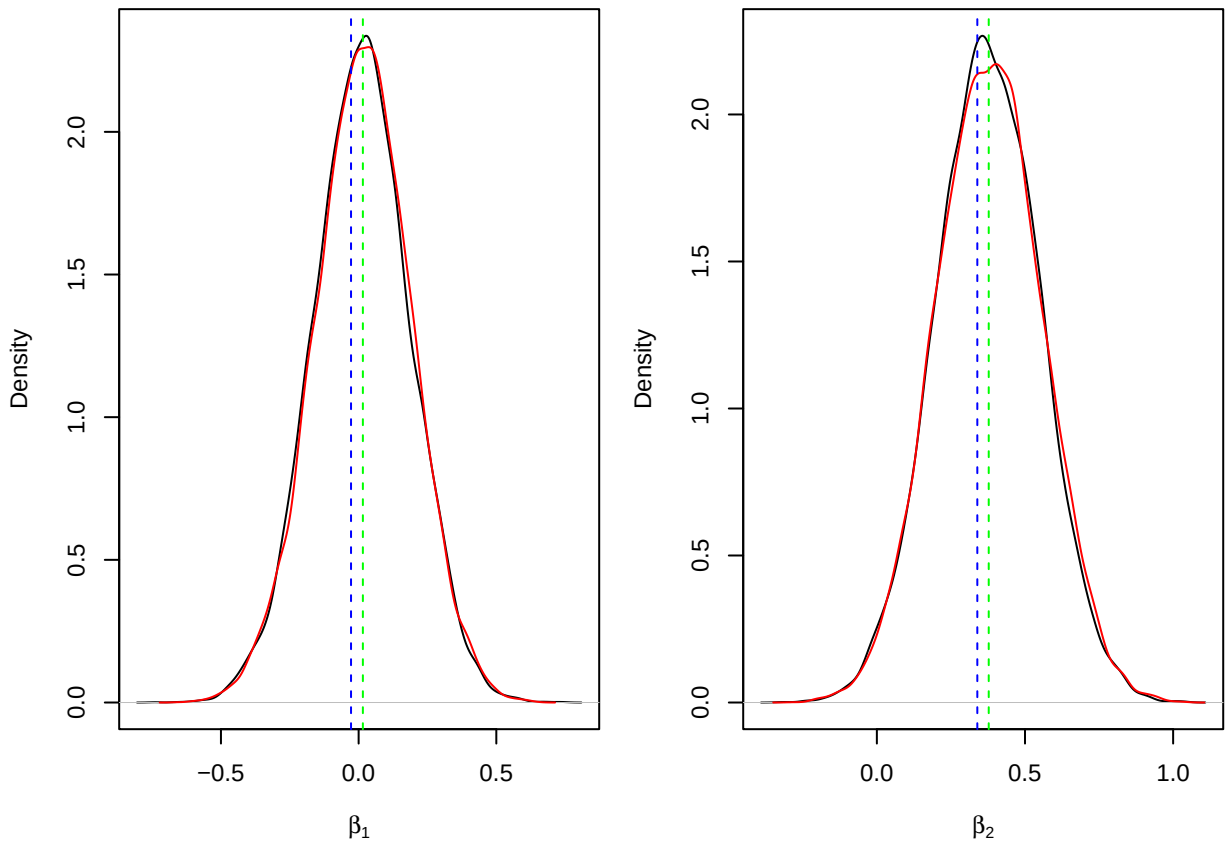

Figure 2: Comparison of posterior for β sampled and computed MAP from both parameterisations.

```
    add = TRUE, col ="blue", lty = 2)
  curve(
    dnorm(x,
          M2[grepl("beta\\[", names(M2))][i],
          sqrt(V2[grepl("beta", rownames(V2)), grepl("beta", rownames(V2))][i,i])),
    add = TRUE, col ="green", lty = 2)
  if (i == 1)
  legend("topright",
         legend = c("MCMC", "Param 1", "Param 2"),
         lty = c(1, 2, 2),
         col = c("black", "blue", "green"), bty = "n")
}
```

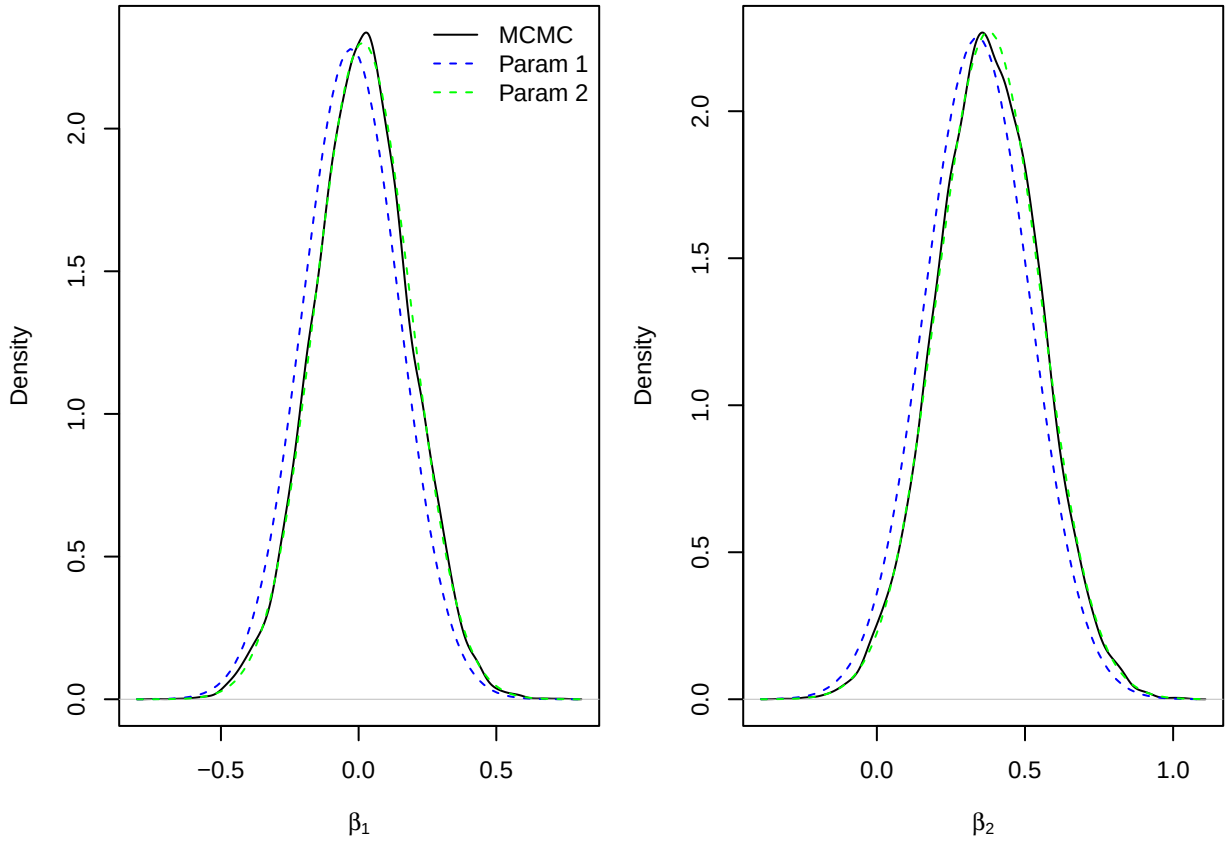

Figure 3: Comparison of Laplace approximation from each parameterisation.

3 Take Away

In the examples I've investigated, parameterisation 1 is invariably more efficient for sampling. However, due to being parameterised on the probability scale, the posterior distributions for the π parameters are further from being Normal compared to parameterisation 2.

For optimisation, efficiency is also better for parameterisation 1, but accuracy of the approximation is better under parameterisation 2 as the intercepts are closer to normal than the probabilities.

Therefore, for sampling, use parameterisation 1, but for approximating the posterior, better to use parameterisation 2, unless speed is of upmost importance, in which case perhaps still use parametersiation 1.