Report 09/15/2022

Zhuoqun Wang

tl; dr

- In this report we show the identifiability issue arised in the "uncentered" parametrization of the mixed-effects model for marker counts through a simple simulation example.
- We tested whether the "centered" parametrization can alleviate the identifiability issue on the same toy example
- The identifiability issue might not affect the hypothesis testing results, but the inference on Ω could be problematic.
- We implemented and tested the sampler that allows distinct element-wise shrinkage in the graphical lasso prior

Model formulation

Add model formulation here

A toy example

Describe simulation setting here

Generate data from LTN

```
source('src/LTN/mixed_effects_shotgun.R')
source('src/LTN/mixed_effects_shotgun_uncentered_glasso.R')
source('src/LTN/utils.R')
library(ape)
library(mvtnorm)
set.seed(1)
# set parameters
N = 200
d = 5
tree = rtree(d + 1)
taxa_abundance = c(1,1,1,1,1,1)
taxa_abundance = taxa_abundance/sum(taxa_abundance)
taxa_marker_len = c(1,1,1,1,1,1)
tlr = function(x, tree){
  yyl = count2y(x,tree)
  return(qlogis(yyl$YL/yyl$Y))
}
tlr_inv = function(x, tree){
```

```
  rt=data.tree::as.Node(tree)
  K=length(tree$tip.label)
  maxLen=max(do.call(c,(lapply(ape::nodepath(tree),length))))
  #nodepath traversal of leaves: pre-order
  nodemat=do.call(rbind,lapply(ape::nodepath(tree),function(x){if(length(x)<maxLen){c(x,rep(rt$name,maxL
  nodemat=apply(nodemat, 1:2, function(x){paste0('n',x)})
  rt$Set(name=1:K,traversal = 'pre-order',filterFun = data.tree::isLeaf)
  rt$Do(function(x) {
    x$leftchild=names(x$children[1])
    x$rightchild=names(x$children[2])
  }, traversal = "pre-order",
  filterFun = data.tree::isNotLeaf)
  lc=stats::na.omit(rt$Get('leftchild'))
  rc=stats::na.omit(rt$Get('rightchild'))
  nam=c(paste0('n',lc),paste0('n',rc),paste0('n',rt$name))
  return(psi2p(x, nam, nodemat))
}
tlr_marker_len = tlr(taxa_marker_len, tree)
Xtest = matrix(c(rep(0, N/2), rep(1, N/2)), ncol = 1)
Xadjust = matrix(c(rep(1,N),rnorm(N * 2)), ncol = 3)
g = 4
refflabel = rep(1:4, N/4)
gam = matrix(rnorm(d*g), ncol = g)
beta1 = matrix(rep(1, d), nrow = 1)
beta2 = matrix(rep(-1:1, d), nrow = 3, byrow = F)
var_error = diag(d)
total_count = 1000
# simulate data
psi_mean = t(apply(Xtest %*% beta1 + Xadjust %*% beta2 + t(gam[,refflabel]), 1, function(x){x + tlr_marl
psi = t(apply(psi_mean, 1, function(x){rmvnorm(1,x,var_error)}))
prob = t(apply(psi, 1, function(x){tlr_inv(x,tree)}))
cnt = t(apply(prob, 1, function(x){rmultinom(1,total_count,x)}))
yyl = apply(cnt,1,function(x){count2y(x,tree)})
Y = do.call(rbind,lapply(yyl,function(x)x$Y))
YL = do.call(rbind,lapply(yyl,function(x)x$YL))
```

Fit the mixed-effects model with uncentered paramtrization to the data

```
lambda_mat = matrix(rep(10, d^2),d,d)
niter = 500
g1 = gibbs_shotgun_lambdamat(N = N,
                    p = d,
                    g = g,
                    YL = YL,
                    Y = Y,
                    tlr_marker_len = tlr_marker_len,
                    Xtest = Xtest,
                    Xadjust = Xadjust,
                    adjust = T,
                    grouplabel = refflabel,
                    niter = niter,
                    reff = T,
                    gprior_m = 100,
                    reffcov = 2,
```

```
                    lambda_fixed = lambda_mat,
                    # lambda_fixed = 10,
                    verbose = F)
```

Mixing & Posterior inference
PJAP & PMAPs

```
burnin = niter/2 + 1
beta1_samples = data.frame(do.call(rbind,g1$BETA1))
PMAP = apply(abs(beta1_samples) > .Machine$double.xmin, 2, mean)
PJAP = sum(apply(abs(beta1_samples) > .Machine$double.xmin, 1, sum) != 0)/nrow(beta1_samples)
cat('PMAP', PMAP,'\n')
```

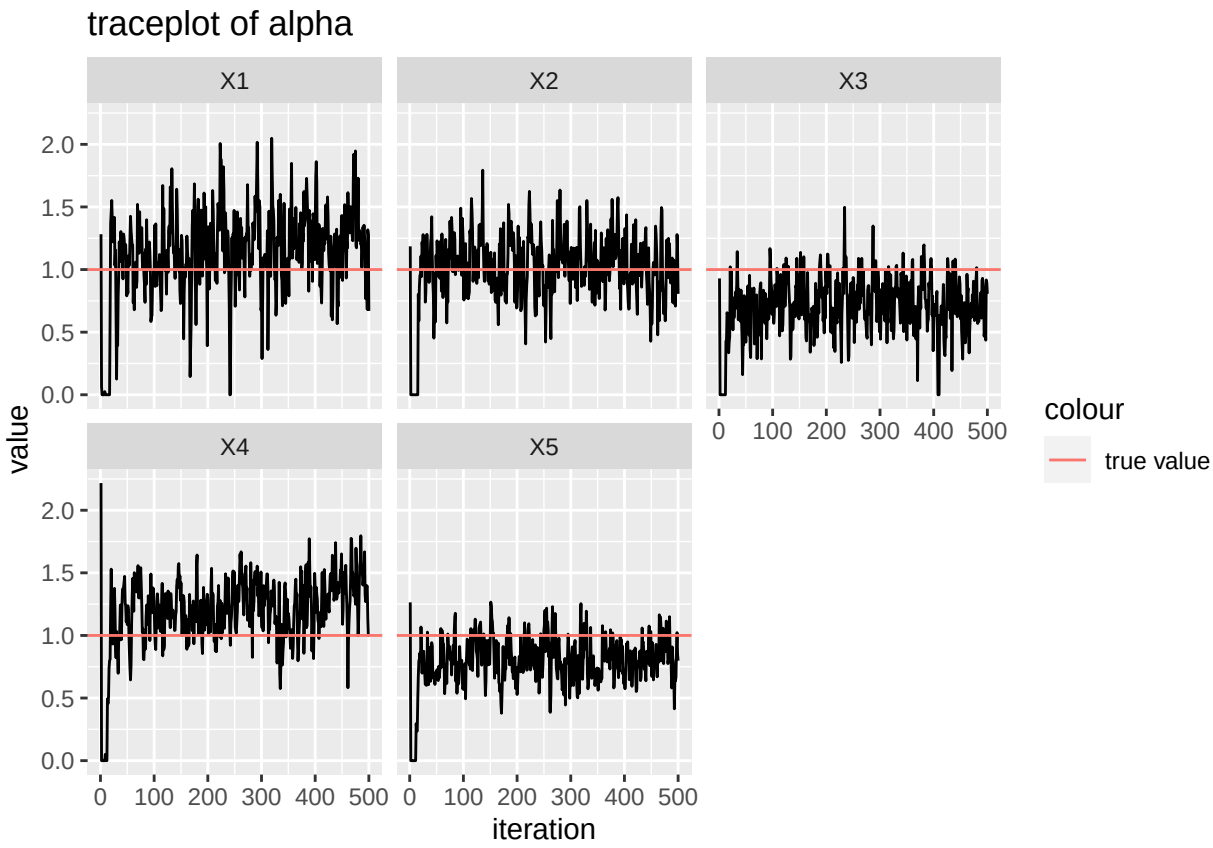
```
## PMAP 0.972 0.972 0.972 0.982 0.98
```

```
cat('PJAP', PJAP,'\n')
```

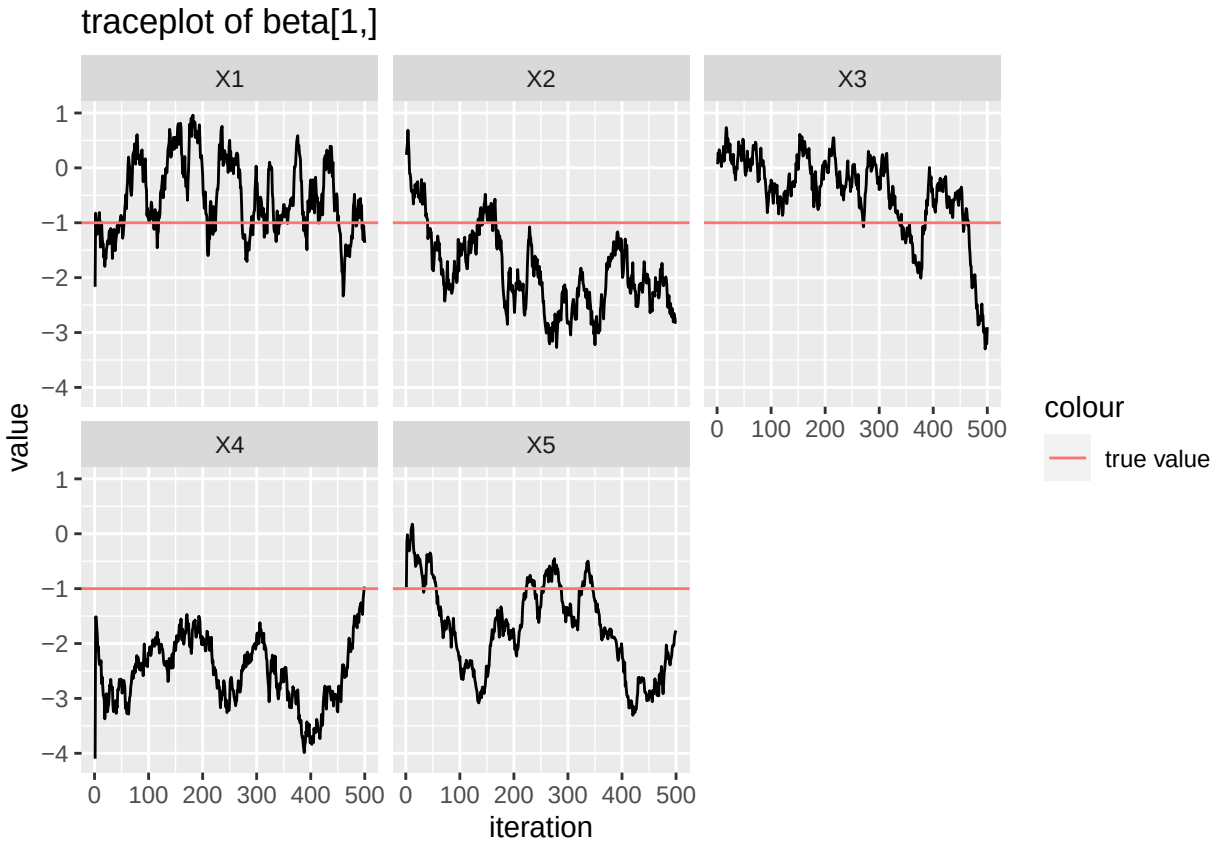
```
## PJAP 0.99
```

traceplots

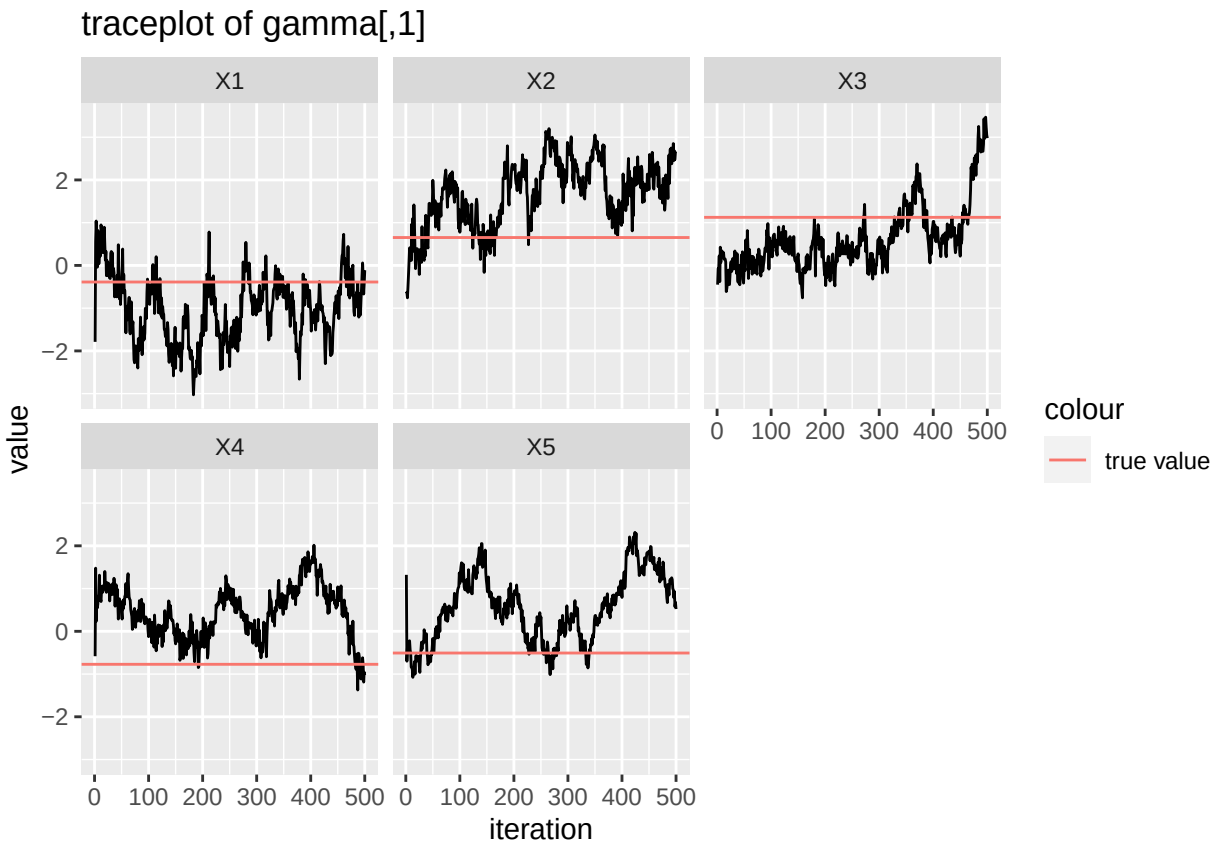
```
library(ggplot2)
library(reshape2)
# alpha
beta1_samples[,'iteration'] = 1:nrow(beta1_samples)
ggplot(melt(beta1_samples, id.vars = 'iteration')) +
geom_line(aes(x = iteration, y = value)) +
facet_wrap(~variable) +
geom_hline(data = data.frame(t(beta1)), aes(yintercept = `t.beta1.`, color = 'true value'))  +
ggtitle('traceplot of alpha')
```



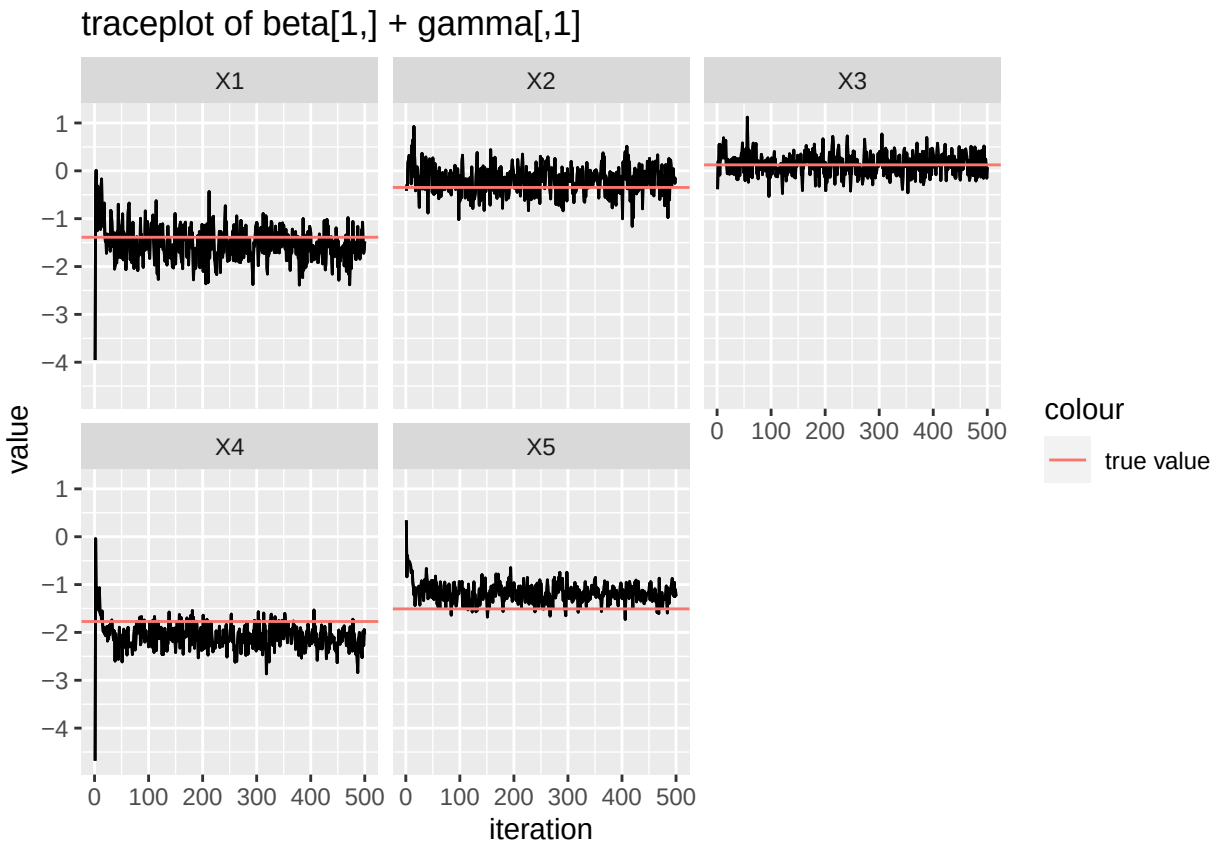

```
# beta intercept
intercept_samples = data.frame(do.call(rbind,lapply(g1$BETA2, function(x){x[1,]})))
intercept_samples[,'iteration'] = 1:nrow(intercept_samples)
ggplot(melt(intercept_samples, id.vars = 'iteration')) +
geom_line(aes(x = iteration, y = value)) +
facet_wrap(~variable) +
geom_hline(aes(yintercept = -1, color = 'true value'))  +
ggtitle('traceplot of beta[1,]')
```



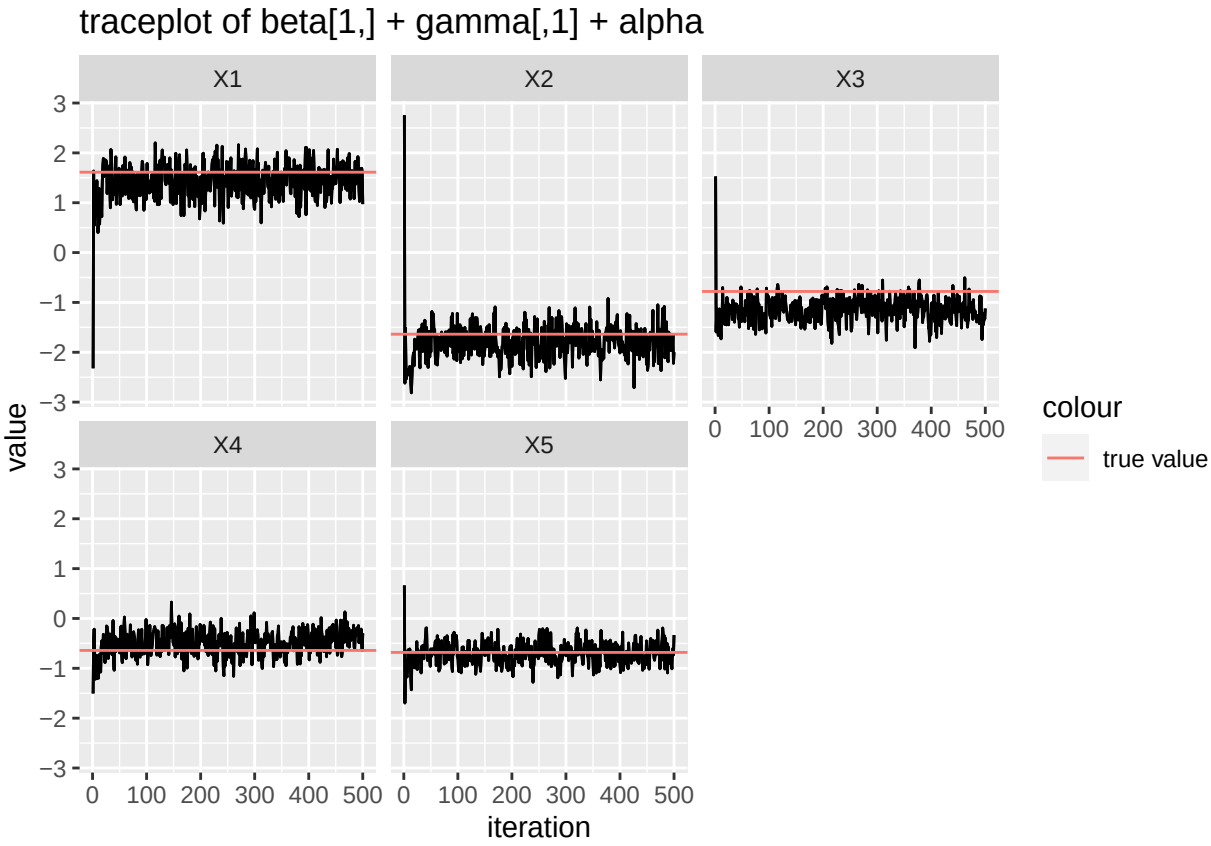

```
# gamma random effects
gamma_samples = g1$GAM
gamma_samples = data.frame(do.call(rbind,lapply(gamma_samples, function(x){x[,1]}))) # subgroup 1
gamma_samples[,'iteration'] = 1:nrow(gamma_samples)
ggplot(melt(gamma_samples, id.vars = 'iteration')) +
geom_line(aes(x = iteration, y = value)) +
geom_hline(data = data.frame(variable = paste0('X',1:5), value = gam[,1]),aes(yintercept = `value`, col
facet_wrap(~variable) +
ggtitle('traceplot of gamma[,1]')
```




```
# mean of subgroup 1 without alpha
mean_samples = intercept_samples[,-ncol(intercept_samples)] + gamma_samples[,-ncol(gamma_samples)]
mean_samples$iteration = 1:nrow(mean_samples)
ggplot(melt(mean_samples, id.vars = 'iteration')) +
geom_line(aes(x = iteration, y = value))  +
geom_line(aes(x = iteration, y = value)) +
geom_hline(data = data.frame(variable = paste0('X',1:5), value = -1 + gam[,1]),aes(yintercept = `value`
facet_wrap(~variable) +
ggtitle('traceplot of beta[1,] + gamma[,1]')
```




```
# mean of subgroup 4 with alpha
gamma_samples = g1$GAM
gamma_samples = data.frame(do.call(rbind,lapply(gamma_samples, function(x){x[,4]}))) # subgroup 4
gamma_samples[,'iteration'] = 1:nrow(gamma_samples)
mean_samples = intercept_samples[,-ncol(intercept_samples)] + gamma_samples[,-ncol(gamma_samples)] + be
mean_samples$iteration = 1:nrow(mean_samples)
ggplot(melt(mean_samples, id.vars = 'iteration')) +
geom_line(aes(x = iteration, y = value)) +
geom_line(aes(x = iteration, y = value)) +
geom_hline(data = data.frame(variable = paste0('X',1:5), value = gam[,4]),aes(yintercept = `value`, col
facet_wrap(~variable) +
ggtitle('traceplot of beta[1,] + gamma[,1] + alpha')
```


traceplot of beta[1,] + gamma[,1] + alpha
X1
X2
X3
X4
X5
value
iteration
colour
true value