Inspecting Convergence of a Model

Jacob Carey, Steven Cristiano, and Robert Scharpf

30 April 2018

Contents

1   Introduction

A Markov Chain Monte Carlo posterior simulation should be visually inspected to assess convergence.

```
# load CNPBayes
suppressMessages(library(CNPBayes))
# load packages for manipulating and visualizing data
suppressMessages(library(dplyr))
suppressMessages(library(tidyr))
suppressMessages(library(ggplot2))
```

2   Workflow

The number of starting values, burnin MCMC simulatioms, and MCMC simulations after burnin are controlled by a `McmcParams` object. Here, we specify a small number so that the example runs quickly. As discussed in overview, four types of models are possible–SB, MB, SBP, and MBP. Below we fit only the SBP model with 2 components.
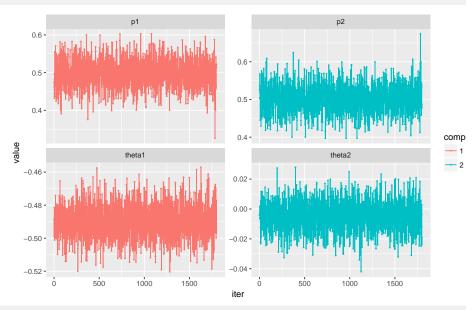
```
set.seed(1)
N <- 200
n <- 81
lrr <- c(rnorm(100, -0.5, sd=0.1), rnorm(100, 0, sd=0.1))
mp <- McmcParams(iter=600, burnin=10, nStarts=4)
mlist <- gibbs(model="SBP", mp=mp, dat=lrr, k_range=c(2, 2))
## Fitting SBP models
##   k: 2, burnin: 10, thin: 1
##       r: 1.22
##       eff size (minimum): 928.2
##       eff size (median): 1112.8
##   k: 2, burnin: 20, thin: 2
##       r: 1.01
##       eff size (minimum): 1668.2
##       eff size (median): 1800
##       marginal likelihood: 35.55
```

Had more than 1 model been fit, the models in `mlist` would be sorted by decreasing value of the marginal likelihood. Note, the marginal likelihood is only meaningful if the models have converged. The function `gibbs` requires that the effective sample size (number of independent MCMC draws) for all parameter chains is at least 500 and the multivariate Gelman Rubin diagnostic to be less than 1.2. Otherwise, starting values for 4 (`nStarts`) new models will be independently initialized and the burnin and number of `thin` iterations will be increased by a factor of 2. This process is repeated until the maximum number of burnin iterations has been reached (see `max_burnin`) or the effective size and Gelman Rubin criteria are both satisfied. For each model returned, the `nStarts` chains will be combined. In addition to the above automated diagnostics, we suggest plotting the chains as illustrated below.
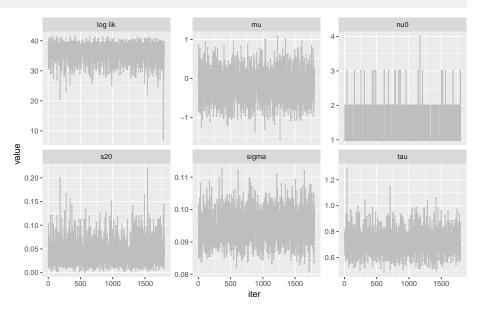
```
model <- mlist[[1]]
figs <- ggChains(model)
```

```
figs[[1]]
```



```
figs[[2]]
```


The marginal likelihood of the models can be retrieved using the `marginal_lik` accessor.

```
m.lik <- round(sapply(mlist, marginal_lik), 1)
m.lik
```

3   References