Convergence of the chains

Muhammad Yaseen

2026-11-04

Table of contents

1 Abstract **1**

2 Several chains **2**

3 Traces **2**

4 Gelman-Rubin **3**

5 Raftery-Lewis **4**

6 Heidelberger-Welch **5**

7 Effective sample size and Monte Carlo error **7**

8 The draws for further analysis **7**

9 posterior and bayesplot **8**

10 Interpretation of convergence **9**

11 References **9**

1 Abstract

Posterior summaries are valid only after the chains have reached their stationary distribution and are long enough. Crossa et al. (2011) used three diagnostics: Raftery and Lewis (1995), Heidelberger and Welch (1983) and Gelman and Rubin (1992). `bayes_ammi()` computes all three, together with effective sample sizes, through the package **coda**.

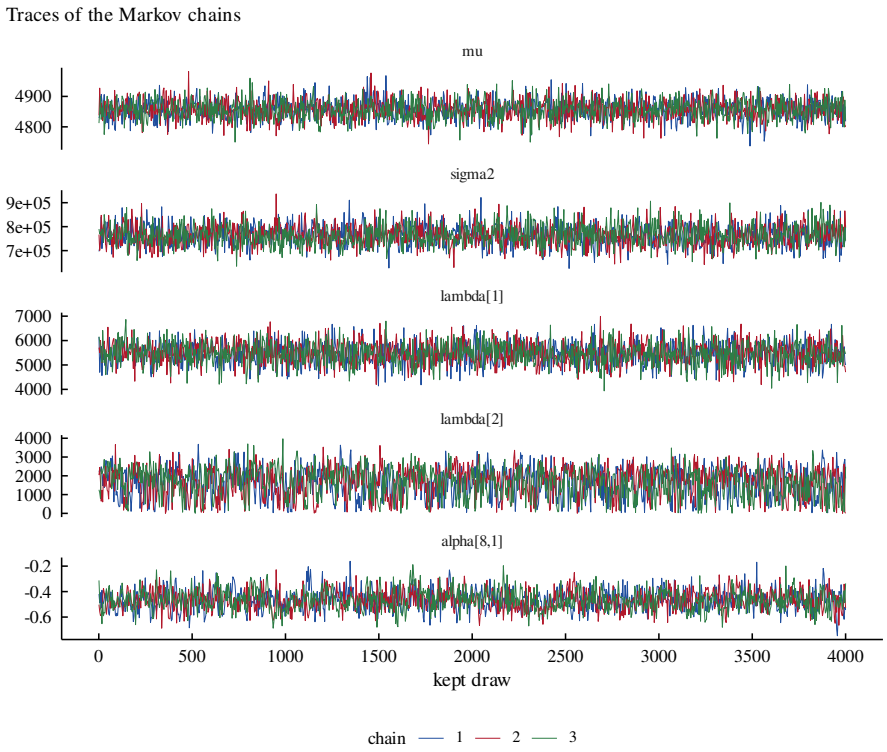
2 Several chains

```
fit <-
  bayes_ammi(
      .data   = Maiz
    , .y      = y
    , .gen    = entry
    , .env    = site
    , .rep    = rep
    , .nIter  = 4000
    , .k      = 2
    , .burnin = 1000
    , .thin   = 1
    , .chains = 3
    , .priors = bam_priors(
          mu_mean     = 0
        , mu_var      = 1e+8
        , tau_mean    = 0
        , tau_var     = 1e+8
        , delta_mean  = 0
        , delta_var   = 1e+8
        , lambda_mean = 0
        , lambda_var  = 1e+8
        , nu          = 1
        , s2          = NULL
        )
    , .level  = 0.95
    , .seed   = 5
    , .align  = TRUE
    , .model  = "AMMI"
    , .het    = FALSE
    , .waic   = TRUE
    )
```

The first chain starts at the least-squares fit and the others at dispersed values (singular values scaled by factors between 0.5 and 1.5, perturbed scores, error variance scaled between 0.5 and 2).

3 Traces

```
plot(x = fit, which = "trace", pars = c("mu", "sigma2", "lambda[1]",
 ↪  "lambda[2]", "alpha[8,1]"))
```



4 Gelman-Rubin

The potential scale reduction factor compares the variance between chains with the variance within them. Values below 1.1 (better, 1.01) indicate that the chains have mixed.

```
fit$diagnostics$gelman
```

```
Potential scale reduction factors:

          Point est. Upper C.I.
mu                 1          1
sigma2             1          1
tau[1]             1          1
tau[2]             1          1
tau[3]             1          1
tau[4]             1          1
tau[5]             1          1
tau[6]             1          1
tau[7]             1          1
```

```
tau[8]              1           1
tau[9]              1           1
delta[1]            1           1
delta[2]            1           1
delta[3]            1           1
delta[4]            1           1
delta[5]            1           1
delta[6]            1           1
delta[7]            1           1
delta[8]            1           1
delta[9]            1           1
delta[10]           1           1
delta[11]           1           1
delta[12]           1           1
delta[13]           1           1
delta[14]           1           1
delta[15]           1           1
delta[16]           1           1
delta[17]           1           1
delta[18]           1           1
delta[19]           1           1
delta[20]           1           1
lambda[1]           1           1
lambda[2]           1           1
```

```
summary(object = fit$summary$rhat)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.9999  1.0000  1.0002  1.0002  1.0004  1.0014
```

5 Raftery-Lewis

The dependence factor I measures how much longer the chain must be than an independent sample of the same precision; $I > 5$ indicates strong autocorrelation.

```
fit$diagnostics$raftery[[1]]
```

```
Quantile (q) = 0.025
Accuracy (r) = +/- 0.005
Probability (s) = 0.95

           Burn-in  Total Lower bound  Dependence
           (M)      (N)   (Nmin)       factor (I)
 mu        2        3866  3746         1.030
```

```
sigma2    3        4374  3746         1.170
tau[1]    2        3635  3746         0.970
tau[2]    2        3711  3746         0.991
tau[3]    2        3635  3746         0.970
tau[4]    2        3787  3746         1.010
tau[5]    2        3711  3746         0.991
tau[6]    2        3711  3746         0.991
tau[7]    2        3635  3746         0.970
tau[8]    2        3635  3746         0.970
tau[9]    2        3711  3746         0.991
delta[1]  2        3711  3746         0.991
delta[2]  2        3866  3746         1.030
delta[3]  2        3787  3746         1.010
delta[4]  2        3635  3746         0.970
delta[5]  2        3561  3746         0.951
delta[6]  2        3635  3746         0.970
delta[7]  2        3946  3746         1.050
delta[8]  2        3635  3746         0.970
delta[9]  2        3787  3746         1.010
delta[10] 2        3787  3746         1.010
delta[11] 2        3711  3746         0.991
delta[12] 2        3711  3746         0.991
delta[13] 2        3635  3746         0.970
delta[14] 2        3711  3746         0.991
delta[15] 2        3711  3746         0.991
delta[16] 2        3711  3746         0.991
delta[17] 3        4120  3746         1.100
delta[18] 2        3711  3746         0.991
delta[19] 2        3635  3746         0.970
delta[20] 2        3636  3746         0.971
lambda[1] 2        3711  3746         0.991
lambda[2] 8        9870  3746         2.630
```

6 Heidelberger-Welch

The stationarity test is applied to the whole chain, then after dropping the first 10%, 20%, … of it. The half-width test asks whether the mean is estimated precisely enough. Each test has level 0.05, so among many parameters a few failures are expected by chance.

```
fit$diagnostics$heidel[[1]]
```

```
          Stationarity start     p-value
          test         iteration
mu        passed         1       0.2958
```

sigma2	passed	1	0.4442
tau[1]	passed	1	0.9243
tau[2]	passed	1	0.1717
tau[3]	passed	1	0.7242
tau[4]	passed	1	0.5047
tau[5]	passed	1	0.9581
tau[6]	passed	1	0.5442
tau[7]	passed	1	0.4945
tau[8]	passed	1	0.5535
tau[9]	passed	1	0.7765
delta[1]	passed	1	0.3812
delta[2]	passed	1	0.1691
delta[3]	passed	1	0.6711
delta[4]	passed	1	0.3446
delta[5]	passed	1	0.2919
delta[6]	passed	1	0.3637
delta[7]	passed	1	0.9329
delta[8]	passed	1	0.7061
delta[9]	passed	1	0.1740
delta[10]	passed	1	0.4458
delta[11]	passed	801	0.0677
delta[12]	passed	1	0.9949
delta[13]	passed	1	0.0578
delta[14]	passed	1	0.2482
delta[15]	passed	1	0.4003
delta[16]	passed	1	0.3308
delta[17]	passed	1	0.5631
delta[18]	passed	1	0.1621
delta[19]	passed	1	0.6172
delta[20]	passed	1	0.4041
lambda[1]	passed	1	0.2844
lambda[2]	passed	1	0.4288

	Halfwidth test	Mean	Halfwidth
mu	passed	4858.2	1.02
sigma2	passed	762251.7	1728.80
tau[1]	passed	-242.6	2.92
tau[2]	passed	-253.6	2.80
tau[3]	passed	-36.7	2.85
tau[4]	passed	359.5	2.85
tau[5]	passed	391.2	2.87
tau[6]	passed	470.4	2.75
tau[7]	passed	-186.1	2.87
tau[8]	passed	-572.2	2.86
tau[9]	passed	70.3	2.86

```
delta[1]  passed     -1249.7    4.46
delta[2]  passed      -653.0    4.34
delta[3]  passed       245.6    4.28
delta[4]  passed       374.6    4.39
delta[5]  passed       101.4    4.40
delta[6]  passed      1450.5    4.33
delta[7]  passed     -1630.8    4.29
delta[8]  passed      -829.7    4.39
delta[9]  passed       108.5    4.47
delta[10] passed     -1922.0    4.48
delta[11] passed       449.7    4.95
delta[12] passed      2661.4    4.41
delta[13] passed      1471.4    4.43
delta[14] passed      1191.0    4.46
delta[15] passed       191.5    4.33
delta[16] passed       545.5    4.27
delta[17] failed        21.7    4.41
delta[18] passed      -306.0    4.58
delta[19] passed     -2204.7    4.25
delta[20] failed       -19.2    4.23
lambda[1] passed      5464.0   15.56
lambda[2] passed      1644.9   67.79
```

7 Effective sample size and Monte Carlo error

```
fit$summary[block %in% c("mu", "sigma2", "lambda"), list(parameter, mean,
↪  sd, mc_se, ess, rhat)]
```

```
   parameter       mean         sd       mc_se       ess      rhat
1:        mu   4857.976    32.5537   0.2925239 12384.485 0.9999954
2:    sigma2 763066.073 43412.8267 500.5970862  7520.721 1.0002382
3: lambda[1]   5468.348   457.4447   4.6530521  9664.991 1.0003939
4: lambda[2]   1654.395   784.8158  21.0012922  1396.508 1.0000922
```

The Monte Carlo standard error `mc_se` is $\mathrm{sd}/\sqrt{\mathrm{ess}}$. Run longer chains when it is not small compared with the precision you need. The weak second singular value mixes more slowly than the other parameters.

8 The draws for further analysis

`fit$draws` is a `coda::mcmc.list`, so every function of **coda** applies:

```
coda::autocorr.diag(mcmc.obj = fit$draws[, c("lambda[1]", "lambda[2]")],
↪  lags = c(1, 5, 10))
```

```
        lambda[1] lambda[2]
Lag 1  0.08931897 0.7099201
Lag 5  0.02070703 0.3129273
Lag 10 0.01404242 0.1274023
```

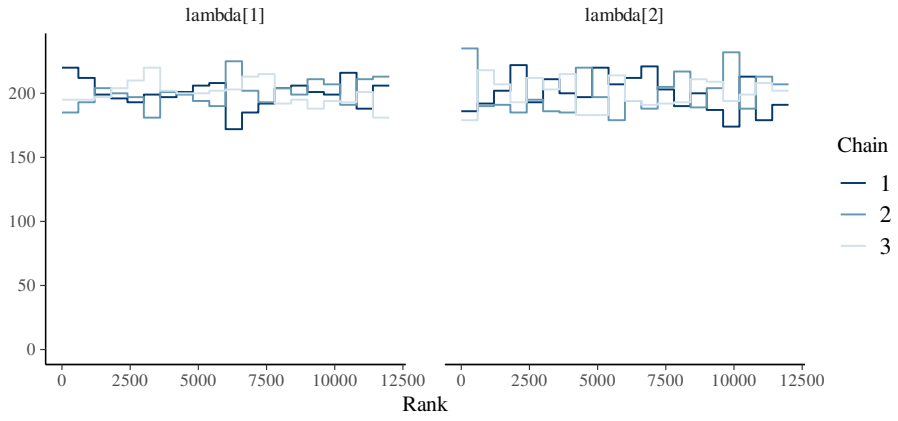
9 posterior and bayesplot

The draws also convert to a `draws_df` of **posterior**, which gives the rank-normalised $\widehat{R}$ and the bulk and tail effective sample sizes of Vehtari et al. (2021), and the plots of **bayesplot** take the `mcmc.list` directly (Gabry et al. 2019):

```
dr <- posterior::as_draws_df(x = fit)
posterior::summarise_draws(
    .x = posterior::subset_draws(x = dr, variable = c("mu", "sigma2",
    ↪  "lambda[1]", "lambda[2]"))
  , "mean", "sd", "rhat", "ess_bulk", "ess_tail"
  )
```

```
# A tibble: 4 x 6
  variable     mean      sd  rhat ess_bulk ess_tail
  <chr>       <dbl>   <dbl> <dbl>    <dbl>    <dbl>
1 mu          4858.    32.6  1.00   11956.   12016.
2 sigma2    763066. 43413.   1.00    6507.    9830.
3 lambda[1]   5468.   457.   1.00    9062.   10658.
4 lambda[2]   1654.   785.   1.01    1242.    3663.
```

```
bayesplot::mcmc_rank_overlay(x = fit$draws, pars = c("lambda[1]",
 ↪  "lambda[2]"))
```



Rank plots are flat and overlapping when the chains mix well.

10 Interpretation of convergence

```
bam_interpret(
    x             = fit
  , higher_better = TRUE
  , top           = 5
  )$findings[1:2]
```

```
[1] "The chains have converged by the criteria of Crossa et al. (2011) (largest potential sc
[2] "6 of 198 Heidelberger-Welch stationarity and half-width tests failed (about 5% fail by
```

11 References

Crossa, J., Perez-Elizalde, S., Jarquin, D., Cotes, J. M., Viele, K., Liu, G. and Cornelius, P. L. (2011). Bayesian estimation of the additive main effects and multiplicative interaction model. *Crop Science*, 51(4), 1458–1469. doi:10.2135/cropsci2010.06.0343

Gelman, A. and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4), 457–472.

Raftery, A. E. and Lewis, S. M. (1995). The number of iterations, convergence diagnostics and generic Metropolis algorithms. In W. R. Gilks, D. J. Spiegelhalter and S. Richardson (eds), *Practical Markov Chain Monte Carlo*. Chapman and Hall, London.

Heidelberger, P. and Welch, P. D. (1983). Simulation run length control in the presence of an initial transient. *Operations Research*, 31(6), 1109–1144.

Plummer, M., Best, N., Cowles, K. and Vines, K. (2006). CODA: convergence diagnosis and output analysis for MCMC. *R News*, 6(1), 7–11.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B. and Burkner, P.-C. (2021). Rank-normalization, folding, and localization: an improved R-hat for assessing convergence of MCMC. *Bayesian Analysis*, 16(2), 667–718.

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M. and Gelman, A. (2019). Visualization in Bayesian workflow. *Journal of the Royal Statistical Society A*, 182(2), 389–402.