Convergence diagnostics for MCMC chains

MLPM Summer School 2015

Catalina A. Vallejos, Nils Eling, John C. Marioni

22 September 2015

Convergence diagnostics for MCMC chains

To assess convergence is not a trivial task, especially in large-dimensional settings . . .

What can we do?

- Visual inspection: trace-plots, cumulative means plots, etc
- Statistical tests: e.g. the ones implemented in the `coda` library

We can also run multiple chains (with different starting values) and check if all chains converge to the same values.

Convergence diagnostics for MCMC chains

To illustrate convergence diagnostic tools, we create 4 **artificial** chains

```
set.seed(1)
Chain1 = rnorm(n = 10000, mean = 5, sd = 2)
set.seed(2)
Chain2 = c(rnorm(n = 5000, mean = (1:5000)/1000, sd = 1),
           rnorm(n = 5000, mean = 5, sd = 1))
set.seed(3)
Chain3 = c(rep(1, 2000), rep(-2, 500),
           rep(2, 5000), rep(5, 2500)) +
         rnorm(n = 10000, mean = 0, sd = 1)
set.seed(4)
Chain4 = sin((1:10000)/1000) + rnorm(10000, 0, 1.5)
```

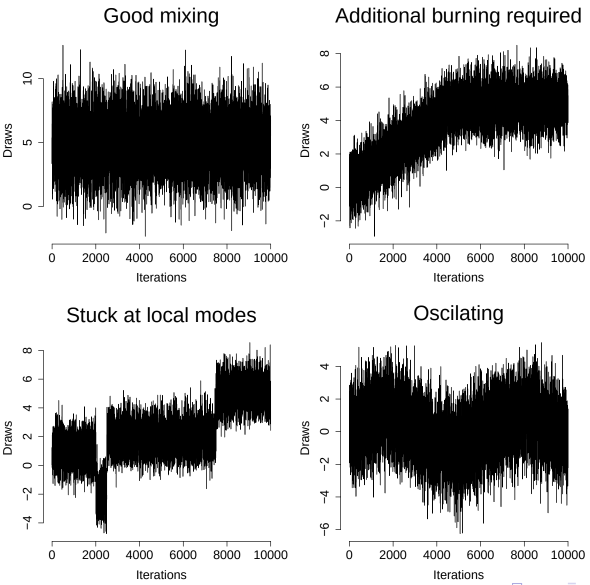
Convergence diagnostics for MCMC chains

Visual inspection - Traceplots

```
par(mfrow = c(2,2))
plot(Chain1, type = "l", bty = "n", xlab = "Iterations",
     ylab = "Draws", main = expression("Good mixing"),
     cex.main = 2.5, cex.lab = 1.5, cex.axis = 1.5)
plot(Chain2, type = "l", bty = "n", xlab = "Iterations",
     ylab = "Draws", main = expression("Additional burning required"),
     cex.main = 2.5, cex.lab = 1.5, cex.axis = 1.5)
plot(Chain3, type = "l", bty = "n", xlab = "Iterations",
     ylab = "Draws", main = expression("Stuck at local modes"),
     cex.main = 2.5, cex.lab = 1.5, cex.axis = 1.5)
plot(Chain4, type = "l", bty = "n", xlab = "Iterations",
     ylab = "Draws", main = expression("Oscilating"),
     cex.main = 2.5, cex.lab = 1.5, cex.axis = 1.5)
```

Convergence diagnostics for MCMC chains

Visual inspection - Traceplots



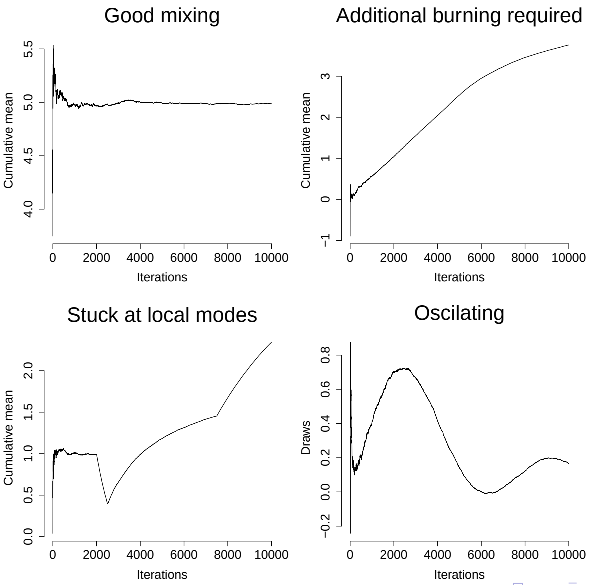
Convergence diagnostics for MCMC chains

Visual inspection - Cumulative means plots

```
par(mfrow = c(2,2))
plot(cumsum(Chain1)/1:10000, type = "l", bty = "n",
     xlab = "Iterations", ylab = "Cumulative mean",
     main = expression("Good mixing"),
     cex.main = 2.5, cex.lab = 1.5, cex.axis = 1.5)
plot(cumsum(Chain2)/1:10000, type = "l", bty = "n",
     xlab = "Iterations", ylab = "Cumulative mean",
     main = expression("Additional burning required"),
     cex.main = 2.5, cex.lab = 1.5, cex.axis = 1.5)
plot(cumsum(Chain3)/1:10000, type = "l", bty = "n",
     xlab = "Iterations", ylab = "Cumulative mean",
     main = expression("Stuck at local modes"),
     cex.main = 2.5, cex.lab = 1.5, cex.axis = 1.5)
plot(cumsum(Chain4)/1:10000, type = "l", bty = "n",
     xlab = "Iterations", ylab = "Draws",
     main = expression("Oscilating"),
     cex.main = 2.5, cex.lab = 1.5, cex.axis = 1.5)
```

Convergence diagnostics for MCMC chains

Visual inspection - Cumulative means plots



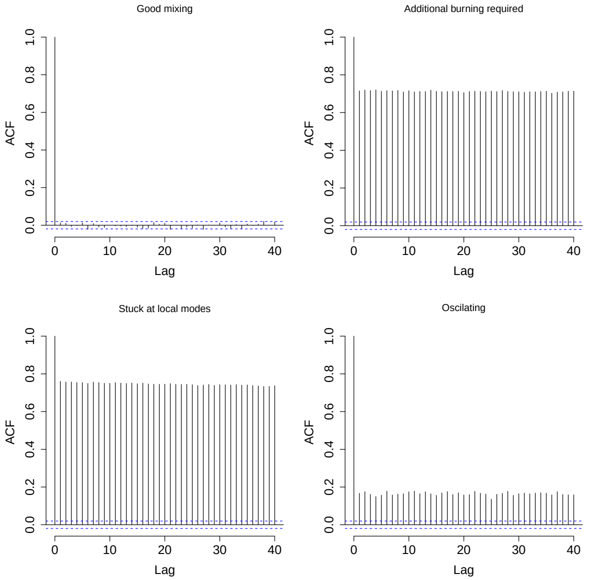
Convergence diagnostics for MCMC chains

Visual inspection - Auto-correlation plots

```
par(mfrow = c(2,2))
acf(Chain1, bty = "n",
     main = expression("Good mixing"),
     cex.main = 2.5, cex.lab = 1.5, cex.axis = 1.5)
acf(Chain2, bty = "n",
     main = expression("Additional burning required"),
     cex.main = 2.5, cex.lab = 1.5, cex.axis = 1.5)
acf(Chain3, bty = "n",
     main = expression("Stuck at local modes"),
     cex.main = 2.5, cex.lab = 1.5, cex.axis = 1.5)
acf(Chain4, bty = "n",
     main = expression("Oscilating"),
     cex.main = 2.5, cex.lab = 1.5, cex.axis = 1.5)
```

Convergence diagnostics for MCMC chains

Visual inspection - Auto-correlation plots



Convergence diagnostics for MCMC chains

Statistical tests

The R library makes available several statistical test for convergence diagnostics. To load this library use

```
library(coda)
```

To use these tools, a first step is to transform chains into objects of class mcmc.

```
Chain1.mcmc = mcmc(Chain1)

Chain2.mcmc = mcmc(Chain2)

Chain3.mcmc = mcmc(Chain3)

Chain4.mcmc = mcmc(Chain4)
```

Convergence diagnostics for MCMC chains

Statistical tests - Geweke

Compares the means of two non-overlapping segments of the chain (typically the first 10% and last 50% of draws). It returns a z-score which is adjusted for autocorrelation

```r
geweke.diag(Chain1.mcmc)$z
```

```
##        var1
## -0.02164115
```

```r
geweke.diag(Chain2.mcmc)$z
```

```
##      var1
## -57.39918
```

```r
geweke.diag(Chain3.mcmc)$z
```

```
##      var1
## -2.722006
```

```r
geweke.diag(Chain4.mcmc)$z
```

```
##     var1
## 1.416636
```

Convergence diagnostics for MCMC chains

Statistical tests

Other diagnostics implemented in `coda` include

- Gelman and Rubin
- Raftery and Lewis
- Heidelberg and Welch

Because of time restrictions, we won't explore all of these today.

NOTE: For better results it is important to combine visual inspection and test-based convergence diagnostics

Convergence diagnostics for MCMC chains

Each of these convergence diagnostics are designed to assess the convergence of a chain for a single parameter

What to do for high-dimensional models?

Once again, there is no trivial answer. Some ideas:

- Select a random subset of model parameters
- Calculate averages within sets of parameters