STAT 532
Fall 2013 and 2014
Convergence Diagnostic Summaries

Contributors

Convergence Diagnostic	Contributors
Potential Scale Reduction Factor	*Matt (2013)*
n effective	*Matt (2013)*
Geweke's	*Marie, Bobby (2014); Chris, Liz (2013)*
Raftery and Lewis'	*Maya (2014); Amber, Tan (2013)*
Brooks and Gelman's (multivariate)	*Jay, Koji (2014); Alyssa, Tony (2013)*
Heidelberg and Welchs'	*Lauren (2014); Kevin, Pat (2013)*
QED (Michael Lerch's)	*Maya, Lauren, Marie (2014)*
Hellinger Distance (Ed Boone's)	*Jay, Bobby, Koji (2014)*

Effective Sample Size $\hat{n}_{eff}$ in `coda`

Matt Tyers

Calculation of the effective sample size $\hat{n}_{eff}$ in `coda` is done using a different (and much more code-efficient) technique than the algorithm presented in Chapter 11 of the Gelman, et.al. text.

Function `effectiveSize()` first checks to see if it has been passed an MCMC list. If so, it breaks the list apart by variables and runs itself on the set of chains for each variable. Otherwise (or after this), it calls `spectrum0.ar()` to calculate the spectral density at frequency zero of each chain. The spectral density *spec* is calculated in `spectrum0.ar()` as

$$spec = \frac{Variance.unexplained.by.Autoregressive.Model}{(1-\Sigma Autoregressive.coefficients)^2}$$

If the spectral density is zero, i.e. there is no variance in the chain left unexplained by the autoregressive model, `effectiveSize` returns an effective sample size of $\hat{n}_{eff} = 0$ for the chain. If not, it returns

$$\hat{n}_{eff} = \frac{number.of.iterations \times V(chain)}{spec}$$

where $V(chain)$ is the overall variance of the chain and *spec* is the spectral density as calculated above.

Potential Scale Reduction Factor $\hat{R}$ in `coda`

Calculation of the potential scale reduction factor $\hat{R}$ in `coda` using `gelman.diag()` uses the within-chain variance W and between-chain variance B employed in the Gelman, et.al. text (though without splitting the chains), but uses them differently, and allows for multivariate chains.

Function `gelman.diag()` requires the input of a mcmc object, and allows the user to specify whether to use a transformation on the chain, whether to accept an "auto-burnin", whether the chains are to be considered multivariate, and a confidence level for the upper CI limit reported.

If the "auto-burnin" is accepted, `gelman.diag()` discards the first half of the chains as burnin, otherwise, the full chains are retained. It then extracts the number of iterations, the number of chains, the number of variables used, and the variable names from the mcmc object, and then converts the mcmc object to a matrix. It then computes the variance of each chain s_j^2, average within-chain variance W, and between-chain variance B, using the same calculations as given in Gelman, et.al. However, s_j^2, W, and B are stored as vectors with a value for each variable in the mcmc object the function is given. In the non-multivariate case, it generates diagonal matrices $s2$, w, and b for each, and all subsequent calculations are done using these matrices, which cuts down on processing time.

In the non-multivariate case, function `gelman.diag()` returns a vector with two outputs, a point estimate for $\hat{R}$, and an upper CI bound of the confidence specified by the user. These are calculated as

$psrf = (\sqrt{df_{adj} R^2_{est}}, \sqrt{df_{adj} R^2_{upper}})$

where

$R^2_{est} = R^2_{fixed} + R^2_{random}$ and $R^2_{upper} = R^2_{fixed} + F_{(\frac{1+conf}{2}, df_B, df_W)} R^2_{random}$

$R^2_{fixed} = \frac{N.iter-1}{N.iter}$ and $R^2_{random} = (1 + \frac{1}{N.chain})(\frac{1}{N.iter})(\frac{b}{w})$

and

$df_{adj} = \frac{df_V+3}{df_V+1}$ where $df_V = \frac{2V^2}{var(V)}$

$df_B = N.chain - 1$ and $df_W = \frac{2w^2}{var(w)}$

In these,

$V = \frac{(N.iter-1)}{N.iter} w + \frac{1+\frac{1}{N.chain}}{N.iter} b$

and

$var(V) = \frac{(N.iter-1)^2 var(w) + (1+\frac{1}{N.chain})^2 var(b) + 2(N.iter-1)(1+\frac{1}{N.chain}) cov(w,b)}{N.iter^2}$

where

$var(w) = \frac{var(s2)}{N.chain}$ and $var(b) = \frac{2b^2}{N.chain-1}$

$cov(w,b) = \frac{N.iter}{N.chain} var(s2, \bar{x}^2 - 2\hat{\mu} var(s2), \bar{x})$ and $\hat{\mu} = mean(\bar{x})$

In the multivariate case, it computes the Choleski factorization of the matrix for W and stores it as CW. It then solves for X_1 in

$CW \times X_1 = B$

and then for X_2 in

$CW \times X_2 = (X_1)^T$

and takes the eigenvalues of X_2. The multivariate $\hat{R}$ is then calculated as

$\hat{R} = \sqrt{(1 - \frac{1}{N.iter}) + (1 + \frac{1}{N.var})(\frac{eigenvalues.of.X_2}{N.iter})}$

Gelman and Brooks Convergence Diagnostic (Multivariate)

Jay Rosencrantz (2014), Alyssa Peck (2013

1 Introduction

The potential scale reduction factor $\hat{R}$ was created by Gelman and Rubin and uses the within-chain variance W and the between-chain variance B as a tool for assessing convergence in an MCMC setting. In the univariate case, $\hat{R}$ is given as:

$$\hat{R} = \frac{\hat{V}}{W}$$

$$V = \frac{n-1}{n}W + \frac{1+\frac{1}{m}}{n}B$$

This differs somewhat from *Bayesian Data Analysis*, Gelman et al in that the between-chain variance B is scaled by $1 + \frac{1}{m}$. In addition, Gelman and Brooks do not take the square root of this $\hat{R}$, as is done in *Bayesian Data Analysis*.

2 Multivariate Extension

The multivariate methods for the potential scale reduction factor $\hat{R}$ (PSRF) is a straightforward extension of the univariate case. Scalars change to vectors and vectors change to matricies. Now, W and B will be defined by:

$$W = \frac{1}{m(n-1)}\sum_{j=1}^{m}\sum_{i=1}^{n}(\phi_{jt} - \bar{\phi}_{j\cdot})(\phi_{jt} - \bar{\phi}_{j\cdot})^T$$

$$B = \frac{n}{(m-1)}\sum_{j=1}^{m}(\phi_{j\cdot} - \bar{\phi}_{\cdot\cdot})(\phi_{j\cdot} - \bar{\phi}_{\cdot\cdot})^T$$

where ϕ_{jt} is an $m * n$ matrix and W,B are vectors of length p (number of parameters). The actual formulaic expression for $\hat{R}$ is now somewhat more complicated:

$$\hat{R}^p = \frac{n-1}{n} + (\frac{m+1}{m})\lambda_1$$

where λ_1 is the largest eigenvalue of the positive definite matrix $W^{-1}B/n$. The version of the scale reduction factor gives an approximate upper bound to the maximum of all univariate $\hat{R}$ values for the p parameters. By accounting for correlation between parameters, the estimate of $\hat{R}$ will be a more conservative estimate than the p univariate estimates. 4

3 Drawbacks

Calculation using this multivariate extension is computationally intensive, which is becoming less and less of a problem now. Additionally, this calculation of $\hat{R}^p$ only gives a "summary measure" of total convergence for all p parameters. If interest lies only in certain parameters, this $\hat{R}$ may not be appropriate. Gelman and Brooks also mention that, similar to the univariate case, the diagnostic can be very sensitive to initial starting points in the multiple chains scenario.

4 References

1. Brooks, Stephen P., and Andrew Gelman. "General Methods for Monitoring Convergence of Iterative Simulations." Journal of Computational and Graphical Statistics 7.4 (1998): 434. Web.

2. Gelman, Andrew. "Chapter 11." Bayesian Data Analysis. Boca Raton, FL: Chapman & Hall/CRC, 2004. N. pag. Print.

Brooks and Gelman's diagnostic (multivariate)

This is a multiple variables version of Gelman's diagnostic. Notable difference from univariate $\hat{R}$ would be that, the coeffcient of B/n is $(1 + \frac{1}{m})$, and that we extract scalar of $\hat{R}$ using eigen value. We also do not take the square root of $\hat{R}$. Following is the detail of the calculation.

At first, we define W and B, which are the within chain variability and between chain variability each, of p-dimension version. $\psi_{jt}^{(i)}$ denotes that i th element of the parameter vector in chain j at time t.

$$W = \frac{1}{m(n-1)} \Sigma_{j=1}^{m} \Sigma_{t=1}^{n} (\psi_{jt} - \bar{\psi_{j\cdot}})(\psi_{jt} - \bar{\psi_{j\cdot}})^T$$

where ψ_{jt} is m*n matrix and $\bar{\psi_{j\cdot}}$ is a vector of length m.

$$B/n = \frac{1}{m-1} \Sigma_{j=1}^{m} (\bar{\psi_{j\cdot}} - \bar{\psi_{\cdot\cdot}})(\bar{\psi_{j\cdot}} - \bar{\psi_{\cdot\cdot}})^T$$

This is for between variability for chain to chain. V is defined as follows,

$$\hat{V} = \frac{n-1}{n} W + (1 + \frac{1}{m}) B/n$$

Then, the multivariate version of $\hat{R}$ is,

$$\hat{R^p} = max_a \frac{a^T \hat{V} a}{a^T W a}$$

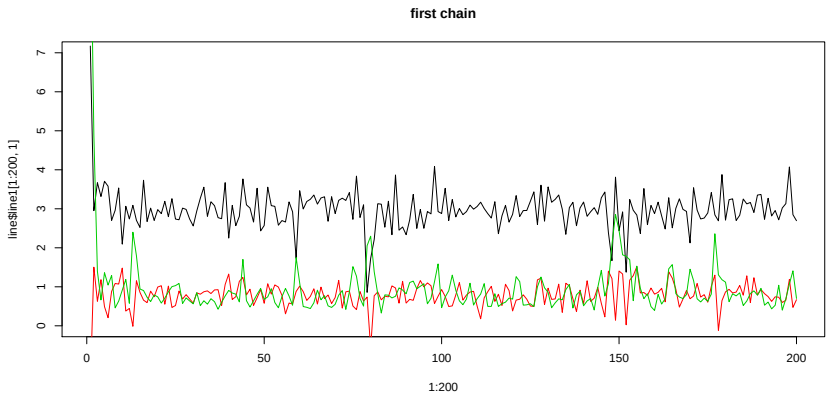
,which means the largest eigen value of $\frac{\hat{V}}{W}$. Maybe one can understand intuitively ,remembering the Principal Component Analysis, when we take the largest eigen value of variance-covariance matrix.
We took largest eigenvalue, so as not to underestimate $\hat{R}$. This is the multivariate case of potential scale reduction factor(PSRF) or MPSRF.

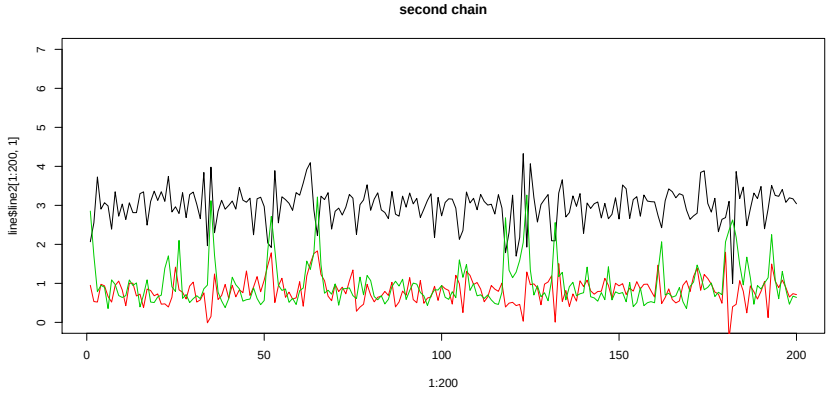
- The advantages is:
  It is useful when there are correlations among variables, and more conservative than univariate $\hat{R}$.

- The disadvantages is:
  computation is difficult even for low dimensions

- Example of MPSRF:
  MPSRF can be calculated using *gelman.diag* function in *coda* package, with the argument *multivariate* =*T*.

```
## Loading required package:  lattice
```






```
## Potential scale reduction factors:
##
##       Point est. Upper C.I.
## alpha       1.02       1.02
## beta        1.00       1.00
## sigma       1.04       1.12
##
## Multivariate psrf
##
## 1.01
```

R Code

```
library(coda)
data(line)
plot(1:200, line$line1[1:200, 1], type = "l", col = 1, main = "first chain",
    ylim = c(0, 7))
lines(1:200, line$line1[1:200, 2], type = "l", col = 2)
lines(1:200, line$line1[1:200, 3], type = "l", col = 3)
plot(1:200, line$line2[1:200, 1], type = "l", col = 1, main = "second chain",
    ylim = c(0, 7))
lines(1:200, line$line2[1:200, 2], type = "l", col = 2)
lines(1:200, line$line2[1:200, 3], type = "l", col = 3)
```

```
mc.list <- mcmc.list(line[[1]], line[[2]])
gelman.diag(mc.list, multivariate = T)
```

Geweke's Diagnostic
The Guide for the Rest of Us

Marie Liley Fall 2014

The basic idea:

The Geweke diagnostic tests for stationarity of a chain. The concept is if the first part of the chain and the last part of the chain are exploring the same parameter space (ie. attained stationarity) then we can say the target distribution has been reached. The Geweke diagnostic takes two non-overlapping segments (the default is the first 10% and last 50%) of the Markov chain and performs a test for a difference in means using a Z test statistic. The null hypothesis is that the means are equal and the alternative hypothesis is that the means are different. If the mean of the first 10% is not markedly different than the last 50%, then we conclude that the chain has reached stationarity in the first 10% of the chain. If the means are different, then we conclude stationarity has not been reached.

The Geweke diagnostic function can be found in the R *Coda* package. The function is as follows: geweke.diag(x, frac1=0.1, frac2=0.5). x is an MCMC object and frac1 and frac2 control what parts of the chain are used to test for stationarity. A very important feature of this diagnostic is that it tests each chain individually ie. if three chains are used then this test must be performed on each of the three chains. This function will output a Z-statistic. If the absolute value of the Z-statistic is less than two, we can conclude that the chain achieved stationarity in the first 10% of the chain. If the absolute value of the Z-statistic is greater than two, then we conclude that the samples are being drawn from different distributions. If stationarity is not reached, try requiring a longer burn in and then retesting for stationarity.

Okay, but can I see an example?

Let's take a look at when a chain achieves stationarity and when a chain does not achieve stationarity.

```
require(coda)
non.station<-diffinv(rnorm(999))
geweke.diag(non.station)

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

  var1
-5.737

station<-rnorm(1000,0,1)
geweke.diag(station)

Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

 var1
0.832
```

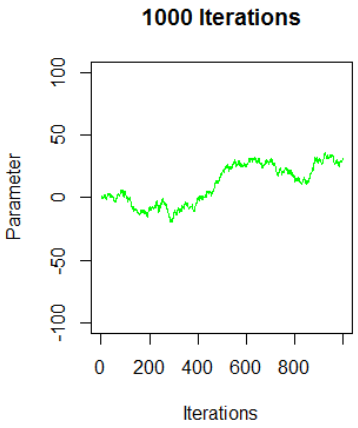


Figure 1: Non-Stationary

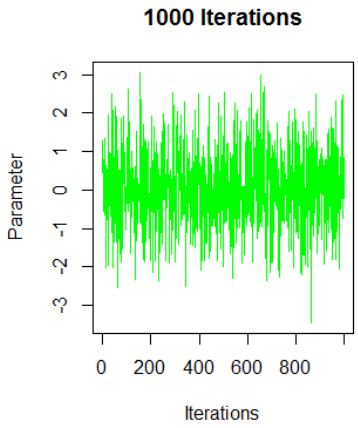


Figure 2: Stationary

Figure one corresponds to the Z-score=-5.737. This Z-score indicates that the chain did not achieve stationarity. Figure two corresponds to the Z-score=0.832. This Z-score indicates that the chain did achieve stationarity.

What are the pros and cons of this diagnostic?

Advantages

- Can be used to assess appropriate amount of burn in
- Easy to use (built in function in the Coda package)
- Interpretation of a Z-score isn't hard
- Tests a single chain

Disadvantages

- Have to test every chain individually (tedious if a lot of chains)
- If the samples are reordered, Geweke will produce a different Z-statistic!
- Test is sensitive to the size of the plotting window- ie. window size can change Z-statistic

References

1. Lerch, Michael. *Equivalence testing for MCMC convergence assessment*
2. Fang, Qijun (2014) *A Brief Introduction to Geweke's Diagnostics.* http://math.arizona.edu/ piegorsch/675/GewekeDiagnostics.pdf
3. Cowles, M. K., and Carlin, B. P. (1996) Markov chain Monte Carlo convergence diagnostics: A comparative review. Journal of the American Statistical Association Markov chain Monte Carlo convergence diagnostics: A comparative review. Journal of the American Statistical Association

Geweke's Diagnostic

bobby Hsu

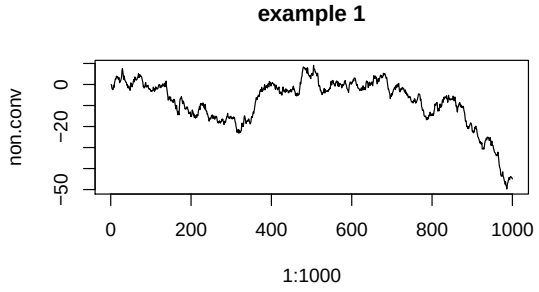
Description

Geweke's diagnostic compares two non-overlapping sections of a MCMC chain and see how different the mean of each section are from each other. The results can help making decision on burn-in period, and assess if the chain has reached stationarity. The Geweke test statistic (Z) is the standard z-test between the two sections, but instead of the sample standard deviation of two sections, Geweke uses spectral density analysis, which adjusted for the autocorrelation within the chain.

Example

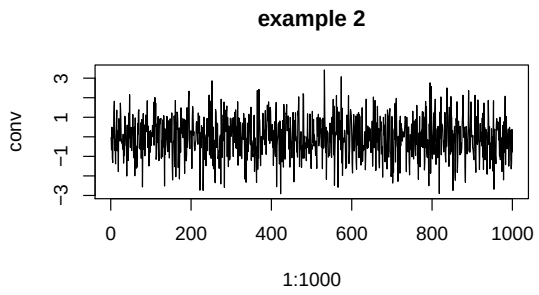
The default in `geweke.diag` uses the first 10% and the last 50% of the MCMC chain. Example 1 compared sections from a non-stationary correlated chain. A large absolute value for z-score indicated that stationary might not be reached.

```
non.conv<- mcmc(diffinv(rnorm(999)))
geweke.diag(non.conv)


Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

var1
1.36
```



Example 2 compared sections of a MCMC chain from a standard normal distribution. Small absolute value for z-score indicated little problems in convergence.

```
conv<- mcmc(rnorm(1000))
geweke.diag(conv)


Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5

  var1
-0.166
```



Conclusion

Geweke's diagnostic tests the convergence of a single chain, and is easy to interpret by comparing means from non-overlapping sections using z-score. However, the diagnostic is sensitive to the size of the sections (windows). A wide window could mask the non-stationary portion because Geweke is only concerned with the overall mean in each section compared. I find it useful to compare Geweke's results with the sampling history plot. One can often see from the traceplot that if the Geweke statistic make sense, or if one need to adjust the window size.

Raftery and Lewis's Diagnostic

Maya Tsidulko

1 Introduction

The Raftery-Lewis convergence diagnostic proposes the number of samples to be taken in order to achieve a specified level of MCMC precision. Failure of the test would indicate that a longer Markov chain is needed. The diagnostic was first developed by Raftery and Lewis in 1992, with an updated version in 1996. To motivate the idea behind this diagnostic, let's suppose one is interested in a quantity Q_q such that $P(Q \leq Q_q|y) = q$, where q is some arbitrary cumulative probability. We define $\widehat{Q}_q$ as an estimator for Q_q, corresponding to an estimated probability $P(Q \leq \widehat{Q}_q) = \widehat{P}_q$. By theory, we know that the simulated posterior distribution converges to the true distribution as the simulation size grows, and thus $\widehat{Q}_q$ can achieve any degree of accuracy is the simulator is run for a very long time. The cost of running a very long simulation, though, is high as it can be wasteful. It turns out that we can use coverage probability to measure the accuracy of $\widehat{Q}_q$ and stop the chain when we reach the desired level of accuracy.

2 Summary of Method

Two user inputs are required for the Raftery and Lewis diagnostic: the coverage probability s, and the amount of tolerance r. Then, the proposed number of samples is the number to ensure with probability s that $P(\widehat{Q}_q < Q|y)$ is found to be within $\pm r$. For example, suppose we want s to be 0.95 and r to be 0.005. This input corresponds to requiring that the estimate of the cumulative distribution function of the 2.5th percentile to be estimated to within ± 0.5 percentage points with probability 0.95. The Raftery-Lewis diagnostics test finds the number of iterations, M, that need to be discarded (burn-ins) and the number of iterations needed, N, to achieve a desired precision. The quantile of interest q may be determined as the empirical quantile of interest based on a pilot samples. Then, the samples Q^t drawn in the MCMC algorithm are used in constructing a binary $0-1$ process Z_t by setting $Z_t = 1$ if $Q^t \leq \widehat{Q}_q$ and 0 otherwise for all t. Finally, the Z_t's are thinned such that the thinned chain appears to be a dichotomous Markov chain. From this thinned chain, the Markov transition matrix is determined and necessary sample size can be calculated to achieve s and r as specified by the user.

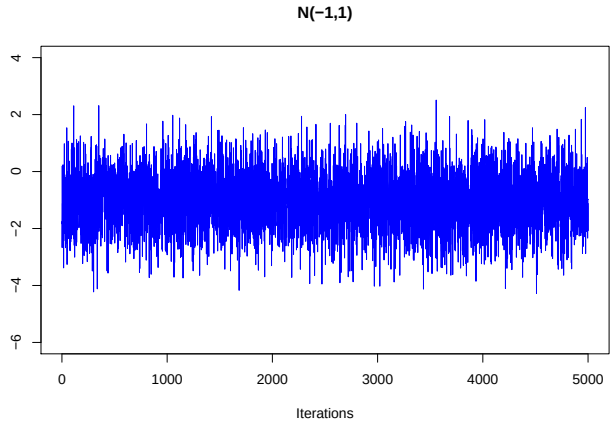
3 Advantages

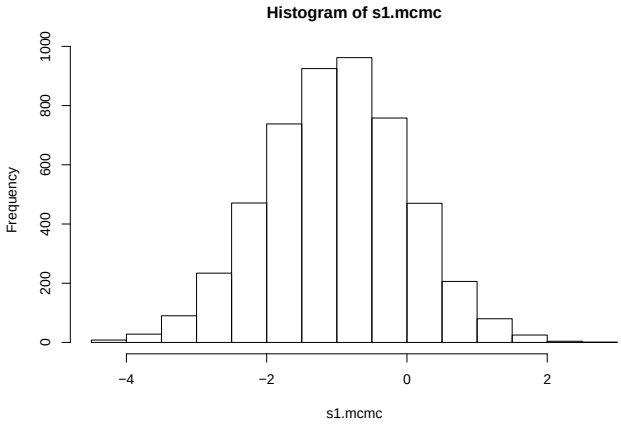
A potential advantage of the Raftery-Lewis diagnostic is that it is specific to inferences such as probabilities or quantiles, which are often of interest to the researcher. Since this diagnostic proposes a total number of samples to be drawn, it could be paired with additional convergence diagnostics. The length of the chains may be initially determined by the Raftery-Lewis diagnostic and then another diagnostic, such as the Quantile Equivalence Diagnostic (Michael Lerch's proposed diagnostic) can be used to check if the desired precision has, in fact, been reached. 13

4 Example in R

In the following example, we simulate a chain from $N(-1, 1)$, and use the Raftery-Lewis diagnostic to estimate the number of samples to be drawn.

```
s1<-rnorm(5000,-1,1)
s1.mcmc<-mcmc(s1.1[1:5000])

raftery.diag(s1.mcmc)

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

 Burn-in  Total Lower bound  Dependence
 (M)      (N)   (Nmin)       factor (I)
 2        3741  3746         0.999
```





The method requires 3746 pilot iterations (N_{min}) to obtain necessary information to use this method. Since we have run 5000 iterations, R shows the diagnostic results. Otherwise, it will show an error saying that a minimum of 3746 iterations are needed. Small burn-in simulations such as here ($M = 2$) suggest that the chain is becoming stationary quite fast, which is consistent with the traceplot. 3741 in Total (N) is the total number of iterations we should run to estimate the 2.5th percentile (q) of the posterior distribution to the specified accuracy (0.005) and probability (0.95), which means we do not need any more iterations (in fact, in this case $N < N_{min}$). The dependence factor I indicates the extent to which autocorrelation inflates the required sample. $I = 0.999$ indicates that our simulations are not autocorrelated, also seen by the consistent movement of the chains about the parameter space.

Heidelberg and Welchs' Diagnostic

Lauren Goodwin 2014
Kevin Ferris 2013

Introduction

The Heidelberg and Welch diagnostic is performed on a single chain and consists of two parts. The first part of the diagnostic uses a Cramer-von-Mises statistic to test the null hypothesis that the chain comes from a stationary distribution. The second part of the diagnostic determines if the chain needs more iterations to make accurate inferences.

Summary of Method

Part One

To test if the chain comes from a stationary distribution we set up the null hypothesis, the chain is from a stationary distribution, and perform the following:

1. Calculate the Cramer-von-Mises statistic on the whole chain. If the null hypothesis fails to be rejected at a 5% significance level then, we conclude that the chain comes from a stationary distribution and the first part is passed.
2. If the null hypothesis is rejected then the first 10% of the chain is discarded and the Cramer-von-Mises statistic is computed again. If the null hypothesis fails to be rejected then we conclude that the chain comes from a stationary distribution and the first part is passed.
3. If the null hypothesis is rejected then the next 10% of the chain is discarded and the test is performed again.
4. The process is repeated until the null hypothesis fails to be rejected or 50% of the chain is discarded.
5. If 50% of the chain is discarded, the first part of the diagnostic fails, and we need to run the chain longer.

Part Two

If the first part of the diagnostic is passed, the second part takes the portion of the chain that has not been discarded and determines if the size of the chain is large enough to make accurate inferences about the posterior distribution.

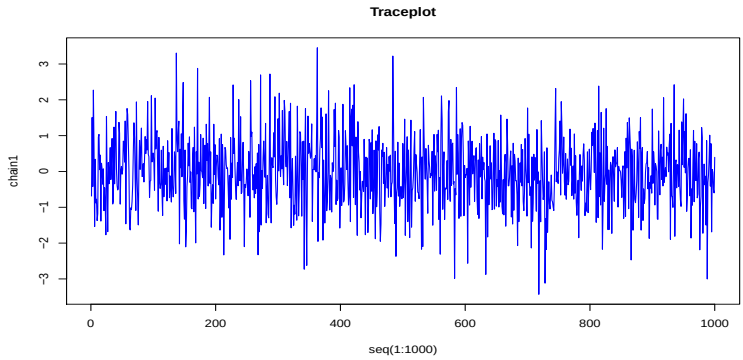
1. The halfwidth test is performed to calculate the half width of the $(1 - \alpha)\%$ posterior interval for the mean.
2. If the ratio of the half width and the posterior mean is lower than some ϵ, then the second part of the diagnostic passes. If not, the second part of the diagnostic fails and the chain needs to run longer.

Example

Scenario 1

One chain with 1000 draws from a $N(0,1)$.

```
chain1<-mcmc(rnorm(1000,0,1))
```




```
heidel.diag(chain1)
```

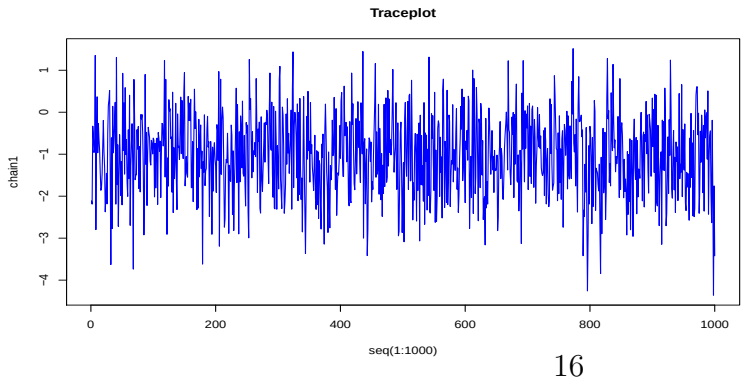
```
     Stationarity start     p-value
     test         iteration
var1 passed       401       0.0509

     Halfwidth Mean  Halfwidth
     test
var1 failed    0.111 0.0764
```

The first part of the diagnostic is passed on the whole chain, and we can conclude the chain comes from a stationary distribution. The second part of the diagnostic fails, so the chain needs run longer.

Scenario 2

One chain with 1000 draws from a $N(-1,1)$.

```
chain1<-mcmc(rnorm(1000,-1,1))
```



```
heidel.diag(chain1)
```

```
     Stationarity start     p-value
     test         iteration
var1 passed       1         0.173

     Halfwidth Mean  Halfwidth
     test
var1 passed    -1.03 0.0607
```

The first part of the diagnostic is passed on the whole chain, and we can conclude the chain comes from a stationary distribution. The second part of the diagnostic passes on the whole chain.

Hellinger Distance

Jay Rosencrantz, Bobby Hsu, Koji Toma

2014

1 Introduction

The Hellinger Distance approach can be used as a diagnostic tool for comparison of multiple chains as an assessment of convergence. We are able to use a sampling-based estimate of the Hellinger distance between two distributions in order to assess similarity (and thus convergence) between two distributions, which can be thought of as chains in the MCMC setting. As a general rule, we know that the distance will be *small* for very similar distributions and *large* for very dissimilar distributions. Thus the Hellinger distance is bounded: $0 \leq H(f,g) \leq 1$, where $H(f,g)$ is the Hellinger distance between two probability distributions f and g, which is explicitely as:

$$H(f,g) = \sqrt{\frac{1}{2}\int(\sqrt{f(x)} - \sqrt{g(x)})^2 dx}$$

While it is possible to split a single chain into multiple "sub-chains" in order to assess convergence of a single chain, we will focus on multiple chain comparisons, which can easily be extended to a single chain broken into pieces.

2 Practical Approach

The practical approach to calculating and using Hellinger distance as a powerful diagnostic tool can be broken down into three steps: (1) obtain simulations from the posterior distribution for multiple chains, (2) estimate a continuous probability distribution for each chain by finding the approximate density curve associated with each chain, and (3) calculate the resulting approximate Hellinger's distance for each possible chain comparison.

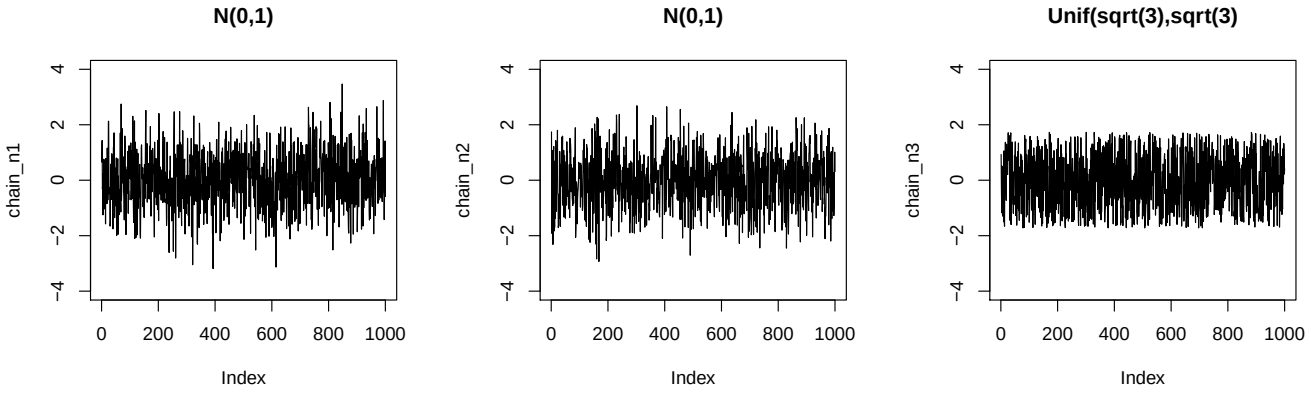
Once we have a list of Hellinger's distance for each possible chain combination, we can consider whether these distances are large enough to indicate dissimilarities between chains or not. A potential problem with this idea is discussed in Section 4.

3 Example

We set up 3 chains with 1000 simulated observations each. Two chains were simulated from a standard Normal distribution, and the other was from an uniform($\sqrt{3}$,$\sqrt{3}$). The traceplot for each chain was shown below. We compared all 3 chains with each other using Hellinger distance. Chains 1 and 2 were both from a N(0,1) distribution, had a small Hellinger distance. Comparing chains 1 and 3, which came from different distributions, the Hellinger distance was large (the same for chains 2 and 3).

```
chain_n1 <- rnorm(1000)
chain_n2 <- rnorm(1000)
chain_n3 <- runif(1000, -sqrt(3), sqrt(3))

plot(chain_n1, type = "l", ylim = c(-4, 4), main="N(0,1)")
plot(chain_n2, type = "l", ylim = c(-4, 4), main="N(0,1)")
plot(chain_n3, type = "l", ylim = c(-4, 4), main="Unif(sqrt(3),sqrt(3)")

chain_n1_hat <- density(chain_n1, from = -4, to = 4)
chain_n2_hat <- density(chain_n2, from = -4, to = 4)
chain_n3_hat <- density(chain_n3, from = -4, to = 4)
```

```r
dx <- diff(chain_n1_hat$x)[1]

H12 <- sqrt(1 / 2 * sum(dx * (sqrt(chain_n1_hat$y) - sqrt(chain_n2_hat$y))^2))
H13 <- sqrt(1 / 2 * sum(dx * (sqrt(chain_n1_hat$y) - sqrt(chain_n3_hat$y))^2))
H23 <- sqrt(1 / 2 * sum(dx * (sqrt(chain_n2_hat$y) - sqrt(chain_n3_hat$y))^2))
```




```
## Hellinger distance for chain 1 and 2 = 0.04350892
## Hellinger distance for chain 1 and 3 = 0.1732152
## Hellinger distance for chain 2 and 3 = 0.1602982
```

Using Gelman diagnostic for the above 3 chains, the $\hat{R}$ value was 1, which did not detect the differences between chains. In this case, using Hellinger distance, we were able to tell the differences between chains, which indicated that there might be problems in convergence.

```
## Potential scale reduction factors:
##
##      Point est. Upper C.I.
## [1,]          1          1
```

4 Problems

When examining Hellinger's distances, one has to ask the question, "how large is too large?". The answer is that the distance is somewhat arbitrary, but literature indicates that (similar to a p-value with $\alpha = .05$), we can assume the chains are similar if the distance is less than 0.05.

5 References

1. Boone, Edward L., Jason R.w. Merrick, and Matthew J. Krachey. "A Hellinger Distance Approach to MCMC Diagnostics." Journal of Statistical Computation and Simulation 84.4 (2014): 833-49. Web.

2. "Hellinger Distance." Encyclopedia of Mathematics. N.p., n.d. Web. 11 Nov. 2014.

Quantile Equivalence Diagnostic

Maya, Lauren, and Marie
Fall 2014

Introduction and Motivation

The Quantile Equivalence Diagnostic (QED) is a diagnostic that Michael Lerch, a PhD student in our department, is proposing in his research. The motivation for the QED builds on the idea that quantiles are often of interest to the researcher. It borrows from some existing diagnostics such as the PSRF, Boone's Hellinger, and Raftery and Lewis'. Specifically, the QED, as the PSRF and Boone's Hellinger, develops on the belief that the final distribution is more important than the sampling path, and that running multiple chains provides power in assessing convergence. The QED also follows Raftery and Lewis' path of addressing tail quantiles and probabilties explicitly.

Summary of Method

The QED aims at two similar inferential objectives: to find quantiles, and to find probabilities. It relies on an equivalence testing framework to assess convergence. To introduce the QED for quantiles, an example from Michael Lerch's working paper *"Equivalence testing for MCMC convergence assessment"* is provided below. A method of calculation then follows, as well as a brief discussion of the needed adaptation when the objective is to find probabilities.

Quantiles

Consider the following example:"Suppose that the quantity of interest, or one of a suite of quantities of interest, is a quantile of a posterior parameter ψ . Let's consider the 0.025 quantile of ψ, perhaps one endpoint of a credible interval, and denote it as $Q^{0.025}$. $Q^{0.025}$ is, in general, a specific value defined by the posterior density, $p(\psi \mid y)$. A standard procedure to estimate this value when using MCMC is to run multiple chains and then produce the empirical quantile from the amalgamation of these chains. We will denote this estimate as $\hat{Q}^{0.025}$. Now, consider $\hat{p}_i$ to be the proportion of samples in the i^{th} chain that are smaller than $\hat{Q}^{0.025}$. Ideally, $\hat{p}_i$ would each be equal to 0.025. Were this true, we would be rather confident in $\hat{Q}^{0.025}$ in being a good estimate of $Q^{0.025}$..."

The actual chances of $\hat{p}_i$=0.0255, however, are very small. We specify a tolerence level ϵ such that, as long as $|0.025\text{-}p_i| < \epsilon$, we are content. Therefore, a test assessing, how likely is that $|0.025\text{-}p_i| < \epsilon$, is required. To perform this test, samples within a chain are given a 0 or a 1, creating the proportion $\hat{p}_i$. Specifically, 0 is assigned to samples greater than $\hat{Q}^p$ and 1 is assigned to samples less than or equal to $\hat{Q}^p$. We can then use a normal approximation to the binomial distributon, in which case $\hat{p}_i \sim N\left(p_i, \sqrt{\frac{p(1-p)}{n}}\right)$, when approximating the

unknown p_i and with known p. This allows for a frequentist flavored hypothesis test,

$$H_0\text{: } \mid p - p_i \mid > \epsilon \text{ for some } i$$

$$H_a\text{: } \mid p - p_i \mid \leq \epsilon \text{ for all } i.$$

The default assumption is that the chains are *not* in agreement, and evidence is needed to conclude that the chains *are* in agreement. This idea is counterintuitive for the classic frequentist version, and is given the name of *Equivalence Testing*. The evidence is assessed using $\hat{p}_i$, with the user specifying p, margin of error ϵ, and significance level α.

Decision Rule

A Uniformly Most Powerful α level test rejects the null hypothesis when,

$\mid p - p_i \mid \sqrt{n/(p(1-p))} < \sqrt{\chi^2_{\alpha,1}}$,

and a non-centrality parameter of $\sqrt{\frac{n}{p(1-p)}}\epsilon$. The test works via the intersection-union principle, and is run for each chain. If, for each chain, the null hypothesis rejected, the QED suggests that both convergence and desired precision have been obtained. If, however, for at least one chain, the null hypothesis is not rejected, the QED suggests continued sampling should be performed to reach convergence and/or the desired precision. The QED thus assesses convergence by consistency across chains.

Probabilities

When the objective is to find posterior probabilities such as $p(\psi < Q^p|y) = p$, rather than quantiles, p is unknown while Q^p is known (the opposite is true for the quantile objective). To adapt the QED to this case, Q^p is replaced with the specified value, and the known value of p is replaced with its estimate $\hat{p}$. The estimate $\hat{p}$ is the resulting proportion of samples that are less that Q^p.

Graphical Display

To graphically compliment the QED, the Quantile Equivalence Plot may be used. The QEP does not aim at replacing a sample path plot, but only in supplementing it in the case that the researcher is interested in the inferences described above. The steps to create a QEP are:

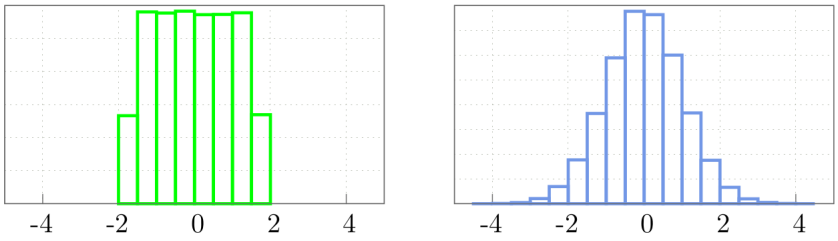
- Plot empirical quantiles against probabilities
- Plot overall empirical quantile for specified probability
- Empirical quantiles will be different in each chain. Assess the discrepancies
- At specified probability, plot each chains empirical quantile
- At overall empirical quantile, plot each chains observed probability
- Connect within a chain with a line

Discussion/Concerns

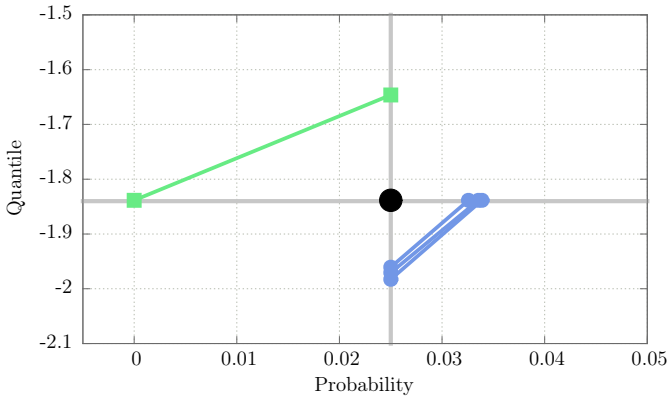
A hypothesis test in a Bayesian setting may be surprising. A criticism of hypothesis tests is that the null hypothesis is "trivial". However, the null hypothesis used here is non-trivial. Thus, using a hypothesis test in this situation is deemed appropriate. Another consideration is the frequentist definition of probability as a long frequency. Because data is generated by a random number generator, this long run definition of probability is well suited in this situation.

Example

Lets draw one chain from a $uniform(-\sqrt{3}, \sqrt{3})$ distrubtion and another chain from a $N(0, 1)$ distribution. These two distributions have the same mean and variance but look very different.



When using the QEP for the .025 quantile:



If we set $\epsilon = 0.005$ and $\alpha = .1$, then the QED fails. So we conclude that the chains are not in agreement. If we set $\epsilon = 0.05$ and $\alpha = .1$, then the QED fails. So we conclude that the chains are not in agreement.