Math 640: Bayesian Time Series (group project)

Adam Bouyamourn, Michael Bowerman, Hiep Nguyen

April 2018

1 Introduction

We extend the material covered in class to the time series context. We derive an AR(1) model in the Bayesian context and use Gibbs sampling to draw from the posterior using our collected data. For the vector of autoregressive parameter ϕ, we present first a model with a truncated normal prior for ϕ with support $\{-1, 1\}$ before deriving a model with a normal prior centered at μ with variance σ_ϕ^2. In the first model, it is assumed *a priori* that the time series is stationary (i.e. that it has a unit root, $|\phi| < 1$). In the second case, using a non-truncated normal prior, it is not immediately assumed that the AR(1) model is stationary. Here, we suggest that this model could be implemented and treated as a unit root test.

2 Data

We examine three datasets in our project. The first data set is the global mean land-ocean temperature deviations from 1951 to 1980 (30 years). We have first-differenced the data, and assess that it is well modelled by an AR(1) process.

The second dataset contains real Brazilian monthly GDP between 01/1980 and 12/1997, for a total of 216 months. The true model for this data is MA(1), and we include it in order to assess the performance of the Bayesian AR(1) model in the context of known model misspecification.

The last dataset contains nominal Brazilian annual GDP in 91 years, from 1900 to 1990. This is an explosive time series, for which we know that $\phi > 1$.

The first dataset is accessible via the package 'astsa' in R. The second and third dataset were downloaded from `http://www2.stat.duke.edu/~mw/data-sets/ts_data/brazil_econ`.

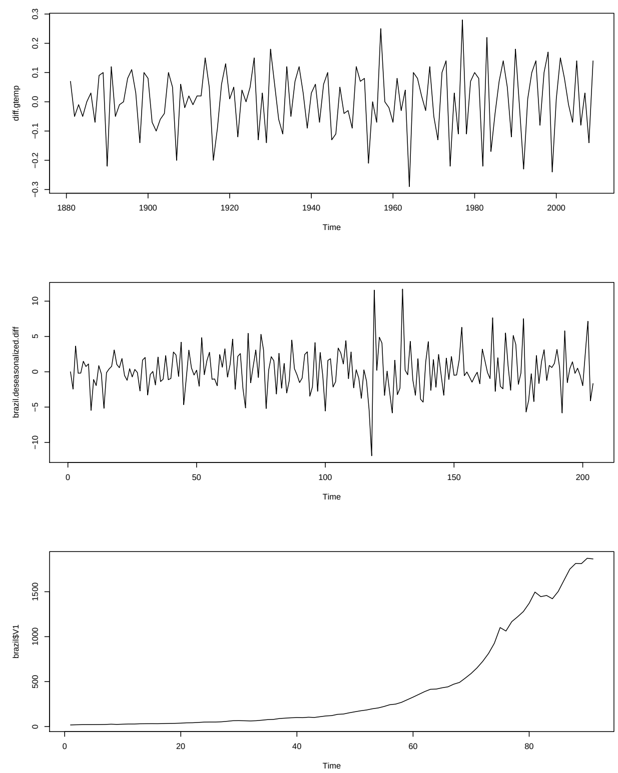

Figure 1: Time series visualization of three datasets: differenced global temperature data; differenced real Brazilian GDP data; nominal Brazilian GDP data

3 Derivation of model

Let $\{Y_t : t \geq 1\}$ be a sequence of random variable. Consider a stationary AR(1) process, which is given by

$$y_t = c + \phi y_{t-1} + \epsilon_t,$$

where y_k is the value of of the time series at time k and $k \in \mathbb{N}$, ϕ is the autoregressive parameter, ϵ_t is the white noise at time t and c is a constant. Now, consider the likelihood function of the complete sample for the AR(1), which is given by

$$\mathcal{L}(\mathbf{y}|\phi, \sigma_\varepsilon^2) = \prod_{t=2}^{T} \frac{1}{\sqrt{2\pi\sigma_\varepsilon^2}} \exp\Big[\frac{-(y_t - \phi y_{t-1})^2}{2\sigma_\varepsilon^2}\Big]$$

Assuming a truncated normal prior for ϕ, with bounds (-1,1) centered at μ with variance σ_ϕ^2, we have

$$p(\phi|\mathbf{y}, \sigma_\varepsilon^2, \mu, \sigma_\phi^2) \propto \prod_{t=2}^{T} \frac{1}{\sqrt{2\pi\sigma_\varepsilon^2}} \exp\Big[\frac{-(y_t - \phi y_{t-1})^2}{2\sigma_\varepsilon^2}\Big] \Big[\frac{\frac{1}{\sqrt{2\pi\sigma_\phi^2}} \exp\frac{-(\phi-\mu)^2}{2\sigma_\phi^2}}{\varphi(\frac{1-\mu}{\sigma_\phi}) - \varphi(\frac{-1-\mu}{\sigma_\phi})}\Big],$$

where $\varphi(\bullet)$ denotes the standard normal distribution. Now,

$$\begin{aligned}
p(\phi|\mathbf{y},\sigma_\varepsilon^2,\mu,\sigma_\phi^2) &\propto \exp\Big[\frac{-1}{2\sigma_\varepsilon^2}\sum_{t=2}^{T}(y_t-\phi y_{t-1})^2\Big]\exp\Big[\frac{-(\phi-\mu)^2}{2\sigma_\phi^2}\Big] \\
&\propto \exp\Big[\frac{-1}{2\sigma_\varepsilon^2}\sum_{t=2}^{T}(y_t-\phi y_{t-1})^2-\frac{1}{2\sigma_\phi^2}(\phi-\mu)^2\Big] \\
&= \exp\Big[\frac{-1}{2\sigma_\varepsilon^2}\sum_{t=2}^{T}(-2\phi y_t y_{t-1}+\phi^2 y_{t-1}^2)-\frac{1}{2\sigma_\varepsilon^2}(\phi-\mu)^2\Big] \\
&\propto \exp\Big[\frac{\sum_{t=2}^{T}y_t y_{t-1}}{\sigma_\varepsilon^2}\phi-\frac{1}{2\sigma_\varepsilon^2}\sum_{t=2}^{T}y_{t-1}^2\phi^2-\frac{1}{2\sigma_\phi^2}\phi^2+\frac{\mu}{\sigma_\phi^2}\phi\Big] \\
&= \exp\Big[\phi\Big(\frac{\sum y_t y_{t-1}}{\sigma_\varepsilon^2}+\frac{\mu}{\sigma_\phi^2}\Big)-\phi^2\Big(\frac{\sum y_{t-1}^2}{2\sigma_\varepsilon^2}+\frac{1}{2\sigma_\phi^2}\Big)\Big] \\
&= \exp\Big[-\frac{1}{2}\Big(\frac{\sum y_{t-1}^2}{\sigma_\varepsilon^2}+\frac{1}{\sigma_\phi^2}\Big)\Big(\phi^2-2\phi\Big(\frac{\sum y_t y_{t-1}}{\sigma_\varepsilon^2}+\frac{\mu}{\sigma_\phi^2}\Big)\Big(\frac{\sum y_{t-1}^2}{\sigma_\varepsilon^2}+\frac{1}{\sigma_\phi^2}\Big)^{-1}\Big)\Big] \\
&\propto \exp\Big[\frac{\sum_{t=2}^{T}y_t y_{t-1}}{\sigma_\varepsilon^2}\phi-\frac{1}{2\sigma_\varepsilon^2}\sum_{t=2}^{T}y_{t-1}^2\phi^2-\frac{1}{2\sigma_\phi^2}\phi^2+\frac{\mu}{\sigma_\phi^2}\phi\Big] \\
&= \exp\Big[\phi\Big(\frac{\sum y_t y_{t-1}}{\sigma_\varepsilon^2}+\frac{\mu}{\sigma_\phi^2}\Big)-\phi^2\Big(\frac{\sum y_{t-1}^2}{2\sigma_\varepsilon^2}+\frac{1}{2\sigma_\phi^2}\Big)\Big] \\
&= \exp\Big[-\frac{1}{2}\Big(\frac{\sum y_{t-1}^2}{\sigma_\varepsilon^2}+\frac{1}{\sigma_\phi^2}\Big)\Big(\phi^2-2\phi\Big(\frac{\sum y_t y_{t-1}}{\sigma_\varepsilon^2}+\frac{\mu}{\sigma_\phi^2}\Big)\Big(\frac{\sum y_{t-1}^2}{\sigma_\varepsilon^2}+\frac{1}{\sigma_\phi^2}\Big)^{-1}\Big)\Big] \\
&\propto \exp\Big[\frac{-1}{2\big(\frac{\sum y_{t-1}^2}{\sigma_\varepsilon^2}+\frac{1}{\sigma_\phi^2}\big)^{-1}}\Big(\phi-\big(\frac{\sum y_t y_{t-1}}{\sigma_\varepsilon^2}+\frac{\mu}{\sigma_\phi^2}\big)\big(\frac{\sum y_{t-1}^2}{\sigma_\varepsilon^2}+\frac{1}{\sigma_\phi^2}\big)^{-1}\Big)^2\Big]
\end{aligned}$$

Thus, $\phi|\mathbf{y},\mu,\sigma_\varepsilon^2,\sigma_\phi^2 \sim TN_{(-1,1)}\Big(\big(\frac{\sum y_t y_{t-1}}{\sigma_\varepsilon^2}+\frac{\mu}{\sigma_\phi^2}\big)\big(\frac{\sum y_{t-1}^2}{\sigma_\varepsilon^2}+\frac{1}{\sigma_\phi^2}\big)^{-1},\big(\frac{\sum y_{t-1}^2}{\sigma_\varepsilon^2}+\frac{1}{\sigma_\phi^2}\big)^{-1}\Big)$, which is also how the AR(1) parameter is estimated in the literature.
If we use a normal prior on ϕ instead of a truncated normal,we would have

$$\phi|\mathbf{y},\mu,\sigma_\varepsilon^2,\sigma_\phi^2 \sim N\Big(\big(\frac{\sum y_t y_{t-1}}{\sigma_\varepsilon^2}+\frac{\mu}{\sigma_\phi^2}\big)\big(\frac{\sum y_{t-1}^2}{\sigma_\varepsilon^2}+\frac{1}{\sigma_\phi^2}\big)^{-1},\big(\frac{\sum y_{t-1}^2}{\sigma_\varepsilon^2}+\frac{1}{\sigma_\phi^2}\big)^{-1}\Big)$$

Further assume the flat prior on μ and the non-informative priors for σ_ε^2 and σ_ϕ^2, $\mu \sim Unif(-50,50), \pi(\sigma_\varepsilon^2)\propto\frac{1}{\sigma_\varepsilon^2}$, and $\pi(\sigma_\phi^2)\propto\frac{1}{\sigma_\phi^2}$. We have

$$p(\phi,\mu,\sigma_\varepsilon^2,\sigma_\phi^2|\mathbf{y}) \propto \prod_{t=2}^{T}\Big(\frac{1}{\sqrt{2\pi\sigma_\varepsilon^2}}\exp\Big[\frac{-(y_t-\phi y_{t-1})^2}{2\sigma_\varepsilon^2}\Big]\Big)\Big(\frac{1}{\sqrt{2\pi\sigma_\phi^2}}\exp\Big[\frac{-(\phi-\mu)^2}{2\sigma_\phi^2}\Big]\Big)\Big(\frac{1}{2}\Big)\Big(\frac{1}{\sigma_\varepsilon^2}\Big)\Big(\frac{1}{\sigma_\phi^2}\Big)$$

. The full conditional for μ is

$$p(\mu|\phi,\sigma_\varepsilon^2,\sigma_\phi^2,\mathbf{y}) \propto \exp\Big[\frac{-(\phi-\mu)^2}{2\sigma_\phi^2}\Big],$$

which implies that $\mu|\phi, \sigma^2_\varepsilon, \sigma^2_\phi, \mathbf{y} \sim TN_{(-50,50)}(\phi, \sigma^2_\phi)$.
Now,

$$p(\sigma^2_\varepsilon|\phi, \mu, \sigma^2_\phi, \mathbf{y}) \propto \left(\sigma^2_\varepsilon\right)^{-(n/2+1)}\exp\Big[-\frac{1}{2}\sum_{t=2}^{T}\left(y_t - \phi y_{t-1}\right)^2\frac{1}{\sigma^2_\varepsilon}\Big],$$

which implies that $\sigma^2_\varepsilon|\phi, \mu, \sigma^2_\phi, \mathbf{y} \sim IG\Big(\frac{n}{2}, \frac{1}{2}\sum_{t=2}^{T}(y_t - \phi y_{t-1})^2\Big)$.
In addition,

$$p(\sigma^2_\phi|\phi, \mu, \sigma^2_\varepsilon, \mathbf{y}) \propto \left(\sigma^2_\phi\right)^{-(1/2+1)}\exp\Big[-\frac{1}{2}\left(\phi - \mu\right)^2\frac{1}{\sigma^2_\phi}\Big],$$

which means $\sigma^2_\phi|\phi, \mu, \sigma^2_\varepsilon, \mathbf{y} \sim IG\Big(\frac{1}{2}, \frac{1}{2}(\phi - \mu)^2\Big)$.
Because the full conditional posterior distributions are recognizable for each parameter in the model, we can implement a Gibbs sampler to draw samples from the posteriors. In the following section, we fit the model using the normal prior on ϕ with both stationary and non-stationary data, to demonstrate the ability of the model to both fit time series data and to detect deviations from stationarity.

4 Results

Global temperature data (true AR(1) process)

Parameter estimates

After sampling 20,000 draws from this posterior distribution, assuming the priors as shown in the derivation section, we provide credible intervals in Table 1 below.

Table 1:

	2.5%	50%	97.5%
ϕ	-0.304	-0.292	-0.281
μ	-0.304	-0.292	-0.272
σ^2_ϵ	0.009	0.011	0.014
σ^2_ϕ	0	0	0.025

Frequentist parameter estimates

Running a frequentist model using the R command ARMA, we estimate the following parameters:

$$Y_t = 0.007 + -0.298 Y_{t-1}$$

Our Bayesian parameter estimates are close to the frequentist estimates. Notice, however, in Figure 3, that the point estimate is towards the left tail of our estimated distribution for ϕ.

Posterior density plots

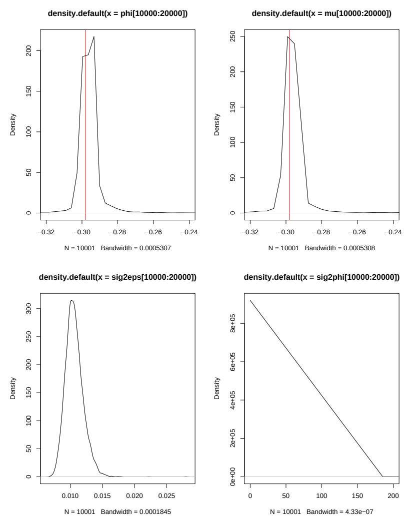

Figure 2: Density plots of the parameter estimates, frequentist point estimate in red

Diagnostics

Running plots

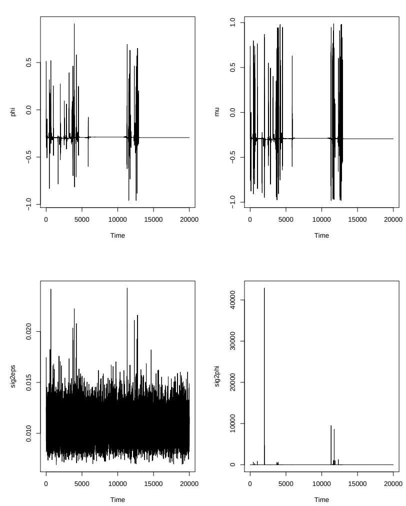

Figure 3: Running plots of the parameter estimates

PACF plots

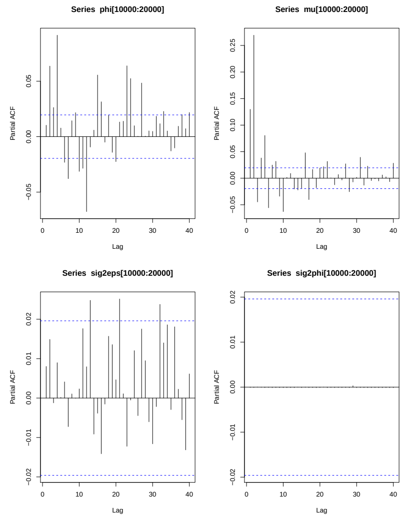

Figure 4: Partial autocorrelation plots of parameter estimates

Running the Geweke diagnostic tests, having discarded the first 10000 draws, we find Geweke z-scores of 1.756 for ϕ, .85 for μ, 1.49 for σ_ϵ^2 and -.928 for σ_ϕ^2. None of these is significant at the 5% level of significance.

Splitting our draws into four chains, we run the Gelman diagnostic test for ϕ, suggesting a scale reduction factor of 1.02, which is below the rule-of-thumb threshold of 1.2. Running the Gelman diagnostic on μ, we report a scale reduction of 1, which is also below the threshold of concern.

First-differenced Brazil GDP data (true MA(1) process)

Parameter estimates

We sample 20,000 draws from the posterior distribution outline above, estimated on first-differenced Brazilian GDP data.

Table 2:

	2.5%	50%	97.5%
ϕ	-0.199	-0.191	-0.181
μ	-0.203	-0.191	-0.168
σ^2_ϵ	7.298	8.559	10.471
σ^2_ϕ	0	0	0.063

Our frequentist model suggests the following parameter estimates:

$$Y_t = .2525 + -0.1974Y_{t-1}$$

Which is again fairly similar to the parameter estimates in the Bayesian case.

Density plots

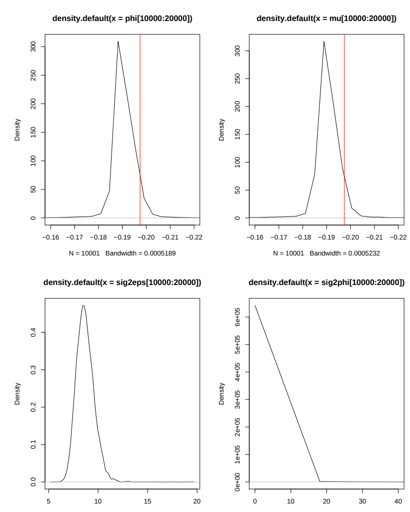

Figure 5: Density plots of the parameter estimates, frequentist point estimate in red

Diagnostics

Running plots

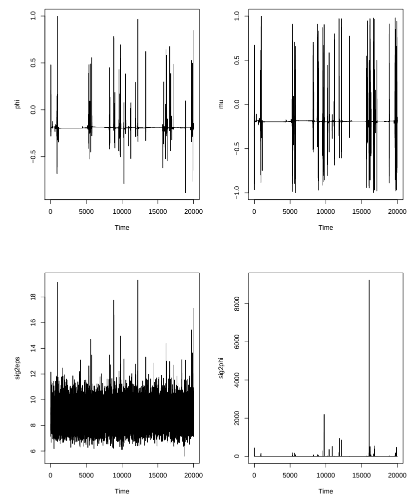

Figure 6: Running plots of the parameter estimates

PACF plots

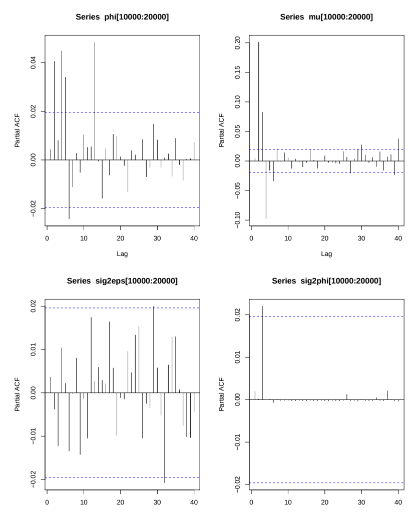

Figure 7: Partial autocorrelation plots of parameter estimates

Running the Geweke diagnostic tests, having discarded the first 10000 draws, we find Geweke z-scores of -0.9401 for ϕ, -1.183 for μ, -0.04679 for σ_ϵ^2 and -1.233 for σ_ϕ^2. None of these is significant at the 5% level of significance.

Splitting our draws into four chains, we run the Gelman diagnostic test for ϕ, suggesting a scale reduction factor of 1.01, which is below the rule-of-thumb threshold of 1.2. Running the Gelman diagnostic on μ, we report a scale reduction of 1, which is also below the threshold of concern.

Nominal Brazil GDP data (explosive AR process)

Parameter estimates

We sample 20,000 draws from the posterior distribution outline above, estimated on first-differenced Brazilian GDP data.

Table 3: Parameter estimates for explosive AR process

	2.5%	50%	97.5%
ϕ	1.041	1.041	1.041
μ	1.041	1.041	1.041
σ^2_ϵ	803.903	$1,048.352$	$1,420.113$
σ^2_ϕ	0	0	0

The frequentist model estimates the following parameters:

$$Y_t = 6.283 + 1.035 Y_{1-1}$$

In both cases, $|\phi| > 1$, which suggest that the autoregressive process is explosive, as we can determine from Figure 1.

Density plots

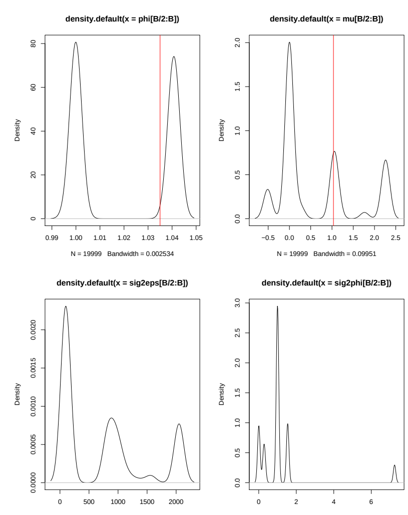

Figure 8: Density plots of the parameter estimates, frequentist point estimate in red

Diagnostics

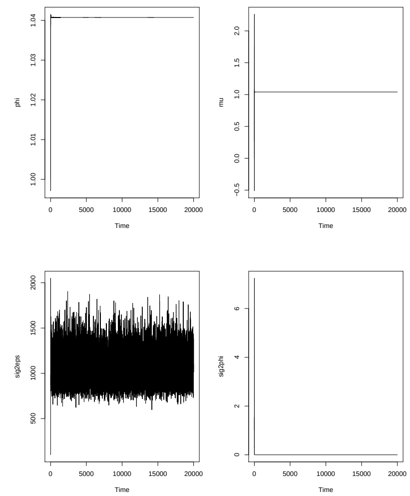

Figure 9: Running plots of the parameter estimates

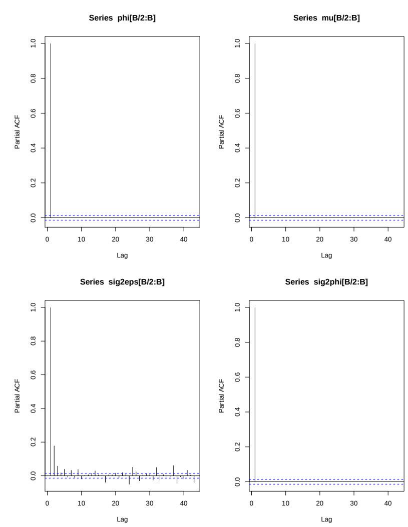

Figure 10: Partial autocorrelation plots of parameter estimates

Running the Geweke diagnostic tests, having discarded the first 10000 draws, we find Geweke z-scores of -1.073 for ϕ, -1.002 for μ, 0.4346 for σ_ϵ^2 and 1.012 for σ_ϕ^2. None of these is significant at the 5% level of significance.

Splitting our draws into four chains, we run the Gelman diagnostic test for ϕ, suggesting a scale reduction factor of 1.06, which is below the rule-of-thumb threshold of 1.2. Running the Gelman diagnostic on μ, we report a scale reduction of 1.06, which is also below the threshold of concern.

5 Conclusions

The case is made here that *a priori* assumptions about stationarity are not necessary to frame ARMA-type models in the Bayesian context. A model was constructed using a non-truncated normal prior on the AR parameter, and it was shown that this model provides a comparable fit to data to the typical frequentist parameter. The model was shown to be robust to the underlying data generation process, as an acceptable model fit was found to data coming from a stationary AR(1) process, a stationary MA(1) process, and a non-stationary AR(1) process.

Of particular interest is the fit to the nominal Brazil GDP data, which was non-stationary. The Bayesian AR(1) model presented here was able to detect the non-unit AR parameter, which in turn suggests that the data is coming from a non-stationary process. We suggest that neither *a priori* stationarity assumptions nor frequentist unit root tests are necessary in evaluating time series data in the Bayesian context.

In future research, the power of the model in testing for unit roots could be assessed using empirical studies. The extension of this model, using a similar construction of prior distributions, to any ARMA(p,q) model would be relatively straightforward, relying on conditional independence to construct the likelihood function. Of particular interest would be the ability to detect unit roots in this general case.

References

[1] H. M. Karakani, J. Niekerk, Paul. Staden. "Bayesian Analysis of AR(1) Model."`arXiv:1611.08747`

6 Code Appendix

```
require(msm)
library(MCMCpack)
library(forecast)
library(stargazer)
library(tseries)
set.seed(2020)

x <- arima.sim(model = list(order = c(1, 0, 0), ar = .3), n = 100)
arma(x,p=1,q=0)
pacf(x)

yt       <- x
ytmin1   <- yt[c(1:(length(yt)-1))]
yt       <- yt[c(2:length(yt))]
prod     <- yt%*%ytmin1
prod     <- prod[1]
ytsq     <- sum(ytmin1^2)
```

```
n        <- length(yt)

B        <- 1000
phi      <- vector("numeric", B)
mu        <- vector("numeric", B)
sig2eps    <- vector("numeric", B)
sig2phi    <- vector("numeric", B)

phi[1]            <- mean(yt)
mu[1]             <- 0
sig2eps[1]        <- var(yt)
sig2phi[1]        <- 1

for(t in 2:B){
  ### sample phi ###
  phi[t] <- rnorm(1, (prod/sig2eps[t-1]+mu[t-1]/sig2phi[t-1])*(
  ytsq/sig2eps[t-1]+1/sig2phi[t-1])^-1,(
  ytsq/sig2eps[t-1]+1/sig2phi[t-1])^-1)
  ### sample mu ###
  mu[t] <- rtnorm(1,phi[t-1],sig2phi[t-1],-1,1)
  ### sample sig2eps ###
  sig2eps[t] <- rinvgamma(1,n/2,(1/2)*sum((yt-phi[t-1]*ytmin1)^2))
  ### sample sig2phi ###
  sig2phi[t] <- rinvgamma(1,1/2,(1/2)*(phi[t-1]-mu[t-1])^2)
}

par(mfrow=c(2,2))

plot.ts(phi)
plot.ts(mu)
plot.ts(sig2eps)
plot.ts(sig2phi)

##############

library(astsa)
require(astsa)

data(gtemp)

plot.ts(gtemp)

diff.gtemp <- diff(gtemp)

##############
```

```
yt       <- diff.gtemp
ytmin1   <- yt[c(1:(length(yt)-1))]
yt       <- yt[c(2:length(yt))]
prod     <- yt%*%ytmin1
prod     <- prod[1]
ytsq     <- sum(ytmin1^2)
n        <- length(yt)

B        <- 20000
phi      <- vector("numeric", B)
mu        <- vector("numeric", B)
sig2eps    <- vector("numeric", B)
sig2phi    <- vector("numeric", B)

phi[1]           <- mean(yt)
mu[1]            <- 0
sig2eps[1]       <- var(yt)
sig2phi[1]       <- 1

for(t in 2:B){
  ### sample phi ###
  phi[t] <- rnorm(1, (prod/sig2eps[t-1]+mu[t-1]/sig2phi[t-1])*(
  ytsq/sig2eps[t-1]+1/sig2phi[t-1])^-1,(
  ytsq/sig2eps[t-1]+1/sig2phi[t-1])^-1)
  ### sample mu ###
  mu[t] <- rtnorm(1,phi[t-1],sig2phi[t-1],-1,1)
  ### sample sig2eps ###
  sig2eps[t] <- rinvgamma(1,n/2,(1/2)*sum((yt-phi[t-1]*ytmin1)^2))
  ### sample sig2phi ###
  sig2phi[t] <- rinvgamma(1,1/2,(1/2)*(phi[t-1]-mu[t-1])^2)
}
warnings()

par(mfrow=c(2,2))

plot.ts(phi)
plot.ts(mu)
plot.ts(sig2eps)
plot.ts(sig2phi)

pacf(phi[10000:20000])
pacf(mu[10000:20000])
pacf(sig2eps[10000:20000])
```

```
pacf(sig2phi[10000:20000])

plot(density(phi[10000:20000]))
plot(density(mu[10000:20000]))
plot(density(sig2eps[10000:20000]))
plot(density(sig2phi[10000:20000]))


geweke.diag(phi)
geweke.diag(mu)
geweke.diag(sig2eps)
geweke.diag(sig2phi)

gelman.diag(phi)
gelman.diag(mu)
gelman.diag(sig2eps)
gelman.diag(sig2phi)


most.thin <- rep(c(rep(F, 19), T), 1000)

summary(most.thin)
length(phi)

pacf(phi[most.thin])
pacf(mu[most.thin])
pacf(sig2eps[most.thin])
pacf(sig2phi[most.thin])

tab <- t(cbind(quantile(phi[most.thin],     probs=c(.025, .5, .975)),
               quantile(mu[most.thin],      probs=c(.025, .5, .975)),
               quantile(sig2eps[most.thin], probs=c(.025, .5, .975)),
               quantile(sig2phi[most.thin], probs=c(.025, .5, .975))))

stargazer(tab)

model.frequentist <- arma(diff.gtemp, order = c(1,0))

###################

# Deseasonalized Brazil GDP

###################
```

```
brazil2 <- read.csv("./brazil_predictors.txt", header = T)

head(brazil2)

brazil.GDP <- ts(brazil2$GDP, frequency=12)

decomposed.brazil <- decompose(brazil.GDP)

decomposed.brazil

brazil.deseasonalized <- na.omit(decomposed.brazil$trend + decomposed.brazil$random)

plot.ts(brazil.deseasonalized)

brazil.deseasonalized.diff <- vector("numeric", length(brazil.deseasonalized))

for(i in 2:length(brazil.deseasonalized)){

  brazil.deseasonalized.diff[i] <-
  brazil.deseasonalized[i]-lag(brazil.deseasonalized[i-1], 1)

}



yt        <- brazil.deseasonalized.diff
ytmin1    <- yt[c(1:(length(yt)-1))]
yt        <- yt[c(2:length(yt))]
prod      <- yt%*%ytmin1
prod      <- prod[1]
ytsq      <- sum(ytmin1^2)
n         <- length(yt)

B         <- 20000
phi       <- vector("numeric", B)
mu         <- vector("numeric", B)
sig2eps     <- vector("numeric", B)
sig2phi     <- vector("numeric", B)

phi[1]             <- mean(yt)
mu[1]              <- 0
sig2eps[1]         <- var(yt)
sig2phi[1]         <- 1

for(t in 2:B){
```

```
  ### sample phi ###
  phi[t] <- rnorm(1, (prod/sig2eps[t-1]+mu[t-1]/sig2phi[t-1])*(
  ytsq/sig2eps[t-1]+1/sig2phi[t-1])^-1,(
  ytsq/sig2eps[t-1]+1/sig2phi[t-1])^-1)
  ### sample mu ###
  mu[t] <- rtnorm(1,phi[t-1],sig2phi[t-1],-1,1)
  ### sample sig2eps ###
  sig2eps[t] <- rinvgamma(1,n/2,(1/2)*sum((yt-phi[t-1]*ytmin1)^2))
  ### sample sig2phi ###
  sig2phi[t] <- rinvgamma(1,1/2,(1/2)*(phi[t-1]-mu[t-1])^2)
}

par(mfrow=c(2,2))

plot.ts(phi)
plot.ts(mu)
plot.ts(sig2eps)
plot.ts(sig2phi)

pacf(phi[10000:20000])
pacf(mu[10000:20000])
pacf(sig2eps[10000:20000])
pacf(sig2phi[10000:20000])

plot(density(phi[10000:20000]))
plot(density(mu[10000:20000]))
plot(density(sig2eps[10000:20000]))
plot(density(sig2phi[10000:20000]))

geweke.diag(phi)
geweke.diag(mu)
geweke.diag(sig2eps)
geweke.diag(sig2phi)

# Gelman diagnostics for phi

first.chain  <- rep(c(T,F,F,F), B/4)
second.chain <- rep(c(F,T,F,F), B/4)
third.chain  <- rep(c(F,F,T,F), B/4)
fourth.chain <- rep(c(F,F,F,T), B/4)

chain1 <- mcmc(phi[first.chain])
chain2  <- mcmc(phi[second.chain])
chain3  <- mcmc(phi[third.chain])
chain4  <- mcmc(phi[fourth.chain])
```

```
allChains <- mcmc.list(list(chain1, chain2, chain3, chain4))

gelman.diag(allChains)

# Gelman diagnostics for mu


chain1 <- mcmc(mu[first.chain])
chain2  <- mcmc(mu[second.chain])
chain3  <- mcmc(mu[third.chain])
chain4  <- mcmc(mu[fourth.chain])

allChains <- mcmc.list(list(chain1, chain2, chain3, chain4))

gelman.diag(allChains)


most.thin <- rep(c(rep(F, 19), T), 1000)

summary(most.thin)
length(phi)

pacf(phi[most.thin])
pacf(mu[most.thin])
pacf(sig2eps[most.thin])
pacf(sig2phi[most.thin])

tab <- t(cbind(quantile(phi[most.thin],     probs=c(.025, .5, .975)),
               quantile(mu[most.thin],      probs=c(.025, .5, .975)),
               quantile(sig2eps[most.thin], probs=c(.025, .5, .975)),
               quantile(sig2phi[most.thin], probs=c(.025, .5, .975))))

stargazer(tab)

model.frequentist.2 <- arma(brazil.deseasonalized.diff, order = c(1,0))


##############

# Brazil GDP -- nonstationary

##############

brazil <- read.table("./brazilgdp.txt")
```

```
plot.ts(brazil$V1)

yt       <- brazil$V1
ytmin1   <- yt[c(1:(length(yt)-1))]
yt       <- yt[c(2:length(yt))]
prod     <- yt%*%ytmin1
prod     <- prod[1]
ytsq     <- sum(ytmin1^2)
n        <- length(yt)

B        <- 10000
phi      <- vector("numeric", B)
mu        <- vector("numeric", B)
sig2eps    <- vector("numeric", B)
sig2phi    <- vector("numeric", B)

phi[1]           <- 1
mu[1]            <- 0
sig2eps[1]       <- 100
sig2phi[1]       <- 1

for(t in 2:B){
  ### sample phi ###
  phi[t] <- rnorm(1, (prod/sig2eps[t-1]+mu[t-1]/sig2phi[t-1])*(
  ytsq/sig2eps[t-1]+1/sig2phi[t-1])^-1,(
  ytsq/sig2eps[t-1]+1/sig2phi[t-1])^-1)
  ### sample mu ###
  mu[t] <- rtnorm(1,phi[t-1],sig2phi[t-1], -100, 100)
  ### sample sig2eps ###
  sig2eps[t] <- rinvgamma(1,n/2,(1/2)*sum((
  yt-phi[t-1]*ytmin1)^2))
  ### sample sig2phi ###
  sig2phi[t] <- rinvgamma(1,1/2,(1/2)*(phi[t-1]-mu[t-1])^2)
}

par(mfrow=c(2,2))

plot.ts(phi)
plot.ts(mu)
plot.ts(sig2eps)
plot.ts(sig2phi)

pacf(phi[B/2:B])
pacf(mu[B/2:B])
```

```
pacf(sig2eps[B/2:B])
pacf(sig2phi[B/2:B])

plot(density(phi[B/2:B]))
plot(density(mu[B/2:B]))
plot(density(sig2eps[B/2:B]))
plot(density(sig2phi[B/2:B]))

geweke.diag(phi)
geweke.diag(mu)
geweke.diag(sig2eps)
geweke.diag(sig2phi)

# Gelman diagnostics for phi

first.chain  <- rep(c(T,F,F,F), B/4)
second.chain <- rep(c(F,T,F,F), B/4)
third.chain  <- rep(c(F,F,T,F), B/4)
fourth.chain <- rep(c(F,F,F,T), B/4)

chain1 <- mcmc(phi[first.chain])
chain2  <- mcmc(phi[second.chain])
chain3  <- mcmc(phi[third.chain])
chain4  <- mcmc(phi[fourth.chain])

allChains <- mcmc.list(list(chain1, chain2, chain3, chain4))

gelman.diag(allChains)


# Gelman diagnostics for mu

chain1 <- mcmc(mu[first.chain])
chain2  <- mcmc(mu[second.chain])
chain3  <- mcmc(mu[third.chain])
chain4  <- mcmc(mu[fourth.chain])

allChains <- mcmc.list(list(chain1, chain2, chain3, chain4))

gelman.diag(allChains)

most.thin <- rep(c(rep(F, 19), T), B/20)

summary(most.thin)
length(phi)
```

```
pacf(phi[most.thin])
pacf(mu[most.thin])
pacf(sig2eps[most.thin])
pacf(sig2phi[most.thin])

tab <- t(cbind(quantile(phi[most.thin],     probs=c(.025, .5, .975)),
               quantile(mu[most.thin],      probs=c(.025, .5, .975)),
               quantile(sig2eps[most.thin], probs=c(.025, .5, .975)),
               quantile(sig2phi[most.thin], probs=c(.025, .5, .975))))

stargazer(tab)


model.frequentist.2 <- arma(brazil$V1, order = c(1,0))

summary(model.frequentist.2)
```