ARIMA process

Tadesse Zemicheal

February 2, 2016

# Simulate AR(1) process with different parameter value and length size.

First, I simulated AR(1) with length of 30 and for 500 simulation. Then I tried to fit AR model with order from 0 to 6. Summary of my result for different length and parameter value is given below.

|  |  |  |
| --- | --- | --- |
| Timeseries length | Parameter value | # of times AR(1) fits better |
| 30 | 0.7 | 313 |
| 100 | 0.7 | 368 |
| 30 | 0.3 | 342 |
| 100 | 0.3 | 365 |
| 30 | -0.3 | 218 |
| 100 | -0.3 | 355 |

The above simulation result shows above half of the simulation out of 500 fits the true AR(1) model. In the other hand, increasing the length of the timeseries makes the model to fit better to its true value. However, changing parameter value doesn't have much effect on the number of times the model fits to its true value, however a small negative valeu of has reduced the number of times the model fits to its true value .

# Question 2 find and fit an ARIMA model to the deere2 dataset

## Appendix

knitr::opts\_chunk$set(echo=FALSE, message = FALSE,   
 warning = FALSE, results = "hide", fig.height = 3, fig.width = 6)   
#----------------------------  
# Simulate AR(1) process with random parameter value of different length  
#------------------------------  
library('dplyr')  
  
#Simulate AR(1) with len=length and alpha1 parameter  
# returns generated timeseries data  
sim.dist<-function(len,alpha1)  
{  
 ar1<-arima.sim(model=list(ar=alpha1,ma=0,sd=1),len)  
 return(ar1)  
}  
  
# Fit AR of or order to X  
# return aic of the fitted model   
fit.model <-function(x,or)  
{  
 #pass argument and return aic  
 fit<-arima(x,c(or,0,0))  
 return(fit$aic)  
}  
  
#Simulate and fit length with paramter value   
# Return number of times the data fits the true value  
simulate.and.fit<-function(len,alpha1)  
{  
  
sim.arima<-failwith(NA,sim.dist)  
#simulate 500 runs   
ar1.sim <- replicate(500,sim.arima(len,alpha1))  
  
#fit model starting from 0 to 6 order of 500 simulation  
aic.model <-matrix(NA,nrow=7,ncol=500)  
fit.fail <-failwith(NA,fit.model)  
for(i in 0:6)  
{  
 aic.model[i+1,]<-apply(ar1.sim,2,function(x){fit.fail(x,i)})  
}  
  
### check how often the model with lowest AIC is the true model   
  
ar1.true.fit <- sum(apply(aic.model,2,which.min)==2) #count number of AR(1) bits other model  
return(ar1.true.fit)  
}  
  
#Simulate the AR(1) model and fit different AR models  
  
alpha1=0.7 #alpha1 parameter  
len=30 #length  
best.fit<- simulate.and.fit(len,alpha1)  
cat(" Out of 500 simulation generated by AR(1), AR(1) fits as best ",best.fit," times")  
  
#### Repeat with longer length ###  
len2=100  
best.fit2<- simulate.and.fit(len2,alpha1)  
  
## Repeat with different parameter (alpha1)  
alph2 = 0.3  
best.fi3<- simulate.and.fit(len,alpha1) # short timeseries  
best.fit4<- simulate.and.fit(len2,alpha1) # for longer timeseries  
#Negative alpha value   
alpha3 = -0.3  
best.fit5<- simulate.and.fit(len,alpha3) # short timeseries  
best.fit6<- simulate.and.fit(len2,alpha3) # for longer timeseries  
########## Question 2 #################################3  
#  
# Find ARIMA model to deere2 dataset  
########################################################  
library(ggplot2)  
library(dplyr)  
library(TSA)  
  
  
data("deere2")  
  
big\_font <- theme\_grey(base\_size = 24)  
source(url("http://stat565.cwick.co.nz/code/fortify-ts.r"))  
deere <- fortify(deere2)  
deere <- rename(deere, deviation = x)  
  
qplot(time, deviation, data = deere, geom = "line") +  
 big\_font  
  
# differencing can remove trends  
  
deere$diff1 <- c(NA, diff(deere$deviation))  
qplot(time, diff1, data = deere, geom = "line") +  
 big\_font  
  
  
deere$diff2 <- c(NA, diff(deere$diff1))  
qplot(time, diff2, data = deere, geom = "line") +  
 big\_font  
# The trend looks removed but still there is high variance,  
#log transform with small   
  
min.dev <- min(deere$deviation)  
deere$diff1\_log <- c(NA,diff(log(deere$deviation +1-min.dev)))  
qplot(time, diff1\_log, data = deere, geom = "line") +  
 big\_font  
#Variance removed with few outlies at the beginning   
  
qplot(time, diff1\_log^2, data = deere) +  
 geom\_smooth() +  
 big\_font  
# a few outliers in the beginning   
  
# looks stationary, let's choose an ARMA(p, q) model  
acf(deere$diff1\_log, lag.max = 50, na.action = na.pass)  
# Looks MA process with significant MA(1)  
# significant at lag 1,3 and 4 and may be 2  
pacf(deere$diff1\_log, lag.max = 50, na.action = na.pass)  
  
#significant at lag 1, little bit at 2, 3,4  
# models to try MA(1), AR(1), ARMA(1, 1), MA(2)  
#lets do grid search till ARMA(5,5)  
n <- nrow(deere)  
  
#Grid search for good fit  
min.aic <- 9999  
p<- -4   
q<- -4  
  
for(i in 0:3)  
{  
 for(j in 0:3)  
 {  
 fit<-arima(log(deere$deviation-min.dev+1), order = c(i, 1, j), xreg = 1:n)  
 if(fit$aic<min.aic)  
 {  
 min.aic <- fit$aic  
 p<- i  
 q<- j  
 }  
 }  
}  
# ARMA(1,3) seems best, check residuals (a.k.a innovations)  
fit\_arma <- arima(log(deere$deviation-min.dev+1), order = c(p, 1, q), xreg = 1:n)  
  
# diagnostics  
# is there any correlation left in the residuals  
acf(residuals(fit\_arma),na.action=na.pass)  
pacf(residuals(fit\_arma),na.action=na.pass)  
# looks good  
  
# check normality  
qqnorm(residuals(fit\_arma))  
qqline(residuals(fit\_arma))  
# Looks normal with few expected outliers   
  
# a time plot of residuals  
deere$residuals <- residuals(fit\_arma)  
qplot(time, residuals, data = deere, geom = "line")  
  
# outliers  
num\_out<-subset(deere, abs(residuals) > 0.3) %>% nrow  
#We have many outliers around the middle  
library(ggplot2)  
library(dplyr)  
library(TSA)  
data(robot)  
  
big\_font <- theme\_grey(base\_size = 24)  
source(url("http://stat565.cwick.co.nz/code/fortify-ts.r"))  
robot <- fortify(robot)  
robot <- rename(robot, distance = x)  
# Timeseries of distance travelled in time  
qplot(time, distance, data = robot, geom = "line") +  
 big\_font  
  
#difference   
robot$diff1<-c(NA,diff(robot$distance))  
qplot(time, diff1, data = robot, geom = "line") +  
 big\_font  
  
#difference   
robot$diff1<-c(NA,diff(robot$distance))  
qplot(time, diff1, data = robot, geom = "line") +  
 big\_font  
  
#different 2  
robot$diff2<- c(rep(NA,1),diff(robot$diff1,lag=1))  
qplot(time, diff2, data = robot, geom = "line") +  
 big\_font  
  
# Looks good with little bit variance.  
qplot(time, diff2^2, data = robot) +  
 geom\_smooth() +  
 big\_font  
# Variance of the   
qplot(time %/%1, diff2^2, data = robot,group=time%/%1,geom="boxplot") +  
 geom\_smooth()+  
 big\_font  
# looks stationary, let's choose an ARMA(p, q) model  
acf(robot$diff2, lag.max = 50, na.action = na.pass)  
# Looks AR(1) with signficant lag at 1   
# significant at lag 1, 2,3 and 5  
pacf(robot$diff2, lag.max = 50, na.action = na.pass)  
  
  
# Check the   
n <- nrow(robot)  
#Grid search for good fit  
min.aic <- 9999  
p<- -4   
q<- -4  
for(i in 0:4)  
{  
 for(j in 0:4)  
 {  
 fit<-arima(robot$distance, order = c(i, 1, j), xreg = 1:n)  
 if(fit$aic<min.aic)  
 {  
 min.aic <- fit$aic  
 p<- i  
 q<- j  
 }  
 }  
}  
# ARMA(1,2) seems best  
fit\_arima <- arima(robot$distance, order = c(p, 1, q), xreg = 1:n)  
  
# diagnostics  
# is there any correlation left in the residuals  
acf(residuals(fit\_arima),na.action = na.pass)  
pacf(residuals(fit\_arima),na.action = na.pass)  
# looks good  
  
# check normality  
qqnorm(residuals(fit\_arima))  
qqline(residuals(fit\_arima))  
# Looks normal with few expected outliers   
  
# a time plot of residuals  
robot$residuals <- residuals(fit\_arima)  
qplot(time, residuals, data = robot, geom = "line")  
  
# outliers  
subset(robot, abs(residuals) > 0.3)  
# We don't have outliers forthis dataset