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 $\alpha_1 = 0.7$ 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 α_1 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)
```

```r
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

# Trend is gone but but there are outliers in the middle
#log transformation is not possible becuase of negative values.
qplot(time, diff2^2, data = deere) +
    geom_smooth() +
    big_font
# a few outliers in the middle

# looks stationary, let's choose an ARMA(p, q) model
acf(deere$diff2, lag.max = 50, na.action = na.pass)
# significant at lag 1,3 and 4 and may be 2
pacf(deere$diff2, lag.max = 50, na.action = na.pass)

#significant at lag 1, little bit at 2, 3,4
# models to try MA(1), AR(2), ARMA(1, 1), MA(2),ARMA(1,1),ARMA(2,2)
#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 0:4)
{
    for(j in 0:4)
    {
        fit<-arima(deere$diff2, order = c(i, 1, j), xreg = 1:n)
        if(fit$aic<min.aic)
        {
            min.aic <- fit$aic
            p<- i
            q<- j
        }
```

```r
    }
}

# ARMA(2,4) seems best, check residuals (a.k.a innovations)
fit_arma24 <- arima(deere$diff2, order = c(2, 1, 4), xreg = 1:n)

# diagnostics
# is there any correlation left in the residuals
acf(residuals(fit_arma24))
pacf(residuals(fit_arma24))
# looks good

# check normality
qqnorm(residuals(fit_arma24))
qqline(residuals(fit_arma24))
# Looks normal with few expected outliers

# a time plot of residuals
deere$residuals <- residuals(fit_arma24)
qplot(time, residuals, data = deere, geom = "line")

# outliers
subset(deere, abs(residuals) > 0.3)
#We have many outliers around the middle

#--------------------------------------------
# Q2(b)
# Fit and find ARIMA model for robot dataset
#--------------------------------------------
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$diff2 <- c(NA,diff(robot$diff1))
qplot(time, diff2, data = robot, geom = "line") +
    big_font
```

```r
#different 2
robot$diff3<-c(NA,diff(robot$diff2))
qplot(time, diff3, data = robot, geom = "line") + geom_smooth() +
    big_font
# Looks good with little bit variance.
# Looks some variance and outliers at the middle
qplot(time, diff3^2, data = robot) +
    geom_smooth() +
    big_font

# Variance of the
qplot(time %/%2, diff3^2, data = robot) +
    geom_smooth()+
    big_font


# looks stationary, let's choose an ARMA(p, q) model
acf(robot$diff3, lag.max = 50, na.action = na.pass)
# Looks AR(1) with signficant lag at 1
# significant at lag 1, 2,3,4, and 5
pacf(robot$diff3, 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$diff2, order = c(i, 1, j), xreg = 1:n)
        if(fit$aic<min.aic)
        {
            min.aic <- fit$aic
            p<- i
            q<- j
        }
    }
}
# ARMA(4,3) seems best, check residuals (a.k.a innovations)
fit_arima <- arima(robot$diff2, order = c(0, 1, 4), 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))
```

```r
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 have many outliers around the middle
```