Package ‘gsarima’

February 20, 2015

Version  0.1-4
Date  2013-06-17
Title  Two functions for Generalized SARIMA time series simulation
Author  Olivier Briet <o.briet@gmail.com>
Maintainer  Olivier Briet <o.briet@gmail.com>
Depends  R (>= 2.4.0)
Imports  MASS
Suggests  gamlss.util (>= 4.2-0)
Description  Write SARIMA models in (finite) AR representation and simulate
generalized multiplicative seasonal autoregressive moving average (time) series
with Normal / Gaussian, Poisson or negative binomial distribution.
License  GPL (>= 2)
URL  http://www.r-project.org
NeedsCompilation  no
Repository  CRAN
Date/Publication  2014-11-25 17:27:34

R topics documented:

      gsarima-package .................................................. 2
      arrep ............................................................ 3
      garsim ........................................................... 4

Index  9
gsarima-package

Two functions for Generalized SARIMA time series simulation

Description

Write SARIMA models in (finite) AR representation and simulate generalized multiplicative seasonal autoregressive moving average (time) series

Details

Package: gsarima
Type: Package
Version: 0.1-4
Date: 2013-06-17
License: GPL (>= 2)
LazyLoad: yes

Use arrep() for converting the SARIMA function into AR representation, and use garsim() to simulate.

Author(s)

Olivier Briet <o.briet@gmail.com>
Maintainer: Olivier Briet <o.briet@gmail.com>

References


Compute the Autoregressive Representation of a Sarima Model

Description
Invert (invertible) SARIMA(p, d, q, P, D, Q) models to ar representation.

Usage
arrep(notation = "arima", phi = c(rep(0, 10)), d = 0, theta = c(rep(0, 10)), Phi = c(rep(0, 10)), D = 0, Theta = c(rep(0, 10)), frequency = 1)

Arguments
- notation: "arima" for notation of the type used by the function arima(stats), "dse1" for type notation used by the package dse1.
- phi: p vector of autoregressive coefficient.
- d: difference operator, implemented: d element of (0,1,2).
- theta: q vector of moving average coefficients.
- Phi: P vector of seasonal autoregressive coefficients.
- D: Seasonal difference operator, implemented: D element of (0,1,2).
- Theta: Q vector of seasonal moving average coefficients.
- frequency: The frequency of the seasonality (e.g. frequency = 12 for monthly series with annual periodicity).

Details
For input, positive values of phi, theta, Phi and Theta indicate positive dependence. Implemented for p,q,P,Q element of c(0,1,2,3,4,5,6,7,8,9,10). The ar representation is truncated at coefficients less than 1.0e-10. Values of theta, Theta near non invertibility (-1 or 1) will not be practical and will cause near infinite lags, especially for Theta and large frequency.

Value
A vector containing a truncated autoregressive representation of a SARIMA model. This can be used as input for the function gar.sim.

Author(s)
Olivier Briet < o.briet@gmail.com >

References
garsim

Simulate a Generalized Autoregressive Time Series

Description

Simulate a time series using a general autoregressive model.

Usage

garsim(n, phi, X = matrix(0, nrow = n), beta = as.matrix(0), sd = 1, family = "gaussian", transform.X = "identity", link = "identity", minimum = 0, zero.correction = "zqf", c = 1, theta = 0)

Arguments

- **n**
  - The number of simulated values.
- **phi**
  - A vector of autoregressive parameters of length p.
- **X**
  - An n by m optional matrix of external covariates, optionally including an intercept (recommended for family = "poisson").
- **beta**
  - An m vector of coefficients.
### sd
Standard deviation for Gaussian family.

### family
Distribution family, defaults to "gaussian".

### transform.Xbeta
Optional transformation for the product of covariates and coefficients, see Details.

### link
The link function, defaults to "identity".

### minimum
A minimum value for the mean parameter of the Poisson and Negative Binomial distributions (only applicable for link= "identity" and family = c("poisson","negative.binomial")). Defaults to 0. A small positive value will allow non-stationary series to "grow" after encountering a simulated value of 0.

### zero.correction
Method for transformation for dealing with zero values (only applicable when link = "log"), see Details.

### c
The constant used for transformation before taking the logarithm (only applicable when link = "log"). A value between 0 and 1 is recommended.

### theta
Parameter theta (for family = "negative.binomial").

### Details
Implemented are the following models:

1. family = "gaussian", link = "identity"
2. family = "poisson", link = "identity"
3. family = "poisson", link = "identity", transform.Xbeta = "exponential"
4. family = "poisson", link = "log", zero.correction = "zq1"
5. family = "poisson", link = "log", zero.correction = "zq2"
6. family = "negative.binomial", link = "identity", transform.Xbeta = "exponential"
7. family = "negative.binomial", link = "identity", transform.Xbeta = "exponential"
8. family = "negative.binomial", link = "log", zero.correction = "zq1"
9. family = "negative.binomial", link = "log", zero.correction = "zq2"

Models 1 to 4 are within the family of GARMA models of Benjamin and colleagues 2003. Model 2 is the extension to higher order p of a Poisson CLAR(1) model proposed by Grunwald and colleagues (2000). Model 3 is a modification of the PAR(p) data generating process (http://www.utdallas.edu/~pxb054000/code/pests.r) of Brandt and Williams (2001). Note that for psi = 0, the model reduces to a standard Poisson model with log-link function. For a model without external variables (only an intercept), the transformation of Xbeta has no consequence and then model 3 is the same as model 2. Model 4 corresponds to model 2.2 of Zeger and Qaqish (1988). The value c is only added to values of zero prior to taking the log. Models 6 to 9 are similar but with negative binomial distribution.

### Value
An autoregressive series of length n. Note that the first p data do not have autoregressive structure.

### Note
Version 0.1-2: bug corrected and garmaFit example given.

### Author(s)
Olivier Briet < o.briet@gmail.com >
References


See Also

'negbin(MASS)', 'arrep'.

Examples

```r
N <- 1000
ar <- c(0.8)
intercept <- -2
frequency <- 1
x <- rnorm(N)
beta.x <- 0.7
# Gaussian simulation with covariate
X <- matrix(c(rep(intercept, N+length(ar)), rep(0, length(ar)), x), ncol=2)
y.sim <- garsim(n=(N+length(ar)), phi=ar, X=X, beta=c(1, beta.x), sd=sqrt(1))
y <- y.sim[1:length(ar):((N+length(ar))]
tsy <- ts(y, freq=frequency)
plot(tsy)
arima(tsy, order=c(1,0,0), xreg=x)

# Gaussian simulation with covariate and deterministic seasonality through first order harmonic
ar <- c(1.4,-0.4)
frequency <- 12
beta.x <- c(0.7, 4, 4)
x <- matrix(nrow = (N+length(ar)), ncol=3)
for (t in 1:length(ar)){
X[t,1] <- 0
X[t,2] <- sin(2*pi*(t- length(ar))/frequency)
X[t,3] <- cos(2*pi*(t- length(ar))/frequency)
}
for (t in (1+length(ar):((N+length(ar))){
X[t,1] <- X[t- length(ar)]
X[t,2] <- sin(2*pi*(t- length(ar))/frequency)
X[t,3] <- cos(2*pi*(t- length(ar))/frequency)
}
y.sim <- garsim(n=(N+length(ar)), phi=ar, X=X, beta= beta.x, sd=sqrt(1))
```
y <- y.sim[(1 + length(ar)):(N + length(ar))]
tsy <- ts(y, freq=frequency)
plot(tsy)
Xreg <- matrix(nrow=N, ncol=3)
for (t in 1:N)
  Xreg[t,1] <- x[t]
Xreg[t,2] <- sin(2*pi*t/frequency)
Xreg[t,3] <- cos(2*pi*t/frequency)
}
arimares <- arima(tsy, order=c(1,1,0), xreg=Xreg)
tsdig(arimares)
arimares

# Negative binomial simulation with covariate
ar <- c(0.8)
frequency <- 1
beta.x <- 0.7
X <- matrix(c(rep(log(intercept), N + length(ar)), rep(0, length(ar)), x), ncol=2)
y.sim <- garsim(n=N + length(ar)), phi=ar, beta=c(1, beta.x), link= "log",
family = "negative.binomial", zero.correction = "zq1", c=1, theta=5, X=X)
y <- y.sim[(1 + length(ar)):(N + length(ar))]
tsy <- ts(y, freq=frequency)
plot(tsy)
library(gamlss.util)
m10 <- garmaFit(y ~ x, family=NB1, alpha=1)
sm10 <- summary(m10)
sm10
1/sm10$sigma # compare with theta specified for negative binomial

# Poisson ARMA(1,1) with identity link and negative auto correlation
N <- 500
phi <- c(-0.8)
theta <- c(0.6)
ar <- arrep(phi=phi, theta=theta)
check <- acf2AR(ARMAacf(ar=phi, ma=theta, lag.max = 100, pacf = FALSE))[100,1:length(ar)]
as.data.frame(cbind(ar, check))
intercept <- 100
frequency <- 1
X <- matrix(c(rep(intercept, N + length(ar))), ncol=1)
y.sim <- garsim(n=N + length(ar)), phi=ar, beta=c(1), link= "identity",
family = "poisson", minimum = -100, X=X)
y <- y.sim[(1 + length(ar)):(N + length(ar))]
tsy <- ts(y, freq=frequency)
plot(tsy)

# Poisson AR(1) with identity link and negative auto correlation
N <- 1000
ar <- c(-0.8)
intercept <- 100
frequency <- 1
X <- matrix(c(rep(intercept, N + length(ar))), ncol=1)
y.sim <- garsim(n=N + length(ar)), phi=ar, beta=c(1), link= "identity",
family = "poisson", minimum = -100, X=X) y<-y.sim[(1+length(ar)):(N+length(ar))] tsy<-ts(y, freq=frequency) plot(tsy)

# Example of negative binomial GSARIMA(2,1,0,0,0,1)x phi<-c(0.5,0.2) theta<-c(0) Theta<-c(0.5) Phi<-c(0) d<-c(1) D<-c(0) frequency<-12 ar<-arrep(phi=phi, theta=theta, Phi=Phi, Theta=Theta, frequency= frequency, d=d, D=D) N<-c(1000) intercept<-c(10) x<- rnorm(N) beta.x<-c(0.7) X<-matrix(c(rep(log(intercept), N+length(ar)), rep(0, length(ar)), x), ncol=2) c<-c(1) y.sim <- garsim(n=(N+length(ar)), phi=ar, beta=c(1,beta.x), link= "log", family = "negative.binomial", zero.correction = "zq1", c=c, theta=5, X=X) y<-y.sim[(1+length(ar)):(N+length(ar))] tsy<-ts(y, freq=frequency) plot(tsy) plot(log(tsy))
Index

∗Topic package
   gsarima-package, 2

∗Topic ts
   arrep, 3
   garsim, 4

arrep, 3
garsim, 4
gsarima (gsarima-package), 2
gsarima-package, 2