Package 'FitARMA'

January 4, 2019

Title Fit ARMA or ARIMA Using Fast MLE Algorithm

Version 1.6.1

Date 2013-09-26

Author A.I. McLeod

Maintainer A.I. McLeod <aimcleod@uwo.ca>

Depends R (>= 2.1.0), FitAR

Description Implements fast maximum likelihood algorithm for fitting ARMA time series. Uses S3 methods print, summary, fitted, residuals. Fast exact Gaussian ARMA simulation.

Classification/ACM G.3, G.4, I.5.1

Classification/MSC 62M10, 91B84

License GPL (>= 2)

URL http://fisher.stats.uwo.ca/faculty/aim

NeedsCompilation yes

Repository CRAN

Date/Publication 2019-01-04 16:33:55 UTC

R topics documented:

FitARMA-package *FitARMA: Fit ARMA or ARIMA using fast MLE algorithm*

Description

Fit ARMA/ARIMA time series model using fast algorithm. All MLE computations in R. Two estimation functions: 'FitARMA' and 'GetFitARMA' are provided. 'FitARMA' provides more options including an option for exact MLE estimation of the intercept term. 'GetFitARMA' is provided for bootstrapping and simulation experiments. S3 Methods 'print', 'summary', 'coef', 'residuals', 'fitted' provided. Fast exact Gaussian ARMA simultation using C.

Details

Package:	FitARMA
Type:	Package
Version:	1.4
Date:	2010-12-01
License:	GLP 2.0 or greater
LazyLoad:	yes

FitARMA is the main function.

Author(s)

A.I. McLeod

References

A. I. McLeod, Ying Zhang (2007). Faster ARMA maximum likelihood estimation, Computational Statistics & Data Analysis 52(4), URL http://dx.doi.org/10.1016/j.csda.2007.07.020

See Also

arima, AcfPlot

Examples

```
data(SeriesA)
#ARIMA(0,1,1) with exact estimation of mean of differenced series
ans<-FitARMA(SeriesA, order=c(0,1,1), MeanMLEQ=TRUE)
ans
coef(ans)
#ARIMA(0,1,1) with sample-mean estimation of mean of differenced series
ans<-FitARMA(SeriesA, order=c(0,1,1))
ans
coef(ans)
```

```
#ARIMA(0,1,1) with mean of differenced series set to zero
#as in 'arima'
ans<-FitARMA(SeriesA, order=c(0,1,1), demean=FALSE)
ans
coef(ans)
# illustrating methods
summary(ans)
resid(ans)
fitted(ans)
ans$racf
#Simulate and fit Gaussian ARMA
z<-SimulateGaussianARMA(0.9, 0.5, 200)
#GetFitARMA is faster than FitARMA.
#Use GetFitARMA for parametric bootstrap and simulation experiments
GetFitARMA(z, p=1, q=1)
```

`coef.FitARMA` *coef method for class FitARMA*

Description

produces table showing parameters, standard errors and Z-ratios

Usage

```
## S3 method for class 'FitARMA'
coef(object, ...)
```

Arguments

`object`	class FitARMA object
`...`	auxiliary parameters

Value

matrix with 3 columns

Author(s)

A.I. McLeod

See Also

`FitARMA`

Examples

```
data(SeriesA)
out<-FitARMA(SeriesA, c(1,0,1))
coef(out)
```

FitARMA *Fit ARMA/ARIMA using fast MLE algorithm*

Description

Fits an ARIMA(p,d,q) model using the algorithm given in McLeod and Zhang (2007).

Usage

```
FitARMA(z, order = c(0, 0, 0), demean = TRUE, MeanMLEQ = FALSE, pApprox = 30, MaxLag = 30)
```

Arguments

`z`	time series
`order`	model order, c(p,d,q)
`demean`	if TRUE, mean parameter included otherwise assumed zero
`MeanMLEQ`	exact MLE for mean, ignored unless demean=TRUE
`pApprox`	order of approximation to be used
`MaxLag`	maximum number of lags for portmanteau test

Details

See McLeod and Ying (2007).

Value

A list with class name "FitARMA" and components:

`loglikelihood`	value of the loglikelihood
`phiHat`	AR coefficients
`thetaHat`	MA coefficients
`sigsqHat`	innovation variance estimate
`muHat`	estimate of the mean
`covHat`	covariance matrix of the coefficient estimates
`racf`	residual autocorrelations
`LjungBox`	table of Ljung-Box portmanteau test statistics
`res`	innovation residuals, same length as z
`fits`	fitted values, same length as z

`demean`	TRUE if mean estimated otherwise assumed zero
`IterationCount`	number of iterations in mean mle estimation
`convergence`	value returned by optim – should be 0
`MLEMeanQ`	TRUE if mle for mean algorithm used
`tsp`	tsp(z)
`call`	result from match.call() showing how the function was called
`ModelTitle`	description of model
`DataTitle`	returns attr(z,"title")

Note

When d>0 and demean=TRUE, the mean of the differenced series is estimated. This corresponds to including a polynomial of degree d.

When d>0, the AIC/BIC are computed for the differenced series and so they are not comparable to the values obtained for models with d=0.

Author(s)

A.I. McLeod, aimcleod@uwo.ca

References

A.I. McLeod andY. Zhang (2008), Faster ARMA maximum likelihood estimation, Computational Statistics & Data Analysis, 52-4, 2166-2176. DOI link: http://dx.doi.org/10.1016/j.csda.2007.07.020

See Also

`GetFitARMA`, `print.FitARMA`, `coef.FitARMA`, `residuals.FitARMA`, `fitted.FitARMA`, `arima`

Examples

```
data(SeriesA) #in datasets()
out1<-FitARMA(SeriesA, c(1,0,1))
out1
coef(out1)
out2<-FitARMA(SeriesA, c(0,1,1))
out2
coef(out2)
```

`fitted.FitARMA` *fitted method for class FitARMA*

Description

The fitted values are the observed minus residuals. If there is differencing, the observed values are those corresponding to the differenced time series.

Usage

```
## S3 method for class 'FitARMA'
fitted(object, ...)
```

Arguments

`object`	class FitARMA object
`...`	auxiliary parameters

Value

vector or ts object

Author(s)

A.I. McLeod

See Also

FitARMA

Examples

```
data(SeriesA)
out<-FitARMA(SeriesA, c(1,0,1))
fitted(out)
```

GetFitARMA *Fit ARMA(p,q) model to mean zero time series.*

Description

The algorithm of McLeod and Zhang (2007) is used.

Usage

```
GetFitARMA(y, p, q, pApprox = 30, init = 0)
```

Arguments

y	time series
p	AR order
q	MA order
pApprox	AR approximation
init	initial parameter estimates

Details

See McLeod and Zhang (2006).

Value

loglikelihood	value of maximized loglikelihood
phiHat	estimated phi parameters
thetaHat	estimated theta parameters
convergence	result from optim
algorithm	indicates "L-BFGS-B" or "Nelder-Mead" according as which algorithm was used in optim

Author(s)

A.I. McLeod, aimcleod@uwo.ca

References

A.I. McLeod andY. Zhang (2008), Faster ARMA maximum likelihood estimation, Computational Statistics & Data Analysis, 52-4, 2166-2176. DOI link: http://dx.doi.org/10.1016/j.csda.2007.07.020

See Also

arima, FitARMA

Examples

```
data(SeriesA)
z<-SeriesA-mean(SeriesA)
GetFitARMA(z, 1, 1)
w<-diff(z, differences=1)
GetFitARMA(w, 0, 1)
```

ImpulseCoefficientsARMA *Impulse coefficients of ARMA*

Description

The coefficients in the infinite MA expansion of the ARMA model are determined.

Usage

```
ImpulseCoefficientsARMA(phi, theta, lag.max)
```

Arguments

phi	AR coefficients
theta	MA coefficients
lag.max	lags 0,...,lag.max

Value

vector length lag.max+1

Author(s)

A.I. McLeod

Examples

```
ImpulseCoefficientsARMA(0.9,0.5,20)
```

InformationMatrixARMA *Expected large-sample information matrix for ARMA*

Description

The expected large-sample information matrix per observation for ARMA(p,q) models is computed.

Usage

```
InformationMatrixARMA(phi = numeric(0), theta = numeric(0))
```

Arguments

phi	AR coefficients
theta	MA coefficients

Details

The information matrix is derived by Box and Jenkins (1970).

Value

a matrix of order (p+q)

Author(s)

A.I. McLeod

References

Box and Jenkins (1970). Time Series Analysis: Forecasting and Control.

See Also

FitARMA

Examples

```
#The covariance matrix estimates of the parameters phi and theta in an ARMA(1,1)
#with phi=0.9 and theta=0.5 and n=200 is
v<-solve(InformationMatrixARMA(0.9,0.5))/200
v
#and the standard errors are
sqrt(diag(v))
```

print.FitARMA *print method for class FitARMA*

Description

a brief summary is printed out of the fitted model

Usage

```
## S3 method for class 'FitARMA'
print(x, ...)
```

Arguments

x	object, class FitARMA
...	optional arguments

Value

the result is displayed

Author(s)

A.I. McLeod

See Also

FitARMA

Examples

```
data(SeriesA)
FitARMA(SeriesA, c(1,0,1))
```

residuals.FitARMA *residuals method for class FitARMA*

Description

The innovation residuals are obtained.

Usage

```
## S3 method for class 'FitARMA'
residuals(object, ...)
```

Arguments

`object`	class FitARMA object
`...`	auxiliary parameters

Value

vector or ts object

Author(s)

A.I. McLeod

See Also

`FitARMA`

Examples

```
data(SeriesA)
out<-FitARMA(SeriesA, c(1,0,1))
resid(out)
```

`SimulateGaussianARMA` *Simulate Gaussian ARMA model*

Description

An exact simulation method is used to simulate Gaussian ARMA models.

Usage

```
SimulateGaussianARMA(phi, theta, n, InnovationVariance = 1, UseC = TRUE)
```

Arguments

`phi`	AR coefficients
`theta`	MA coefficients
`n`	length of series
`InnovationVariance`	innovation variable, default is 1
`UseC`	if `UseC=TRUE`, use C code. Otherwise, use slower R code.

Details

The detailed description is given in Hipel and McLeod (1994, 2006).

Value

a vector containing the time series

Author(s)

A.I. McLeod

References

Hipel, K.W. and McLeod, A.I. (2006). Time Series Modelling of Water Resources and Environmental Systems.

See Also

`arima.sim`

Examples

```
z<-SimulateGaussianARMA(0.9, 0.5, 200)
FitARMA(z, c(1,0,1))
```

---

summary.FitARMA *print method for class FitARMA*

Description

a summary is printed out of the fitted model

Usage

```
## S3 method for class 'FitARMA'
summary(object, ...)
```

Arguments

`object`	object, class FitARMA
`...`	optional arguments

Value

the result is displayed

Author(s)

A.I. McLeod

See Also

FitARMA

Examples

```
data(SeriesA)
out<-FitARMA(SeriesA, c(1,0,1))
summary(out)
```

TacvfARMA — *Theoretical Autocovariance Function of ARMA*

Description

The theoretical autocovariance function of an ARMA(p,q) with unit variance is computed. This algorithm has many applications. In this package it is used for the computation of the information matrix, in simulating p initial starting values for AR simulations and in the computation of the exact mle for the mean.

Usage

```
TacvfARMA(phi = numeric(0), theta = numeric(0), lag.max = 20)
```

Arguments

`phi`	AR coefficients
`theta`	MA coefficients
`lag.max`	computes autocovariances lags 0,1,...,lag.max

Details

The algorithm given by McLeod (1975) is used.

The built-in R function ARMAacf could also be used but it is quite complicated and apart from the source code, the precise algorithm used is not described. The only reference given for ARMAacf is the Brockwell and Davis (1991) but this text does not give any detailed exact algorithm for the general case.

Another advantage of TacvfARMA over ARMAacf is that it will be easier for to translate and implement this algorithm in other computing environments such as MatLab etc.

Value

vector of length lag.max+1 containing the autocovariances is returned

Author(s)

A.I. McLeod

References

McLeod, A.I. (1975), Derivation of the theoretical autocorrelation function of autoregressive moving-average time series, Applied Statistics 24, 255-256.

See Also

ARMAacf, InformationMatrixARMA

Examples

```
#calculate and plot the autocorrelations from an ARMA(1,1) model
# with parameters phi=0.9 and theta=0.5
 g<-TacvfARMA(0.9,0.5,20)
 AcfPlot(g/g[1], LagZeroQ=FALSE)
```

---

tccfAR *Theoretical cross-covariances of auxilary AR process in ARMA(p,q)*

Description

The auxilary AR processes in the ARMA(p,q) model phi(B)z(t)=theta(B)a(t) are defined by phi(B)u(t)=-a(t) and theta(B)v(t)=a(t). The upper off-diagonal p-by-q block of the ARMA information matrix is obtained from the cross-covariances of u(t) and v(t). This function obtains these covariances.

Usage

```
tccfAR(phi, theta)
```

Arguments

`phi`	AR coefficients in ARMA
`theta`	MA coefficients in ARMA

Details

A set of linear equations which determine the covariances is solved. The algorithm is similar in spirit to that for the autocovariances (McLeod, 1975).

Value

vector of cross-covariances

Author(s)

A.I. McLeod

References

McLeod, A.I. (1975), Derivation of the theoretical autocorrelation function of autoregressive moving-average time series, Applied Statistics 24, 255-256.

See Also

InformationMatrixARMA

Examples

```
tccfAR(0.9,0.5)
```

Index