Package ‘FitARMA’

July 5, 2007

Title ARMA fitting

Version 1.0

Date 2007-01-05

Author A.I. McLeod

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

Depends R (>= 2.0.0), FitAR

Description Implements fast maximum likelihood algorithm for fitting ARMA time series

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

R topics documented:

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, Ma
```

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, ignorned 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

McLeod, A.I. and Zhang, Y. (2007, tentatively accepted). Faster ARMA Maximum Likelihood Estimation. Computational Statistics and Data Analysis.

See Also

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

Examples

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

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	initial AR approximation
init	initial parameter estimates

Details

See McLeod and Zhang (2006). For AR(1), exact solution obtained via solving a cubic equation. For MA(1), `optimize` is used instead of `optim`. For valid likelihood computation, we need 2*pApprox >= length(y). So if the supplied value of pApprox does not satify this pApprox is set to length(n)/2. If pApprox is less than 30, a warning message is displayed.

Value

`loglikelihood`
value of maximized loglikelihood

`phiHat` estimated phi parameters

`thetaHat` estimated theta parameters

`convergence` result from `optim` if used. Otherwise 0.

`algorithm` "L-BFGS-B" or "Nelder-Mead" or "optimize" or "cubic"

Author(s)

A.I. McLeod, aimcleod@uwo.ca

References

McLeod, A.I. and Zhang, Y. (2007, tentatively accepted). Faster ARMA Maximum Likelihood Estimation. Computational Statistics and Data Analysis.

See Also

`arima`, `FitARMA`

Examples

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

`GetFitARMAMean` *MLE for ARMA parameters and mean*

Description

Usage

```
GetFitARMAMean(z, p, q, pApprox = 30, init = 0)
```

Arguments

`z`	time series
`p`	AR order
`q`	MA order
`pApprox`	AR order for the approximation
`init`	initial phi(1),...,phi(p),theta(1),...,theta(q)

Details

MLE estimation

Value

`loglikelihood`	value of maximized loglikelihood
`phiHat`	estimated phi parameters
`thetaHat`	estimated theta parameters
`convergence`	result from `optim` if used. Otherwise 0.
`algorithm`	"L-BFGS-B" or "Nelder-Mead" or "optimize" or "cubic"

Author(s)

A.I. McLeod, aimcleod@uwo.ca

References

McLeod, A.I. and Zhang, Y. (2007, tentatively accepted). Faster ARMA Maximum Likelihood Estimation. Computational Statistics and Data Analysis.

See Also

`arima`, `FitARMA`, `GetFitARMA`

Examples

```
data(SeriesA)
GetFitARMA(SeriesA, 1, 1)
```

`HRARMA` *Hannan-Rissanen ARMA estimator*

Description

Estimates the parameters in an ARMA(p,q) model.

Usage

```
HRARMA(p, q, x, AROrder = 20)
```

Arguments

`p`	AR order
`q`	MA order
`x`	time series
`AROrder`	order of approximating AR

Details

See Hannan & Rissanen (1982). This provides an efficient estimator which may be used as initial parameter values in full MLE estimation algorithms.

Value

vector with p+q parameters, phi(1),...,phi(p),theta(1),...,theta(q)

Author(s)

A.I. McLeod

References

Hannan, E.J. and Rissanen, J. (1982). Recursive estimation of mixed autoregressive-moving average order. Biometrika 69, 81-94.

See Also

FitARMA

Examples

```
data(SeriesA)
HRARMA(1,1, SeriesA)
```

`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

Details

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))
```

`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

codearima.sim

Examples

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

`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)
```

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

Description

produces table showing parameters, standard errors and Z-ratios

Usage

```
coef.FitARMA(object, ...)
```

Arguments

`object`	class "FitAR" 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)
```

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

```
fitted.FitARMA(object, ...)
```

Arguments

object	class "FitAR" 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)
```

print.FitARMA *print method for class "FitARMA"*

Description

a brief summary is printed out of the fitted model

Usage

```
print.FitARMA(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

```
residuals.FitARMA(object, ...)
```

Arguments

`object`	class "FitAR" 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)
```

`summary.FitARMA` *print method for class "FitARMA"*

Description

a summary is printed out of the fitted model

Usage

```
summary.FitARMA(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)
```

`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