The fracdiff Package

February 6, 2006

Version 1.3-0

Title Fractionally differenced ARIMA aka ARFIMA(p,d,q) models

Date 2006-02-06

Author S original by Chris Fraley <fraley@stat.washington.edu>. R port by Fritz Leisch <leisch@ci.tu-wien.ac.at>; since 2003-12: Martin Maechler; fdGPH(), fdSperio(), etc by Valderio Reisen and Artur Lemonte.

Maintainer Martin Maechler <maechler@stat.math.ethz.ch>

Description Maximum likelihood estimation of the parameters of a fractionally differenced ARIMA(p,d,q) model (Haslett and Raftery, Appl.Statistics, 1989).

License GPL version 2 or later

R topics documented:

<table>
<thead>
<tr>
<th>diffseries</th>
<th>Fractionally Differenciate Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

Index 9

Description

Diffrenciates the time series data using the approximated binomial expression of the long-memory filter and an estimate of the memory parameter in the ARFIMA(p,d,q) model.

Usage

diffseries(x, d)
fdGPH

Arguments

x numeric vector or univariate time series
d number specifying the fractional difference order.

Value

the fractionally differenced series x.

Author(s)

Valderio A. Reisen (valderio@cce.ufes.br) and Artur J. Lemonte

References

See those in fdSperio; additionally

See Also

fracdiff.sim

Examples

memory.long <- fracdiff.sim(80, d = 0.3)
mGPH <- fdGPH(memory.long$series)
r <- diffseries(memory.long$series, d = mGPH$d)
#acf(r) # shouldn't show structure - ideally

Geweke and Porter-Hudak Estimator for ARFIMA(p,d,q)

Description

Estimate the fractional (or “memory”) parameter *d* in the ARFIMA(p,d,q) model by the method of Geweke and Porter-Hudak (GPH). The GPH estimator is based on the regression equation using the periodogram function as an estimate of the spectral density.

Usage

fdGPH(x, bandw.exp = 0.5)

Arguments

x univariate time series
bandw.exp the bandwidth used in the regression equation
The function also provides the asymptotic standard deviation and the standard error deviation of the fractional estimator.

The bandwidth is $bw = \text{trunc}(n^{\text{bandw.exp}})$, where $0 < \text{bandw.exp} < 1$ and $n$ is the sample size. Default $\text{bandw.exp} = 0.5$.

Value

- $d$: GPH estimate
- $sd.as$: asymptotic standard deviation
- $sd.reg$: standard error deviation

Author(s)

Valderio A. Reisen and Artur J. Lemonte

References

see those in `fdSperio`.

See Also

`fdSperio`, `fracdiff`

Examples

```r
memory.long <- fracdiff.sim(1500, d = 0.3)
fdGPH(memory.long$series)
```

Description

This function makes use Reisen (1994) estimator to estimate the memory parameter $d$ in the ARFIMA(p,d,q) model. It is based on the regression equation using the smoothed periodogram function as an estimate of the spectral density.

Usage

```r
fdSperio(x, bandw.exp = 0.5, beta = 0.9)
```

Arguments

- $x$: univariate time series data.
- $\text{bandw.exp}$: numeric: exponent of the bandwidth used in the regression equation.
- $\text{beta}$: numeric: exponent of the bandwidth used in the lag Parzen window.
Details

The function also provides the asymptotic standard deviation and the standard error deviation of the fractional estimator.

The bandwidths are \( bw = \text{trunc}(n ^ \text{bandw.exp}) \), where \( 0 < \text{bandw.exp} < 1 \) and \( n \) is the sample size. Default \( \text{bandw.exp} = 0.5 \);

and \( bw2 = \text{trunc}(n ^ \beta) \), where \( 0 < \beta < 1 \) and \( n \) is the sample size. Default \( \beta = 0.9 \).

Value

a list with components

d Sperio estimate
sd.as asymptotic standard deviation
sd.reg standard error deviation

Author(s)

Valderio A. Reisen (valderio@cce.ufes.br) and Artur J. Lemonte

References


See Also

`fdGPH, fracdiff`

Examples

```r
memory.long <- fracdiff.sim(1500, d = 0.3)
spm <- fdSperio(memory.long$series)
str(spm, digits=6)
```

---

**fracdiff**

*ML Estimates for Fractionally-Differenced ARIMA (p,d,q) models*

Description

Calculates the maximum likelihood estimators of the parameters of a fractionally-differenced ARIMA (p,d,q) model, together (if possible) with their estimated covariance and correlation matrices and standard errors, as well as the value of the maximized likelihood. The likelihood is approximated using the fast and accurate method of Haslett and Raftery (1989).
Usage

fracdiff(x, nar = 0, nma = 0,
        ar = rep(NA, max(nar, 1)), ma = rep(NA, max(nma, 1)),
        dtol = NULL, drange = c(0, 0.5), h, M = 100)

Arguments

x       time series (numeric vector) for the ARIMA model
nar     number of autoregressive parameters \( p \).
nma     number of moving average parameters \( q \).
ar     initial autoregressive parameters
ma     initial moving average parameters
dtol    interval of uncertainty for \( d \). If dtol is negative or NULL, the fourth root of machine precision will be used. dtol will be altered if necessary by the program.
range    interval over which the likelihood function is to be maximized as a function of \( d \).
h    size of finite difference interval for numerical derivatives. By default (or if negative), \( h = \min(0.1, \varepsilon_{5} * (1 + \text{abs(cllf)}) \), where clff := log. max. likelihood (as returned) and \( \varepsilon_{5} := \text{sqrt(.Machine$double.neg.eps)} \) (typically 1.05e-8). This only influences the cov, cor, and std.error computations; see also fracdiff.var.
M     number of terms in the likelihood approximation (see Haslett and Raftery 1989).

Value

a list containing the following elements:

loglikelihood        logarithm of the maximum likelihood
d        optimal fractional-differencing parameter
ar        vector of optimal autoregressive parameters
ma        vector of optimal moving average parameters
covariance.dpq    covariance matrix of the parameter estimates (order : d, ar, ma).
stderr.dpq   standard errors of the parameter estimates c(d, ar, ma).
correlation.dpq     correlation matrix of the parameter estimates (order : d, ar, ma).
h        size of interval used for numerical derivatives.
dtol    interval of uncertainty for \( d \).
M     as input.
hessian.dpq ......

Method

The optimization is carried out in two levels: an outer univariate unimodal optimization in \( d \) over the interval drange (typically [0..5]), using Brent's fmin algorithm, and an inner nonlinear least-squares optimization in the AR and MA parameters to minimize white noise variance (uses the MINPACK subroutine lmdér). written by Chris Fraley (March 1991).
fracdiff.sim

Note
Ordinarily, nar and nma should not be too large (say < 10) to avoid degeneracy in the model. The function fracdiff.sim is available for generating test problems.

References

See Also
fracdiff.sim

Examples
ts.test <- fracdiff.sim( 5000, ar = .2, ma = -.4, d = .3)
fracdiff( ts.test$series, nar = length(ts.test$ar), nma = length(ts.test$ma))

fracdiff.sim Simulate fractional ARIMA Time Series

Description
Generates simulated long-memory time series data from the fractional ARIMA(p,d,q) model. This is a test problem generator for fracdiff.

Usage
fracdiff.sim(n, ar = NULL, ma = NULL, d,
rand.gen = rnorm, innov = rand.gen(n+q, ...),
n.start = NA, allow.0.nstart = FALSE, ..., mu = 0.)

Arguments
n length of the time series.
ar vector of autoregressive parameters; empty by default.
ma vector of moving average parameters; empty by default.
d fractional differencing parameter.
rand.gen a function to generate the innovations; the default, rnorm generates white N(0,1) noise.
innov an optional times series of innovations. If not provided, rand.gen() is used.
n.start length of “burn-in” period. If NA, the default, the same value as in arima.sim is computed.
allow.0.nstart logical indicating if n.start = 0 should be allowed even when p + q > 0. This not recommended unless for producing the same series as with older versions of fracdiff.sim.
additional arguments for `rand.gen()`. Most usefully, the standard deviation of the innovations generated by `rnorm` can be specified by `sd`.

**Value**

a list containing the following elements:

- `series` time series
- `ar, ma, d, mu, n.start` same as input

**See Also**

fracdiff, also for references; arima.sim

**Examples**

```r
## Pretty (too) short to "see" the long memory
fracdiff.sim(100, ar = .2, ma = .4, d = .3)

## longer with "extreme" ar:
r <- fracdiff.sim(n=1500, ar=-.9, d= 0.3)
plot(as.ts(r$series))
```

### fracdiff.var

**Recompute Covariance Estimate for fracdiff**

**Description**

Allows the finite-difference interval to be altered for recomputation of the covariance estimate for fracdiff.

**Usage**

```r
fracdiff.var(x, fracdiff.out, h)
```

**Arguments**

- `x` a univariate time series or a vector. Missing values (NAs) are not allowed.
- `fracdiff.out` output from `fracdiff` for time series `x`.
- `h` finite-difference interval for approximating partial derivatives with respect to the `d` parameter.

**Value**

a list with the same elements as the output to fracdiff, but with possibly different values for the hessian, covariance, and correlation matrices and for standard error, as well as for `h`.

**See Also**

fracdiff
Examples

```r
## Generate a fractionally-differenced ARIMA(1,d,1) model :
ts.test <- fracdiff.sim(10000, ar = .2, ma = .4, d = .3)
## estimate the parameters in an ARIMA(1,d,1) model for the simulated series
fd.out <- fracdiff(ts.test$ser, nar= 1, nma = 1)
## Modify the covariance estimate by changing the finite-difference interval
fracdiff.var(ts.test$series, fd.out, h = .0001)
```
Index

*Topic ts*

  diffseries, 1  
  fdGPH, 2  
  fdSperio, 3  
  fracdiff, 4  
  fracdiff.sim, 6  
  fracdiff.var, 7

arima.sim, 6, 7

diffseries, 1

fdGPH, 2, 4  
fdSperio, 2, 3, 3  
fracdiff, 3, 4, 4, 6, 7  
fracdiff.sim, 2, 5, 6, 6  
fracdiff.var, 5, 7

rnorm, 6