Package 'pcme'

March 21, 2009

Type Package

Title Maximum entropy estimation of periodically correlated time series

Version 0.51

Date 2009-03-20

Author Georgi Boshnakov and Sophie Lambert-Lacroix

Maintainer Georgi Boshnakov <georgi.boshnakov@manchester.ac.uk>

Description Given a set of autocovariances of a periodically correlated time series, find the model whose entropy is maximal among all models whose autocovariances coincide with the given ones. Some additional functions for periodically correlated models are also provided.

License GPL (>= 2)

URL http://www.maths.manchester.ac.uk/gb/Rpackages/

LazyLoad yes

R topics documented:

`00pcme-package` *Maximum entropy for periodically correlated processes.*

Description

This is an implementation of the method of Lambert-Lacroix and Boshnakov for maximum entropy completion of partially specified autocovariance functions of periodically correlated processes.

Details

Package:	pcme
Type:	Package
Version:	0.51
Date:	2009-03-21
License:	GPL (>= 2)
LazyLoad:	yes

A process $\{X_t\}$ is said to be periodically correlated of period $T > 1$ (or simply T-periodically correlated) if for all k and l

$$Cov(X_{T+k}, X_{T+k-l}) = Cov(X_k, X_{k-l}).$$

Therefore the autocovariance function of $\{X_t\}$ can be denoted by $R_k(l) = Cov(X_k, X_{k-l})$, where $k = 1, \ldots, T$ and $l = 0, 1, 2, \ldots$. We refer to (k, l) as a *season-lag pair*, where k is the season and l is the lag.

If a set of autocovariances is given and l_{max} is the maximal lag of these autocovariances, then we can arrange them in a $T \times (l_{max} + 1)$ matrix R such that $R_k(l)$ occupies the $(k, l+1)$th entry of R:

$$\begin{array}{ccccc} R_1(0) & R_1(1) & R_1(2) & \ldots & R_1(l_{max}) \\ R_2(0) & R_2(1) & R_2(2) & \ldots & R_2(l_{max}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ R_T(0) & R_T(1) & R_T(2) & \ldots & R_T(l_{max}) \end{array}$$

where some of the entries may be missing if the autocovariances are not given for every season-lag pair on $(1 : T) \times (0 : (l_{max} + 1))$.

The maximum entropy (ME) problem is to find a model whose entropy is maximal among all models whose autocovariances coincide with the given ones.

If the autocovariances are given over a set of lags which is contiguous and satisfies an additional technical condition, then the solution of the maximum entropy (ME) problem can be computed with the periodic Levinson-Durbin algorithm, otherwise the problem is non-linear. In any case the solution is a periodic autoregression model of order, say, $p_1, \ldots, p_T$. A season-lag pair is called a *gap* if $R_k(l)$ is missing and $l \leq p_k$. We single out these lags since the maximum entropy problem can be solved by filling the gaps with values that maximise the entropy over the PAR($p_1, \ldots, p_T$) models, see Boshnakov and Lacroix (2009?) for details.

The main function in this package is `pcme`. It solves the ME problem for arbitrary patterns of lags. It implements the method of Boshnakov and Lambert-Lacroix (2009?). See the examples below and also the help page of `pcme` for examples of its use.

Besides pcme, of independent interest to users may be the functions implementing the periodic Levinson-Durbin algorithm, in particular LD and pldinverse (but see also package pear by McLeod and Mehmet (2008) for this kind of functionality).

LD and pldinverse work with periodic partial autocorrelations, $\beta_k(l)$. They are arranged in a matrix in a way analogous to the arrangement of periodic autocovariances described above:

$$\begin{array}{ccccc} \beta_1(0) & \beta_1(1) & \beta_1(2) & \dots & \beta_1(l_{max}) \\ \beta_2(0) & \beta_2(1) & \beta_2(2) & \dots & \beta_2(l_{max}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \beta_T(0) & \beta_T(1) & \beta_T(2) & \dots & \beta_T(l_{max}) \end{array}$$

Also, we use the convention (see Lacroix 2005) that $\beta_k(0) = R_k(0)$ for $k = 1, \dots, T$. In this way the partial correlations completely determine the covariance structure of the process.

Functions pcme.test1 and pcme.testcombn solve a "truckload" of problems by treating subsets of a given autocovariance sequence as unknown.

Example autocovariances are given in pcme.paperex and pcme.someex.

The autocovariances needed for pcme may be theoretical or sample ones. This package does not provide functions for computing sample autocovariances, use package pear (see McLeod and Mehmet (2008)) and partsm (see López-de-Lacalle (2005)).

Author(s)

Sophie Lambert-Lacroix and Georgi Boshnakov

Maintainer: Georgi Boshnakov <Georgi.Boshnakov@manchester.ac.uk>

References

Boshnakov, Georgi and Lambert-Lacroix, Sophie (2009?) Maximum entropy for periodically correlated processes from nonconsecutive autocovariance coefficients. *J. Time Series Anal.* (to appear)

Lambert-Lacroix, Sophie (2005) Extension of autocovariance coefficients sequence for periodically correlated processes. *Journal of Time Series Analysis*, **26**, No. 6, 423-435.

Lopez-de-Lacalle, Javier (2005). partsm: Periodic Autoregressive Time Series Models. R package version 1.0.

McLeod, A. I. and Balcilar, Mehmet. (2008). pear: Package for Periodic Autoregression Analysis. R package version 1.0. http://www.r-project.org

See Also

pcme

---

pcme *Find the maximum entropy completion of a partially specified periodic autocovariance sequence*

Description

Find the maximum entropy completion of a partially specified periodic autocovariance sequence or report that it does not exist.

Usage

```
pcme(r, tau, x0 = NULL, method = "stabilNewton", gradient = TRUE,
     hessian = FALSE, autoshrink = TRUE, ...)
```

Arguments

r	the autocovariances, see Details.
tau	status of the autocovariances, see Details.
x0	initial values, see Details.
method	method to use, one of "nlm" and "stabilNewton".
gradient	logical, if TRUE use analytical gradient
hessian	logical, if TRUE use analytical Hessian
...	other arguments to be passed to the optimisation functions.
autoshrink	if TRUE automaticallly reduce the number of parameters for non-linear optimisation, see Details.

Details

The maximum entropy (ME) problem is solved by the method of Boshnakov and Lambert-Lacroix described in the reference below. A sequence of ME problems is solved until an admissible initial value is found. This sequence is finite and usually no longer than 2. Then one final maximisation solves the original problem. So, the initial value is not a problem. For other properties properties of the method and how it detects the absense of solutions or that they are only non-negative definite see the referenced paper.

The argument `x0` is not really necessary as our method efficiently finds a valid initial value. When the the ME problem has no solution, there is no valid initial value and this is also reported by the method.

By default the non-linear maximization for the modified ME problems is done with a stable hybrid Newton method implemented in this package (see `\stabilNewton`), use argument `method="nlm"` to switch to the core R function `nlm`. The results are usually comparable but in certain circumstances `nlm` may fail, while in our experience the stable version is virtually indestructible. Note that the function `stableNewton` is specifically tailored to this problem and is overly conservative with the convergence criteria.

When `method="nlm"`, `nlm` may decide that the analytical derivatives are wrong and ignore them, use the relevant argument of `nlm` if you wish to prevent this. This usually happens when the autocovariances are close to the boundary of the admissible region since the derivatives may well go bezerk without being wrong. In such cases `nlm` sometimes fails while the stable method always succeeds. It is essential to keep the parameters within the admissible region during optimisation and this is sometimes difficult with `nlm`.

If `autoshrink` is TRUE determine the order of the ME model in advance and optimise with respect to gaps within this order, see Boshnakov and Lacroix (2009?) for more information. The acf for the remaining missing values are then obtained from the periodic Yule-Walker equations. If `autoshrink` is not TRUE the order of all seasons is set to `ncol(r)-1`. The the methods give the same values for the missing autocovariances but the non-linear optimisation in the second one is in larger dimension. Also, the model fitted with `autoshrink=TRUE` is the "true" one, the excessive coefficients in the second method are theoretically zero but numerically only close to zero.

Arguments `gradient` and `hessian` are ignored by `stabilNewton`, when it is used.

Value

a list with the following components:

`entropy`	value of the entropy at thesolution
`Rg`	autocovariances at the gaps
`gaps`	the gaps as season-lag pairs
`enthist`	entropyhistory
`nf`	nf
`ck`	list of ck's
`totaloptimhistory`	
`x0`	initial value

Author(s)

Georgi Boshnakov

References

Boshnakov, Georgi and Lambert-Lacroix, Sophie (2009?) Maximum entropy for periodically correlated processes from nonconsecutive autocovariance coefficients. *J. Time Series Anal.* (to appear)

Examples

```
data(pcme.paperex)
attach(pcme.paperex)  # make rex1 directly visible

gap1p2 <- list(c(1,2))  # define some gaps
gap2p2 <- list(c(2,2))

# internally use this package's modified Newton method
resex1a <- pcme(rex1,gap1p2)
resex2a <- pcme(rex2,gap1p2)
resex3a <- pcme(rex3,gap1p2)
resex4a <- pcme(rex4,gap1p2)
resex5a <- pcme(rex5,gap2p2)

# internally use core R function "nlm"

### with nlm, without analytical derivatives
resex1 <- pcme(rex1,gap1p2,method="nlm",gradient=FALSE)
resex2 <- pcme(rex2,gap1p2,method="nlm",gradient=FALSE)
resex3 <- pcme(rex3,gap1p2,method="nlm",gradient=FALSE)
resex4 <- pcme(rex4,gap1p2,method="nlm",hessian=TRUE,check.analyticals=FALSE)
resex5 <- pcme(rex5,gap2p2,method="nlm",hessian=TRUE,check.analyticals=FALSE)

### With analytical Hessian

resex1h <- pcme(rex1,gap1p2,method="nlm",hessian=TRUE,check.analyticals=FALSE)
resex2h <- pcme(rex2,gap1p2,method="nlm",hessian=TRUE,check.analyticals=FALSE)
resex3h <- pcme(rex3,gap1p2,method="nlm",hessian=TRUE,check.analyticals=FALSE)
resex4h <- pcme(rex4,gap1p2,method="nlm",hessian=TRUE,check.analyticals=FALSE)
resex5h <- pcme(rex5,gap2p2,method="nlm",hessian=TRUE,check.analyticals=FALSE)
```

`pcme.entropy` *Compute entropy and its derivatives*

Description

Compute the entropy of a periodically correlated time series model and its derivatives with respect to unknown autocovariances.

Usage

```
sigma2.to.ent(sigma2)
sigma2.to.entgrad(sigma2, gradsigma2)
sigma2.to.enthess(sigma2, gradsigma2, grad2sigma2)
mynorm(x)
```

Arguments

`sigma2`	vector of forward prediction variances.
`gradsigma2`	gradient of forward prediction variances.
`grad2sigma2`	Hessian matrix of forward prediction variances.
`x`	a numeric vector of prediction variances.

Value

for `sigma2.to.ent` the entropy, a numeric scalar.

for `sigma2.to.entgrad` the gradient of the entropy, a numeric vector.

for `sigma2.to.enthess` the Hessian of the entropy, a matrix.

Author(s)

Sophie Lambert-Lacroix

References

Boshnakov, Georgi and Lambert-Lacroix, Sophie (2009?) Maximum entropy for periodically correlated processes from nonconsecutive autocovariance coefficients. *J. Time Series Anal.* (to appear)

Examples

```
sigma2.to.ent(c(1,0.50,0.8))   # -0.3054302
sum(log(c(1,0.50,0.8)))/3  # same

##ex From myhessian.r
R <- matrix(c(1,1,2,0.9,0.8,0.7,0.4,0.5,0.6,0.9,0.9,0.9),nrow=3)
Tau <- matrix(c(1,1,1,0,1,0,1,0,2,1,2,2),nrow=3)
R[,1]<-R[,1]+1
LD(R,Tau)

res <- LDhessian(R,Tau)

sigma2.to.ent(res$sigma2f)
```

```
sigma2.to.entgrad(res$sigma2f,res$gradsigma2f)

H <- sigma2.to.enthess(res$sigma2f,res$gradsigma2f,res$grad2sigma2f)
det(-H)
```

`pcme.paperex` *Example periodic autocovariances*

Description

Periodic autocovariances used in the referenced paper and in the examples in thispackage for illustration of maximum entropy completion.

Usage

```
data(pcme.paperex)
```

Format

A list of five 4×2 matrices, each (possibly) representing periodic autocovariances of a periodically correlated time series with period $T = 2$. The "possibly" qualifier is used because it may not be possible to complete these to complete periodic autocovariance sequences.

The ordering is the season-lag ordering used for all function in package `pcme`. Let $R_k(l) = Cov(X_k, X_{k-l})$ for $k = 1, 2$ and $l = 0, 1, 2, 3$. Then the arrangement of the autocorrelations in each matrix is as follows:

$$\begin{matrix} R_1(0) & R_1(1) & R_1(2) & R_1(3) \\ R_2(0) & R_2(1) & R_2(2) & R_2(3) \end{matrix}$$

Source

reference to a publication or URL from which the data were obtained

References

Boshnakov, Georgi and Lambert-Lacroix, Sophie (2009?) Maximum entropy for periodically correlated processes from nonconsecutive autocovariance coefficients. *Journal of Time Series Analysis* (to appear)

Lambert-Lacroix, Sophie (2005) Extension of autocovariance coefficients sequence for periodically correlated processes. *Journal of Time Series Analysis*, **26**, No. 6, 423-435.

See Also

`pcme`

Examples

```
data(pcme.paperex)
names(pcme.paperex)
pcme.paperex$rex1
```

`pcme.someex` *Example periodic autocovariances*

Description

Periodic autocovariances to use for illustration of maximum entropy completion.

Usage

```
data(pcme.someex)
```

Format

A list of five 5×2 matrices, each (possibly) representing periodic autocovariances of a periodically correlated time series with period $T = 2$. The "possibly" qualifier is used because it may not be possible to complete these to complete periodic autocovariance sequences.

The ordering is the season-lag ordering used for all function in package `pcme`. Let $R_k(l) = Cov(X_k, X_{k-l})$ for $k = 1, 2$ and $l = 0, 1, 2, 3, 4$. Then the arrangement of the autocorrelations in each matrix is as follows:

$$\begin{array}{ccccc} R_1(0) & R_1(1) & R_1(2) & R_1(3) & R_1(4) \\ R_2(0) & R_2(1) & R_2(2) & R_2(3) & R_2(4) \end{array}$$

Source

reference to a publication or URL from which the data were obtained

References

Boshnakov, Georgi and Lambert-Lacroix, Sophie (2009?) Maximum entropy for periodically correlated processes from nonconsecutive autocovariance coefficients. *J. Time Series Anal.* (to appear)

Lambert-Lacroix, Sophie (2005) Extension of autocovariance coefficients sequence for periodically correlated processes. *Journal of Time Series Analysis*, **26**, No. 6, 423-435.

See Also

`pcme`

Examples

```
data(pcme.someex)
names(pcme.someex)
pcme.someex$rex1
```

`pcme.tests` *Solve maximum entropy completion problems on subsets*

Description

Given a periodic autocovariance function, declare subsets of coefficients to be missing and maximise the entropy. Mainly for testing and examples.

Usage

```
pcme.test1(r, ...)
pcme.testcombn(r, w = 1, ...)
```

Arguments

`r`	the periodic autocovariance function, a numeric matrix.
`w`	size of gaps subsets, a numeric vector, see Details.
`...`	parameters to be passed to the entropy maximisation function.

Details

`pcme.test1` solves a sequence of maximum entropy (ME) problems with one gap. The gap is set to each of the non-zero lags, in turn.

`pcme.testcombn` solves a sequence of maximum entropy problems with number of gaps given by `w`. All subsets of `w` elements are considered. If `w` is a vector, then this is done for each element of `w`. Subsets of all lengths may be requested by setting `w="all"`.

Value

For `pcme.test1`, a matrix whose (s, l)th element is the entropy of the ME solution when the gap is set to (s, l).

For `pcme.testcombn`, a list of the entropies. Each entropy has attibutes `gaps` and `gapslist` giving the gaps as a matrix and list, respectively.

Note

We used these functions as a basis for bulk testing.

Setting `w="all"` is convenient but there may be many, many subsets to go through...

Author(s)

Georgi Boshnakov

References

Boshnakov, Georgi and Lambert-Lacroix, Sophie (2009?) Maximum entropy for periodically correlated processes from nonconsecutive autocovariance coefficients. *J. Time Series Anal.* (to appear)

See Also

`pcme`, `pcme-package`

Examples

```
data(pcme.paperex)
resex1   <- pcme.test1(pcme.paperex$rex1)
resexAll <- pcme.testcombn(pcme.paperex$rex1,"all")

data(pcme.someex)
pcme.testcombn(pcme.someex$r1,1)
pcme.testcombn(pcme.someex$r1,1:2)
resr1 <- pcme.testcombn(pcme.someex$r1,"all")
```

`pld` *Compute partial autocorrelations, forward variances and their derivatives.*

Description

Compute partial autocorrelations, forward variances, and their first and second derivatives using the periodic Levinson Durbin algorithm.

Usage

```
LD(R, Tau)
LDgrad(R, Tau)
LDhessian(R, Tau)
```

Arguments

`R`	autocovariances, see details
`Tau`	a matrix indicating which lags aregiven, see details

Details

`R` is a matrix of periodic autocovariances such that $R_k(l)$ is the $(k, l+1)$th entry of the matrix `R`, see `pcme-package` for full description.

`Tau` is a matrix of the same size as `R`. It specifies the missingness status of the elements of `R`. `Tau[k,l] = 1` if $R_k(l-1)$ is given, `Tau[k,l] = 0` if $R_k(l-1)$ is missing and $(k, l-1)$ is a gap, and `Tau[k,l] = 1` if $R_k(l-1)$ is missing but $(k, l-1)$ is not a gap. See `pcme-package` for the terminology and Boshnakov and Lambert-Lacroix (2009?) for details.

Value

For `LD`, a list with components `R`, `Beta`, and `sigma2f`, see the description below.

For `LDgrad`, a list with components `gradsigma2f` and `sigma2f`.

For `LDhessian`, a list with components `sigma2f`, `gradsigma2f`, and `grad2sigma2f`.

`R`	matrix of autocovariances, $R[k, l] = R_k(l-1)$.
`Beta`	matrix of partial autocorrelations, $Beta[k, l] = \beta_k(l-1)$.
`sigma2f`	The forward prediction variances.
`gradsigma2f`	The gradient of the forward prediction variances
`grad2sigma2f`	The Hessian of the forward prediction variances

Note

The filter coefficients have the signs used by Lambert-Lacroix (2005).

$R_k(l)$ is set to 0 for $l > p_k$ (??? check this!)

Unlike higher level functions like `pcme`, the argument `tau` here has a rigid format and gaps cannot be specified by `NA`'s in `R`.

Author(s)

Sophie Lambert-Lacroix

References

Lambert-Lacroix, Sophie (2005) Extension of autocovariance coefficients sequence for periodically correlated processes. *Journal of Time Series Analysis*, **26**, No. 6, 423-435.

See Also

`pldinverse`

Examples

```
# From file LD.R
R <- matrix(c(1,1,2,0.9,0.8,0.7,0.4,0.5,0.6,0.9,0.9,0.9),nrow=3)
Tau <- matrix(c(1,1,1,1,0,0,1,1,2,1,2,2),nrow=3)
R[Tau==2] <- 0
LD(R,Tau)

R[,1]<-R[,1]+1
LD(R,Tau)

# From file LDgrad.r
R <- matrix(c(1,1,2,0.9,0.8,0.7,0.4,0.5,0.6,0.9,0.9,0.9),nrow=3)
Tau <- matrix(c(1,1,1,1,0,0,1,1,2,1,2,2),nrow=3)
R[,1]<-R[,1]+1
LD(R,Tau)

R <- matrix(c(1,1,2,0.9,0.8,0.7,0.4,0.5,0.6,0.9,0.9,0.9),nrow=3)
Tau <- matrix(c(1,1,1,0,1,1,1,1,1,0,2,2),nrow=3)
R[,1]<-R[,1]+1
LD(R,Tau)

# From file LDhessian.r
R <- matrix(c(1,1,2,0.9,0.8,0.7,0.4,0.5,0.6,0.9,0.9,0.9),nrow=3)
Tau <- matrix(c(1,1,1,1,0,0,1,1,2,1,2,2),nrow=3)
R[,1]<-R[,1]+1
LDhessian(R,Tau)

R <- matrix(c(1,1,2,0.9,0.8,0.7,0.4,0.5,0.6,0.9,0.9,0.9),nrow=3)
Tau <- matrix(c(1,1,1,0,1,1,1,1,1,0,2,2),nrow=3)
R[,1]<-R[,1]+1
LD(R,Tau)
```

`pldinverse` *Compute autocovariances from periodic partial autocorrelations*

Description

Compute periodic autocovariances from periodic partial autocorrelations using the inverse periodic Levinson-Durbin algorithm.

Usage

```
pldinverse(beta)
```

Arguments

`beta`	numeric matrix, containing the periodic partial autocorrelations for lags 0, 1, ..., `ncol(beta)-1`, see `pcme-package` for details.

Value

A list containing the following components:

`R`	numeric matrix with the autocovariances.
`Af`	numeric matrix with the periodic filter coefficients.
`sigma2f`	numeric vector with the forward error variances.

Note

The signs of the coefficients of the filter are as in Lambert-Lacroix (2005).

Author(s)

Sophie Lambert-Lacroix

References

Boshnakov, Georgi and Lambert-Lacroix, Sophie (2009?) Maximum entropy for periodically correlated processes from nonconsecutive autocovariance coefficients. *J. Time Series Anal.* (to appear)

Lambert-Lacroix, Sophie (2005) Extension of autocovariance coefficients sequence for periodically correlated processes. *Journal of Time Series Analysis*, **26**, No. 6, 423-435.

See Also

`LD`

Examples

```
#Ex
beta <- matrix(c(3,2,0.6,0,0.85,0,-0.55,0,-0.55,0),nrow=2)
pldinverse(beta)

#Ex for ME method with gaps
#Ex1: We consider a PAR(1,2). The acf is computed for lag 0:3.
#We consider only one gap at season = 1 and lag = 2.
#By construction, we know that the solution is the one
#of the begining.
beta <- matrix(c(1,1,0.5,0.3,0,-0.3,0,0),nrow=2)
R1 <- pldinverse(beta)$R
gaps1 <- matrix(1,2,4)
gaps1[1,3] <- 0

#Ex2
beta <- matrix(c(1,1,0.9,0.3,0,-0.3,0,0),nrow=2)
R2 <- pldinverse(beta)$R
gaps2 <- matrix(1,2,4)
gaps2[1,3] <- 0

#Ex3: exemple nearly to the singularity
beta <- matrix(c(1,1,0.999999,0.3,0,-0.3,0,0),nrow=2)
R3 <- pldinverse(beta)$R
gaps3 <- matrix(1,2,4)
gaps3[1,3] <- 0

#Ex4: We consider a PAR(1,2) with singularity (one pacf coefficient equal
#to 0) and  compute the acf for lag 0:3.
#We consider only one gap at season = 1 and lag = 2.
#demontrer que la solution ME est bien degeneree
beta <- matrix(c(1,1,0.5,0.3,0,1,0,0),nrow=2)
R4 <- pldinverse(beta)$R
gaps4 <- matrix(1,2,4)
gaps4[1,3] <- 0

#Ex5: We construct one example for which there does not exist solution.
beta <- matrix(c(1,1,2,0.3),nrow=2)
R5 <- pldinverse(beta)$R
R5 <- cbind(R5,c(0.7,0.8),c(0.9,0.9))
gaps5 <- matrix(1,2,4)
gaps5[2,3] <- 0
```

`print.pcmesol` *Printing pcmesol objects*

Description

Print method for pcmesol objects.

Usage

```
## S3 method for class 'pcmesol':
print(x, digits = 4, quote = TRUE, prefix = "", ...)
```

Arguments

x	the object to be printed, typically obtained by a call to pcme.
...	additional arguments to print.

Author(s)

Georgi Boshnakov

See Also

print, pcme.

---

stabilNewton *Modified Newton method for function minimisation*

Description

Implements a hybrid method of minimisation based on gradient and Newton methods.

Usage

```
stabilNewton(f, x, alpha = 10^(-4), beta = 0.9,
             xvaltol = 10^(-6), gradtol = 10^{-12}, gradnormtol = 10^{-8},
             itermax = 100, fonly, choosedirtol = 10^(-12), betaflag = TRUE)
```

Arguments

f	evaluate the function, its gradient, and Hessian, see Details.
x	initial value.
alpha	damping constant for Newton direction.
beta	modifier for further damping in Newton direction.
xvaltol	tollerance for stopping rules.
gradtol	tollerance for stopping rules.
gradnormtol	tollerance for stopping rules.
itermax	limit for number of iterations.
fonly	evaluates f, but not derivatives.
choosedirtol	???
betaflag	???

Value

A list with the following components:

res$estimate	the value of the argument where the minimum occurs.
res$karmijo	???
res$iterations	number of iterations.

`res$minimum` f(x), the value of the function at `res$estimate`.

`res$gradient`
gradient of f at `res$estimate`.

`res$hessian` Hessian of f at `res$estimate`.

`res$cntgrad` number of computations of the gradient.

`res$cnthes` number of computations of the Hessian.

`res$optimhistory`
a log of the optimisation steps.

Note

To do: document this function and the method properly and add references.

Author(s)

Georgi Boshnakov

See Also

`nlm`

`workhorse` *Prepare a maximum entropy task*

Description

Prepare a closure for the maximum entropy task at hand. This is a relatively low-level function.

Usage

```
f.pcentropy(r, tau, method, autoshrink = TRUE)
```

Arguments

`r`	a numeric matrix of autocovariances, see Details.
`tau`	a numeric matrix of the same size as `r` or a list of season-lag pairs. Indicates the missingness status of autocovariances, see Details.
`method`	"nlm" or "stabilNewton", indicate method for numerical optimisation, see Details.
`autoshrink`	if TRUE automaticallly reduce the number of parameters for non-linear optimisation, see Details.

Details

This is a relatively low-level function. For normal use pcme should suffice.

r is a matrix with T rows such that r[k,l] is equal to $R_k(l-1)$ but the value r[k,l]=NA designates the season-lag pair (k,l) as missing.

tau is a matrix having the same size as r indicating the status of each element of r: tau[k,l+1] = 1 if $R_k(l)$ is given, tau[k,l+1] = 0 if $R_k(l)$ corresponds to a gap, and tau[k,l+1] = 2 if $R_k(l)$ is missing but (k,l) is not a gap. tau may be set to 0 for all missing season-lag pairs. By default the types of the missing values (gaps or not gaps) are determined automatically, see argument autoshrink which controls this.

tau may also be a list. Each element of this list is a season-lag pair specifying a missing value.

If method is "nlm", then the core R function nlm is used for non-linear optimisation, otherwise an implementation of the modified Newton method provided by this package is used.

If autoshrink is TRUE, then determine the order of the ME model in advance, classify the missing season-lag pairs as gaps or not gaps, and optimise with respect to the gaps, see Boshnakov and Lacroix (2009?) for more information. The acf values for the remaining missing values are then obtained from the periodic Yule-Walker equations.

If autoshrink is not TRUE the order of all seasons is set to ncol(r)-1. The two methods give the same values (numerically) for the missing autocovariances but the non-linear optimisation in the second one is in larger dimension. Also, the model fitted with autoshrink=TRUE is the "true" one, the excessive coefficients in the second method are theoretically zero but numerically only close to zero.

The list returned by f.pcentropy contains functions of several types.

mesolve is the function to call to solve the ME problem. It can be called without arguments. Argument xinit is effectively redundant as mesolve has a very efficient way of locating initial condition. xinit was used for testing the behaviour of the algorithm with extremely weird initial conditions. If xinit is a scalar (usually 0), then all gaps are filled initially with this scalar. In fact, f.pcentropy initialises the gaps with whatever value were found in the corresponding places in the autocovariance function (??? does it deal with NA's?). Since the autocovariances have two indices, it is important when giving initial values to put them in the vector in the same order as mesolve expects. If the length of xinit is greater than 1 and does not match the number of gaps, it is ignored. In short, specifying xinit is hardly needed but cannot do any harm.

f, fg, fgh implement the periodic Levinson-Durbin algorithm and its extension to calculate the entropy, its gradient and Hessian. f calculates entropy only, fg entropy and gradient, and fgh all three. The value returned by these functions is a scalar (the entropy) with attributtes "gradient" and "hessian" for the derivatives.

x0 is an initial value and is normally omitted for the ME problem.

pdq tests if the current autocovariances are p.d. completable. makepdq makes sure that this is so by adding a sufficiently large constant to the lag zero autocovariances.

ftau and ftaufull give the matrix tau describing the missingness of season-lag pairs. ftaufull is the one obtained after interpreting the input arguments, ftau is the one actually used for optimisation. ftau may have fewer columns if autoshrink=TRUE.

findexgaps and ffreelags give the gaps and the non-gaps missing season-lag pairs as two column matrices with each row containing the position of a season-lag pair in ftau (so the corresponding lags are obtained by subtracting 1 from the second column). (??? check this description!)

After a run of mesolve, fent contains the entropy at the optimal point.

p is the model order.

fnf, fckvec, and totalhistory contin information about the running of the algorithm. (??? Add argument to suppress output and storage of this information.)

Value

The function creates a closure for the maximum entropy task and returns a list whose elements give the user access to the relevant parts of this closure.

`mesolve`	function that solves the ME problem.
`f`	the function being optimised.
`fg`	the function being optimised and its gradient.
`fgh`	the function being optimised, its gradient, and Hessian.
`x0`	function that returns the initial value
`pdq`	function to check if an acf sequence is positive definite.
`makepd`	function that makes an acf positive definite.
`ftau`	a matrix describing the gaps.
`ftaufull`	a matrix describing the gaps.
`findexgaps`	???
`ffreelags`	???
`fent`	a function returning the a history of the entropy during the optimisation process.
`p`	model order
`fnf`	function returning the number of evaluations of `f`.
`fckvec`	???
`totalhistoty`	function returning comprehensive history of the calculation.

Author(s)

Georgi Boshnakov

References

Boshnakov, Georgi and Lambert-Lacroix, Sophie (2009?) Maximum entropy for periodically correlated processes from nonconsecutive autocovariance coefficients. *J. Time Series Anal.* (to appear)

Lambert-Lacroix, Sophie (2005) Extension of autocovariance coefficients sequence for periodically correlated processes. *Journal of Time Series Analysis*, **26**, No. 6, 423-435.

See Also

`pcme`

Index