

パッケージ ‘dlm’

April 17, 2013

Title Bayesian and Likelihood Analysis of Dynamic Linear Model (動的線型モデルに関するベイズ分析と尤度分析)

Version 1.1-2

Date 2010-10-05

Suggests MASS

Author Giovanni Petris <GPetris@uark.edu>

Description Maximum likelihood (最尤法), Kalman filtering and smoothing (カルマン・フィルタリングとカルマン平滑化), and Bayesian analysis of Normal linear State Space models, also known as Dynamic Linear Models (動的線型モデルとしても知られる正規線型状態空間モデルに関するベイズ分析)

Maintainer Giovanni Petris <GPetris@uark.edu>

License GPL (>= 2)

Repository CRAN

Date/Publication 2010-10-05 16:59:18

記載のある R のトピックス

arms	3
ARtransPars	5
bdiag	6
convex.bounds	7
dlm	7
dlmBSample	9
dlmFilter	10
dlmForecast	11
dlmGibbsDIG	13
dlmLL	14
dlmMLE	15
dlmModARMA	17
dlmModPoly	18
dlmModReg	19
dlmModSeas	20
dlmModTrig	21

d1mRandom	22
d1mSmooth	23
d1mSum	25
d1mSvd2var	25
dropFirst	26
FF	27
mcmc	29
NelPlo	30
residuals.d1mFiltered	30
rwishart	32
USecon	33

arms

適応的棄却メトロポリスサンプリングを実行する関数

説明

ARMS を用いて確率変数列を生成する。多変量の密度に関しては現在の点を通る直線を複数ランダムに選択し、それらの直線に沿って ARMS を適用する。

用法

```
arms(y.start, myldens, indFunc, n.sample, ...)
```

引数

y.start	開始点
myldens	一変量もしくは多変量の対数目標密度
indFunc	目標密度が凸となる台に関する指標関数
n.sample	必要な標本の数
...	myldens と indFunc に渡されるパラメータ

詳細

厳密に言うと、目標密度の台は有界の凸集合になる必要がある。この条件が成立しない場合は、通常次のようなトリックが使われる。台が有界でない場合は、実質的に確率が 1 となる有界の集合に制限を行う。台が凸でない場合の対処法としては、その台で凸集合を生成し、その凸集合以外の台では myldens が `log(.Machine$double.xmin)` を返す様に考える (例の最後を参照)。

現在の点を通る直線をランダムに選択し、その直線に沿って arms を用いることで次の点が生成される。

myldens から返された値は、決して `log(.Machine$double.xmin)` より小さくなる事がないように注意してほしい。これはゼロ除算を避けるためである。

返り値

`n.sample × length(y.start)` の行列であり、列方向が標本点となる。

注記

この関数は、W.Gilks によるオリジナルの C のコード (一変量の場合用、http://www.mrc-bsu.cam.ac.uk/pub/methodology/adaptive_rejection/ を参照) に基づいている。

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Gilks, W.R., Best, N.G. and Tan, K.K.C. (1995) Adaptive rejection Metropolis sampling within Gibbs sampling (Corr: 97V46 p541-542 with Neal, R.M.), *Applied Statistics* **44**:455–472.

例

```
#### ==> Warning: running the examples may take a few minutes! <== ####
## Not run:
set.seed(4521222)
### Univariate densities
## Unif(-r,r)
y <- arms(runif(1,-1,1), function(x,r) 1, function(x,r) (x>-r)*(x<r), 5000, r=2)
summary(y); hist(y,prob=TRUE,main="Unif(-r,r); r=2")
## Normal(mean,1)
norldens <- function(x,mean) -(x-mean)^2/2
y <- arms(runif(1,3,17), norldens, function(x,mean) ((x-mean)>-7)*((x-mean)<7),
          5000, mean=10)
summary(y); hist(y,prob=TRUE,main="Gaussian(m,1); m=10")
curve(dnorm(x,mean=10),3,17,add=TRUE)
## Exponential(1)
y <- arms(5, function(x) -x, function(x) (x>0)*(x<70), 5000)
summary(y); hist(y,prob=TRUE,main="Exponential(1)")
curve(exp(-x),0,8,add=TRUE)
## Gamma(4.5,1)
y <- arms(runif(1,1e-4,20), function(x) 3.5*log(x)-x,
          function(x) (x>1e-4)*(x<20), 5000)
summary(y); hist(y,prob=TRUE,main="Gamma(4.5,1)")
curve(dgamma(x,shape=4.5,scale=1),1e-4,20,add=TRUE)
## Gamma(0.5,1) (this one is not log-concave)
y <- arms(runif(1,1e-8,10), function(x) -0.5*log(x)-x,
          function(x) (x>1e-8)*(x<10), 5000)
summary(y); hist(y,prob=TRUE,main="Gamma(0.5,1)")
curve(dgamma(x,shape=0.5,scale=1),1e-8,10,add=TRUE)
## Beta(.2,.2) (this one neither)
y <- arms(runif(1), function(x) (0.2-1)*log(x)+(0.2-1)*log(1-x),
          function(x) (x>1e-5)*(x<1-1e-5), 5000)
summary(y); hist(y,prob=TRUE,main="Beta(0.2,0.2)")
curve(dbeta(x,0.2,0.2),1e-5,1-1e-5,add=TRUE)
## Triangular
y <- arms(runif(1,-1,1), function(x) log(1-abs(x)), function(x) abs(x)<1, 5000)
summary(y); hist(y,prob=TRUE,ylim=c(0,1),main="Triangular")
curve(1-abs(x),-1,1,add=TRUE)
## Multimodal examples (Mixture of normals)
lmixnorm <- function(x,weights,means,sds) {
  log(crossprod(weights, exp(-0.5*((x-means)/sds)^2 - log(sds))))
}
y <- arms(0, lmixnorm, function(x,...) (x>(-100))*(x<100), 5000, weights=c(1,3,2),
          means=c(-10,0,10), sds=c(1.5,3,1.5))
summary(y); hist(y,prob=TRUE,main="Mixture of Normals")
curve(colSums(c(1,3,2)/6*dnorm(matrix(x,3,length(x),byrow=TRUE),c(-10,0,10),c(1.5,3,1.5))),
      par("usr")[1], par("usr")[2], add=TRUE)

### Bivariate densities
## Bivariate standard normal
y <- arms(c(0,2), function(x) -crossprod(x)/2,
          function(x) (min(x)>-5)*(max(x)<5), 500)
plot(y, main="Bivariate standard normal", asp=1)
## Uniform in the unit square
y <- arms(c(0.2,.6), function(x) 1,
          function(x) (min(x)>0)*(max(x)<1), 500)
plot(y, main="Uniform in the unit square", asp=1)
polygon(c(0,1,1,0),c(0,0,1,1))
## Uniform in the circle of radius r
y <- arms(c(0.2,0), function(x,...) 1,
          function(x,r2) sum(x^2)<r2, 500, r2=2^2)
plot(y, main="Uniform in the circle of radius r; r=2", asp=1)
curve(-sqrt(4-x^2), -2, 2, add=TRUE)
curve(sqrt(4-x^2), -2, 2, add=TRUE)
## Uniform on the simplex
simp <- function(x) if ( any(x<0) || (sum(x)>1) ) 0 else 1
```

```

y <- arms(c(0.2,0.2), function(x) 1, simp, 500)
plot(y, xlim=c(0,1), ylim=c(0,1), main="Uniform in the simplex", asp=1)
polygon(c(0,1,0), c(0,0,1))
## A bimodal distribution (mixture of normals)
bimodal <- function(x) { log(prod(dnorm(x,mean=3))+prod(dnorm(x,mean=-3))) }
y <- arms(c(-2,2), bimodal, function(x) all(x>(-10))*all(x<(10)), 500)
plot(y, main="Mixture of bivariate Normals", asp=1)

## A bivariate distribution with non-convex support
support <- function(x) {
  return(as.numeric( -1 < x[2] && x[2] < 1 &&
                    -2 < x[1] &&
                    ( x[1] < 1 || crossprod(x-c(1,0)) < 1 ) ) )
}
Min.log <- log(.Machine$double.xmin) + 10
logf <- function(x) {
  if ( x[1] < 0 ) return(log(1/4))
  else
    if (crossprod(x-c(1,0)) < 1 ) return(log(1/pi))
  return(Min.log)
}
x <- as.matrix(expand.grid(seq(-2.2,2.2,length=40),seq(-1.1,1.1,length=40)))
y <- sapply(1:nrow(x), function(i) support(x[i,]))
plot(x,type=n,asp=1)
points(x[y==1,],pch=1,cex=1,col=green)
z <- arms(c(0,0), logf, support, 1000)
points(z,pch=20,cex=0.5,col=blue)
polygon(c(-2,0,0,-2),c(-1,-1,1,1))
curve(-sqrt(1-(x-1)^2),0,2,add=TRUE)
curve(sqrt(1-(x-1)^2),0,2,add=TRUE)
sum( z[,1] < 0 ) # sampled points in the square
sum( apply(t(z)-c(1,0),2,crossprod) < 1 ) # sampled points in the circle

## End(Not run)

```

ARtransPars

定常 AR 過程のパラメータを求める関数

説明

この関数は長さ p のベクトルを、定常 $AR(p)$ 過程の自己回帰係数ベクトルに写像する。この関数を用いると、定常 $AR(p)$ 過程のパラメータを求めることができる。

用法

```
ARtransPars(raw)
```

引数

raw 長さ p のベクトル

詳細

この関数では、まず raw の各要素を \tanh を用いて $(0, 1)$ 区間に写像する。こうして得られた数値は定常 $AR(p)$ 過程の 1 次偏自己相関として扱われ、対応する自己回帰係数のベクトルが計算され返される。

返り値

パラメータ `raw` に対応する定常 $AR(p)$ 過程の自己回帰係数ベクトル

作者

Giovanni Petris, <GPetris@uark.edu>

参考文献

Jones, 1987. Randomly choosing parameters from the stationarity and invertibility region of autoregressive-moving average models. *Applied Statistics*, 36.

例

```
(ar <- ARtransPars(rnorm(5)))  
all( Mod(polyroot(c(1,-ar))) > 1 ) # TRUE
```

bdiag

ブロック対角行列の構築

説明

この関数はブロック対角行列を構築する。

用法

```
bdiag(...)
```

引数

... 個々の行列、もしくは行列のリスト

返り値

引数を組み合わせて得られる行列

作者

Giovanni Petris <GPetris@uark.edu>

例

```
bdiag(matrix(1:4,2,2),diag(3))  
bdiag(matrix(1:6,3,2),matrix(11:16,2,3))
```

convex.bounds	凸集合の境界を見つける
---------------	-------------

説明

二分法を用いて、指定された直線に沿って有界な凸集合の境界を見つける。この関数は、主に `arms` の中で使用される事を意図して作られている。

用法

```
convex.bounds(x, dir, indFunc, ..., tol=1e-07)
```

引数

<code>x</code>	集合の中にある点
<code>dir</code>	方向を指定するベクトル
<code>indFunc</code>	集合に関する指標関数
<code>...</code>	<code>indFunc</code> に渡されるパラメータ
<code>tol</code>	許容誤差

詳細

パラメトリックな表現の直線 $x + t * dir$ に沿って、二分法が用いられる。

返り値

長さ 2 のベクトル `ans`。集合の境界は、 $x + ans[1] * dir$ と $x + ans[2] * dir$ になる。

作者

Giovanni Petris <GPetris@uark.edu>

例

```
## boundaries of a unit circle
convex.bounds(c(0,0), c(1,1), indFunc=function(x) crossprod(x)<1)
```

d1m	d1m オブジェクト
-----	------------

説明

関数 `d1m` は、動的線型モデルのオブジェクトを生成するために使用される。`as.d1m` と `is.d1m` では、引数のオブジェクトは動的線型モデルのオブジェクトであるか否かを確認されずに、強制的に動的線型モデルのオブジェクトに変換される。

用法

```
dlm(...)
as.dlm(obj)
is.dlm(obj)
```

引数

... 名前付きの要素 (m0, C0, FF, V, GG, W、更にオプションで JFF, JV, JGG, JW, X) を持つリスト。この要素のうち最初の 6 つは、時不変の DLM を定義する通常のベクトルや行列である。残りの要素は、時変の DLM のために使用される。X が存在していれば、必ず行列になる。JFF が NULL でなければ FF と同じ次元の行列となる必要があり、その (i, j) 要素は、 $FF[i, j]$ が時不変であれば 0 となり、そうでなければ正の整数 k となる。後者の時変の場合、時点 t における FF の (i, j) 要素は、 $X[t, k]$ になる。同様の解釈が JV, JGG および JW にも成立する。... には、dlm では使用されない追加要素が含まれる可能性がある。なお、この関数には個別の引数として名前付きの要素を渡しても良い。

obj 任意の R のオブジェクト

詳細

関数 dlm は、動的線型モデルのオブジェクトを生成するために用いられる。生成されるオブジェクトは、上述の名前付き要素を持つリストにクラス属性 "dlm" が与えられたものとなる。

クラス "dlm" には、多数のメソッドが存在する。特に、整合性のある DLM であればそれらをあわせることで、もうひとつ別の DLM を作る事が出来る。

返り値

関数 dlm に関しては、"dlm" クラスのオブジェクトとなる。

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Giovanni Petris (2010), An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1-16. <http://www.jstatsoft.org/v36/i12/>.

Petris, Petrone, and Campagnoli, Dynamic Linear Models with R, Springer (2009).

West and Harrison, Bayesian forecasting and dynamic models (2nd ed.), Springer (1997).

関連事項

dlmModReg, dlmModPoly, dlmModARMA, dlmModSeas は、"dlm" クラスの特定のオブジェクトを生成する。

例

```
## Linear regression as a DLM
x <- matrix(rnorm(10),nc=2)
mod <- dlmModReg(x)
is.dlm(mod)
```

```
## Adding dlms
d1mModPoly() + d1mModSeas(4) # linear trend plus quarterly seasonal component
```

d1mBSample

状態ベクトルに関する事後分布からの標本抽出

説明

この関数は状態ベクトルの事後分布から、標本抽出を 1 回分シミュレートする。

用法

```
d1mBSample(modFilt)
```

引数

modFilt 要素 m, U.C, D.C, a, U.R, D.R, mod を持つリストであるが、典型的には d1mFilter からの出力となる (d1mFilter の返り値を参照してほしい)。ここで、mod は "d1m" クラスのオブジェクト、もしくは要素 GG, W (更にオプションで JGG, JW, X) を持つリストである。

詳細

ここでの計算は特異値分解に基づく。

返り値

この関数は状態ベクトルの事後分布から抽出された標本を返す。返り値は、m が時系列ならばそれと `tsp` が同じ時系列となり、そうでなければ行列かベクトルになる。

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Giovanni Petris (2010), An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1-16. <http://www.jstatsoft.org/v36/i12/>.
 Petris, Petrone, and Campagnoli, Dynamic Linear Models with R, Springer (2009).
 West and Harrison, Bayesian forecasting and dynamic models (2nd ed.), Springer (1997).

関連事項

d1mFilter も参照されたい。

例

```
nileMod <- d1mModPoly(1, dV = 15099.8, dW = 1468.4)
nileFilt <- d1mFilter(Nile, nileMod)
nileSmooth <- d1mSmooth(nileFilt) # estimated "true" level
plot(cbind(Nile, nileSmooth$S[-1]), plot.type = "s",
     col = c("black", "red"), ylab = "Level",
     main = "Nile river", lwd = c(2, 2))
for (i in 1:10) # 10 simulated "true" levels
  lines(d1mBSample(nileFilt[-1]), lty=2)
```

dlmFilter

DLM のフィルタリング

説明

この関数ではカルマン・フィルタを用いて、状態ベクトルのフィルタ化値とその分散/共分散行列を計算する。この関数はデフォルトでは "dlmFiltered" クラスのオブジェクトを返す。"dlmFiltered" クラスのオブジェクトについては、`residuals` と `tsdiag` に関するメソッドが存在する。

用法

```
dlmFilter(y, mod, debug = FALSE, simplify = FALSE)
```

引数

<code>y</code>	データ。y はベクトル、行列、一変量もしくは多変量の時系列となる可能性がある。
<code>mod</code>	dlm クラスのオブジェクト、もしくはモデルや事前分布のパラメータを定義する要素 <code>m0</code> , <code>C0</code> , <code>FF</code> , <code>V</code> , <code>GG</code> , <code>W</code> (更にオプションで <code>JV</code> , <code>JGG</code> , <code>JW</code> , <code>X</code>) を持つリスト。
<code>debug</code>	FALSE であればより高速な C のコードが使用されるが、そうでなければ計算はすべて R で実行される。
<code>simplify</code>	出力にデータが含まれるべきか否か?

詳細

ここで行われる計算は、関連する行列の特異値分解 (SVD) に基づく。
y における欠損値は許容される。

返り値

以下に記述する成分を持つリスト。simplify が FALSE であれば、返り値のリストは "dlmFiltered" クラスとなる。

<code>y</code>	行列に変換された入力データ。simplify が FALSE の場合のみ存在する。
<code>mod</code>	引数 mod と同じ (簡素化される可能性あり)。
<code>m</code>	状態ベクトルのフィルタ化値に関する時系列 (もしくは行列)。この系列は、最初の観測値の一時点前から始まる。
<code>U.C</code>	次の項目を参照
<code>D.C</code>	U.C と組み合わせて、推定誤差の分散に関する SVD を構築する。 $m[t,] - \theta[t,]$ の分散は、 $U.C[[t]] \%*\% \text{diag}(D.C[t,]^2) \%*\% t(U.C[[t]])$ によって与えられる。
<code>a</code>	観測値が与えられた下における状態ベクトルの予測値に関する時系列 (もしくは行列) であり、最初の観測値の一時点前の値も含まれる。
<code>U.R</code>	次の項目を参照
<code>D.R</code>	U.R と組み合わせて、予測誤差の分散に関する SVD を構築する。 $a[t,] - \theta[t,]$ の分散は、 $U.R[[t]] \%*\% \text{diag}(D.R[t,]^2) \%*\% t(U.R[[t]])$ によって与えられる。
<code>f</code>	観測値の一期先予測に関する時系列 (もしくは行列)。

警告

mod における観測分散 v は非特異でなければならない。

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Zhang, Y. and Li, X.R., Fixed-interval smoothing algorithm based on singular value decomposition, *Proceedings of the 1996 IEEE International Conference on Control Applications*.
 Giovanni Petris (2010), An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1-16. <http://www.jstatsoft.org/v36/i12/>.
 Petris, Petrone, and Campagnoli, Dynamic Linear Models with R, Springer (2009).

関連事項

d1m オブジェクトの説明に関しては d1m、SVD から分散行列を得る方法に関しては d1mSvd2var、最尤推定に関しては d1mMLE、カルマン平滑化に関しては d1mSmooth、状態ベクトルの事後分布から標本抽出を行う方法に関しては d1mBSample を、それぞれ参照されたい。

例

```
nileBuild <- function(par) {
  d1mModPoly(1, dV = exp(par[1]), dW = exp(par[2]))
}
nileMLE <- d1mMLE(Nile, rep(0,2), nileBuild); nileMLE$conv
nileMod <- nileBuild(nileMLE$par)
V(nileMod)
W(nileMod)
nileFilt <- d1mFilter(Nile, nileMod)
nileSmooth <- d1mSmooth(nileFilt)
plot(cbind(Nile, nileFilt$m[-1], nileSmooth$s[-1]), plot.type=s,
     col=c("black","red","blue"), ylab="Level", main="Nile river", lwd=c(1,2,2))
```

d1mForecast

将来の観測値に関する予測やシミュレーション

説明

この関数は、将来の観測値やシステムの状態に関する期待値と分散を評価する。また、将来の観測値やシステムの状態に関する分布から標本を生成することもできる。

用法

```
d1mForecast(mod, nAhead = 1, method = c("plain", "svd"),
             sampleNew = FALSE)
```

引数

mod	"dlm" クラスのオブジェクト、もしくはリスト (要素は m0, C0, FF, V, GG, W であり、これらはモデルや事前分布のパラメータを定義する)。mod は "dlmFiltered" クラスのオブジェクト (例えば dlmFilter からの出力) でもよい。
nAhead	予測で必要とされる前方ステップ数
method	method="svd" であれば、計算に特異値分解が用いられる。現在実装されているのは、method="plain" のみである。
sampleNew	sampleNew=n (n は整数) ならば、状態や観測値の予測分布から返される標本の数が n となる。

返り値

次の成分を持つリスト

a	将来の状態に関する期待値 (行列)
R	将来の状態に関する分散 (リスト)
f	将来の観測値に関する期待値 (行列)
Q	将来の観測値に関する分散 (リスト)
newStates	シミュレートされた将来の状態の値を含む行列のリスト。 リストの各成分が 1 回分のシミュレーションに対応している。
newObs	newStates と同様だが観測値用

sampleNew=FALSE であれば、最後の 2 つの成分は存在しない。

注記

現在この関数は全て R で書かれているため、特段早いというわけではない。現時点で許容されるモデルは、時不変のモデルのみである。

作者

Giovanni Petris <GPetris@uark.edu>

例

```
## Comparing theoretical prediction intervals with sample quantiles
set.seed(353)
n <- 20; m <- 1; p <- 5
mod <- dlmModPoly() + dlmModSeas(4, dV=0)
W(mod) <- rwishart(2*p,p) * 1e-1
m0(mod) <- rnorm(p, sd=5)
C0(mod) <- diag(p) * 1e-1
new <- 100
fore <- dlmForecast(mod, nAhead=n, sampleNew=new)
ciTheory <- (outer(sapply(fore$Q, FUN=function(x) sqrt(diag(x))), qnorm(c(0.1,0.9))) +
  as.vector(t(fore$f)))
ciSample <- t(apply(array(unlist(fore$newObs), dim=c(n,m,new))[,1,], 1,
  FUN=function(x) quantile(x, c(0.1,0.9)))))
plot.ts(cbind(ciTheory,fore$f[,1]),plot.type="s", col=c("red","red","green"),ylab="y")
for (j in 1:2) lines(ciSample[,j], col="blue")
legend(2,-40,legend=c("forecast mean", "theoretical bounds", "Monte Carlo bounds"),
  col=c("green","red","blue"), lty=1, bty="n")
```

dlmGibbsDIG

 d 個の逆ガンマモデルに関するギブスサンプリング

説明

この関数で実装されているのはギブスサンブラであり、対象となるモデルは特定化において未知の分散が一つ以上存在する一変量の DLM である。

用法

```
dlmGibbsDIG(y, mod, a.y, b.y, a.theta, b.theta, shape.y, rate.y,
            shape.theta, rate.theta, n.sample = 1,
            thin = 0, ind, save.states = TRUE,
            progressBar = interactive())
```

引数

<code>y</code>	データ (ベクトルか一変量の時系列)
<code>mod</code>	観測値が一変量の dlm
<code>a.y</code>	観測値の精度に関する事前分布の平均
<code>b.y</code>	観測値の精度に関する事前分布の分散
<code>a.theta</code>	状態の精度に関する事前分布の平均 (必要ならリサイクルされる)
<code>b.theta</code>	状態の精度に関する事前分布の分散 (必要ならリサイクルされる)
<code>shape.y</code>	観測値の精度に関する事前分布の形状パラメータ
<code>rate.y</code>	観測値の精度に関する事前分布の尺度パラメータ
<code>shape.theta</code>	状態の精度に関する事前分布の形状パラメータ (必要ならリサイクルされる)
<code>rate.theta</code>	状態の精度に関する事前分布の尺度パラメータ (必要ならリサイクルされる)
<code>n.sample</code>	ギブスサンブラに求められる反復回数
<code>thin</code>	破棄される反復結果の回数 (<code>thin+1</code> 回に一回毎に保存が行われる)
<code>ind</code>	推定される必要がある状態の分散に関する指標 (順番を連ねたベクトル)
<code>save.states</code>	シミュレートされた状態を出力に含めるべきか否か?
<code>progressBar</code>	関数を実行している間、テキスト型の進捗バーを表示すべきか否か?

詳細

d 個の逆ガンマモデルは時不変の一変量 DLM であり、その観測分散は未知であり、状態分散 (対角行列の対角要素) も未知である。これらの要素のいくつかは既知であるかもしれないが、その場合通常それらは 0 となる。未知の分散に対しては、独立な逆ガンマ事前分布を仮定する。この様な分布は、平均と分散、もしくは形状と尺度によって規定することができる。未知の状態分散に関する事前分布のパラメータには、リサイクル規則が適用される。 w の対角要素におけるいくつかが既知の場合、引数 `ind` を用いれば、状態分散が未知である部分を指定することができる。`save.states = TRUE` の場合、ギブスサンブラで生成された観測不可能な状態が返される。モデルや使用例に関する更なる詳細については、パッケージのビニエットを参照されたい。

返り値

この関数はシミュレートされた値をリストにして返す。

dV	観測分散のシミュレーション値
dW	状態分散（未知の対角要素）のシミュレーション値
theta	状態ベクトルのシミュレーション値

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Giovanni Petris (2010), An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1-16. <http://www.jstatsoft.org/v36/i12/>.
 Petris, Petrone, and Campagnoli, Dynamic Linear Models with R, Springer (2009).

例

```
## See the package vignette for an example
```

dlmLL	状態空間モデルに関する対数尤度の評価
-------	--------------------

説明

この関数は状態空間モデルの対数尤度を計算する。

用法

```
dlmLL(y, mod, debug=FALSE)
```

引数

y	データ（ベクトル、行列、もしくは時系列）
mod	"dlm" クラスのオブジェクト、もしくはモデルや事前分布のパラメータを定義する要素 m0, C0, FF, V, GG, W を持つリスト。
debug	debug=TRUE なら R のコード、そうでなければより高速な C のコードがこの関数で使われる。

詳細

ここで行われる計算は、特異値分解に基づく。y における欠損値は許容される。

返り値

この関数は対数尤度の符号を反転して返す。

警告

mod における観測分散 V は非特異でなければならない。

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Durbin and Koopman, Time series analysis by state space methods, Oxford University Press, 2001.
[邦訳] J. ダービン・S.J. クープマン著. 和合肇・松田安昌訳. (2004). 状態空間モデリングによる時系列分析入門. シーエーピー出版.

関連事項

d1mMLE、モデルの方程式の定義に関しては d1mFilter を参照されたい。

例

```
##---- See the examples for d1mMLE ----
```

d1mMLE	最尤法によるパラメータ推定
--------	---------------

説明

この関数は、状態空間モデルの特定化において未知パラメータの MLE を返す。

用法

```
d1mMLE(y, parm, build, method = "L-BFGS-B", ..., debug = FALSE)
```

引数

y	データ（ベクトル、行列、もしくは時系列）
parm	未知パラメータに関する初期値のベクトル（最適化ルーチンに渡される）
build	parm と同じ長さのベクトルを受け取って、d1m クラスのオブジェクト（もしくはそう解釈できるリスト）を返す関数
method	この引数は optim に渡される
...	optim と build に渡される追加引数
debug	尤度の計算は、debug=TRUE なら全て R で行われ、そうでなければ C の関数が用いられる。

詳細

対数尤度の評価は d1mLL により行われる。最適化に関しては optim が呼ばれる。モデルを parm 以外の追加パラメータ（引数... を通じて build に渡される）に依存させることもできる。

返り値

関数 d1mMLE は optim から返された値を返す。

警告

引数 `build` が返す `dlm` において、観測分散 V は非特異でなければならない。

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Giovanni Petris (2010), An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1-16. <http://www.jstatsoft.org/v36/i12/>.

Petris, Petrone, and Campagnoli, Dynamic Linear Models with R, Springer (2009).

関連事項

`dlmLL`, `dlm`

例

```
data(NelPlo)
### multivariate local level -- seemingly unrelated time series
buildSu <- function(x) {
  Vsd <- exp(x[1:2])
  Vcorr <- tanh(x[3])
  V <- Vsd %o% Vsd
  V[1,2] <- V[2,1] <- V[1,2] * Vcorr
  Wsd <- exp(x[4:5])
  Wcorr <- tanh(x[6])
  W <- Wsd %o% Wsd
  W[1,2] <- W[2,1] <- W[1,2] * Wcorr
  return(list(
    m0 = rep(0,2),
    C0 = 1e7 * diag(2),
    FF = diag(2),
    GG = diag(2),
    V = V,
    W = W))
}

suMLE <- dlmMLE(NelPlo, rep(0,6), buildSu); suMLE
buildSu(suMLE$par)[c("V","W")]
StructTS(NelPlo[,1], type="level") ## compare with W[1,1] and V[1,1]
StructTS(NelPlo[,2], type="level") ## compare with W[2,2] and V[2,2]

## multivariate local level model with homogeneity restriction
buildHo <- function(x) {
  Vsd <- exp(x[1:2])
  Vcorr <- tanh(x[3])
  V <- Vsd %o% Vsd
  V[1,2] <- V[2,1] <- V[1,2] * Vcorr
  return(list(
    m0 = rep(0,2),
    C0 = 1e7 * diag(2),
    FF = diag(2),
    GG = diag(2),
    V = V,
    W = x[4]^2 * V))
}

hoMLE <- dlmMLE(NelPlo, rep(0,4), buildHo); hoMLE
buildHo(hoMLE$par)[c("V","W")]
```

dlmModARMA

ARMA 過程の DLM 表現を生成する

説明

この関数は、dlm クラスのオブジェクト（一変量か多変量の特定の ARMA 過程を表す）を生成する。

用法

```
dlmModARMA(ar = NULL, ma = NULL, sigma2 = 1, dV, m0, C0)
```

引数

ar	自己回帰係数を含むベクトル、もしくは行列のリスト (多変量の場合)
ma	移動平均係数を含むベクトル、もしくは行列のリスト (多変量の場合)
sigma2	イノベーションの分散 (もしくは分散行列)
dV	観測雑音の分散 (多変量の場合は分散行列の対角要素)。V は対角行列とし、そのデフォルト値を 0 と仮定する。
m0	m_0 であり、状態ベクトルの事前分布に関する期待値
C0	C_0 であり、状態ベクトルの事前分布に関する分散行列

詳細

ARMA 過程の DLM 表現には多数の可能性があるが、この関数の返り値によって与えられるのは、その中の 1 つだけである。

返り値

この関数は dlm クラスのオブジェクト (ar, ma, sigma2 によって特定される ARMA モデルを表す) を返す。

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Giovanni Petris (2010), An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1-16. <http://www.jstatsoft.org/v36/i12/>.
 Petris, Petrone, and Campagnoli, Dynamic Linear Models with R, Springer (2009).
 Durbin and Koopman, Time series analysis by state space methods, Oxford University Press, 2001.
 [邦訳] J. ダービン・S.J. クープマン著. 和合肇・松田安昌訳. (2004). 状態空間モデリングによる時系列分析入門. シーエーピー出版.

関連事項

dlmModPoly, dlmModSeas, dlmModReg

例

```
## ARMA(2,3)
dlmModARMA(ar = c(.5,.1), ma = c(.4,2,.3), sigma2=1)
## Bivariate ARMA(2,1)
dlmModARMA(ar = list(matrix(1:4,2,2), matrix(101:104,2,2)),
            ma = list(matrix(-4:-1,2,2)), sigma2 = diag(2))
```

dlmModPoly

 n 次の多項式 DLM を生成する

説明

この関数は n 次の多項式 DLM を生成する。

用法

```
dlmModPoly(order = 2, dV = 1, dW = c(rep(0, order - 1), 1),
           m0 = rep(0, order), C0 = 1e+07 * diag(nrow = order))
```

引数

order	多項式モデルの次数。デフォルト値は確率的な線型トレンドに対応している。
dV	観測雑音の分散
dW	状態雑音の分散行列における対角要素
m0	m_0 であり、状態ベクトルの事前分布に関する期待値
C0	C_0 であり、状態ベクトルの事前分布に関する分散行列

返り値

dlm クラスのオブジェクト (指定された n 次の多項式モデルを表す)

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Giovanni Petris (2010), An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1-16. <http://www.jstatsoft.org/v36/i12/>.
 Petris, Petrone, and Campagnoli, Dynamic Linear Models with R, Springer (2009).
 West and Harrison, Bayesian forecasting and dynamic models (2nd ed.), Springer, 1997.

関連事項

dlmModARMA, dlmModReg, dlmModSeas

例

```
## the default
dlmModPoly()
## random walk plus noise
dlmModPoly(1, dV = .3, dW = .01)
```

dlmModReg 回帰モデルの DLM 表現を生成する

説明

この関数は線型回帰モデルの dlm 表現を生成する。

用法

```
dlmModReg(X, addInt = TRUE, dV = 1, dW = rep(0, NCOL(X) + addInt),
          m0 = rep(0, length(dW)),
          C0 = 1e+07 * diag(nrow = length(dW)))
```

引数

X	計画行列
addInt	論理値: 切片を加えるべきか否か?
dV	観測雑音の分散
dW	状態雑音の分散行列における対角要素
m0	m_0 であり、状態ベクトルの事前分布に関する期待値
C0	C_0 であり、状態ベクトルの事前分布に関する分散行列

詳細

dW を非零のベクトルに設定すると、動的回帰モデルの DLM 表現が得られる。dW のデフォルト値は 0 で、これは標準的な線型回帰に対応している。現在対応しているのは、一変量の回帰のみである。

返り値

dlm クラスのオブジェクト (指定された回帰モデルを表す)

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Giovanni Petris (2010), An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1-16. <http://www.jstatsoft.org/v36/i12/>.
 Petris, Petrone, and Campagnoli, Dynamic Linear Models with R, Springer (2009).
 West and Harrison, Bayesian forecasting and dynamic models (2nd ed.), Springer, 1997.

関連事項

dlmModARMA, dlmModPoly, dlmModSeas

例

```
x <- matrix(runif(6,4,10), nc = 2); x
dlmModReg(x)
dlmModReg(x, addInt = FALSE)
```

dlmModSeas	季節要素に関する DLM を生成する
------------	--------------------

説明

この関数は季節成分に関する DLM 表現を生成する。

用法

```
dlmModSeas(frequency, dV = 1, dW = c(1, rep(0, frequency - 2)),
            m0 = rep(0, frequency - 1),
            C0 = 1e+07 * diag(nrow = frequency - 1))
```

引数

frequency	一周期に相当する時点数
dV	観測雑音の分散
dW	状態雑音の分散行列における対角要素
m0	m_0 であり、状態ベクトルの事前分布に関する期待値
C0	C_0 であり、状態ベクトルの事前分布に関する分散行列

返り値

dlm クラスのオブジェクト (frequency で一周期となる過程の季節要素を表す)

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Giovanni Petris (2010), An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1-16. <http://www.jstatsoft.org/v36/i12/>.
 Petris, Petrone, and Campagnoli, Dynamic Linear Models with R, Springer (2009).
 Harvey, Forecasting, structural time series models and the Kalman filter, Cambridge University Press, 1989.

関連事項

dlmModARMA, dlmModPoly, dlmModReg、季節成分のフーリエ表現に関しては dlmModTrig を参照されたい。

例

```
## seasonal component for quarterly data
dlmModSeas(4, dV = 3.2)
```

dlmModTrig

DLM の周期成分に関するフーリエ表現を生成する

説明

この関数が生成する dlm は、指定された周期成分を表す。

用法

```
dlmModTrig(s, q, om, tau, dV = 1, dW = 0, m0, C0)
```

引数

s	周期（整数の場合）
q	DLM に残す調和項の数
om	周波数
tau	周期（整数でない場合）
dV	観測雑音の分散
dW	状態雑音の分散を表す単一の数値
m0	m_0 であり、状態ベクトルの事前分布に関する期待値
C0	C_0 であり、状態ベクトルの事前分布に関する分散行列

詳細

周期成分は s, om, tau の内のどれか一つだけで指定される。この関数では、s が与えられた場合周期は整数と仮定するが、tau で指定される周期は整数ではないと仮定する。tau の代わりに、周波数 om でも指定ができる。引数 q はモデルに含まれる調和項の数を指定する。tau か om が与えられる場合には、更に q も指定が必要になる。これは、この場合に示唆されるフーリエ表現が、無限に多くの調和項を持つためである。他方 s が与えられる場合には、q のデフォルト値はフーリエ表現が持つ調和項全てとなり、 $\text{floor}(s/2)$ となる。

最終的に得られる dlm の状態分散は、適切な次元の単位行列に dW をかけたものになる。

返り値

dlm クラスのオブジェクト（周期成分を表す）

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Giovanni Petris (2010), An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1-16. <http://www.jstatsoft.org/v36/i12/>.
 Petris, Petrone, and Campagnoli, Dynamic Linear Models with R, Springer (2009).
 West and Harrison, Bayesian forecasting and dynamic models (2nd ed.), Springer (1997).

関連事項

dlmModSeas, dlmModARMA, dlmModPoly, dlmModReg

例

```

dlmModTrig(s = 3)
dlmModTrig(tau = 3, q = 1) # same thing
dlmModTrig(s = 4) # for quarterly data
dlmModTrig(s = 4, q = 1)
dlmModTrig(tau = 4, q = 2) # a bad idea!
m1 <- dlmModTrig(tau = 6.3, q = 2); m1
m2 <- dlmModTrig(om = 2 * pi / 6.3, q = 2)
all.equal(unlist(m1), unlist(m2))

```

dlmRandom

ランダムな DLM

説明

(時不変もしくは時変の) "dlm" クラスのオブジェクトをランダムに生成する。さらに、そのモデルからシミュレートした状態や観測値も一緒に返すことができる。

用法

```
dlmRandom(m, p, nobs = 0, JFF, JV, JGG, JW)
```

引数

m	観測ベクトルの次元
p	状態ベクトルの次元
nobs	モデルからシミュレートする状態や観測値の数
JFF	モデルが時変の FF 要素を持つか否か?
JV	モデルが時変の V 要素を持つか否か?
JGG	モデルが時変の GG 要素を持つか否か?
JW	モデルが時変の W 要素を持つか否か?

詳細

この関数は DLM に関する状態行列、観測行列、分散をランダムに生成する。DLM における状態や観測値の次元には、特定の値が設定できる。状態行列 GG は、厳密に 1 未満の固有値を持つように保証される (これは、DLM が時不変であれば漸近的に定常となることを示唆している)。デフォルトの挙動としては、時不変の DLM を生成する。JFF が TRUE の場合、FF の全要素が nobs 個の観測値に渡って毎時点で変化するようなモデルが生成される。これは、JV, JGG, JW に関しても同様である。

返り値

この関数は以下の成分を持つリストを返す。

mod	"dlm" クラスのオブジェクト
theta	モデルからシミュレートされた状態ベクトルの行列
y	モデルからシミュレートされた観測値の行列

nobs が 0 の場合には、mod 成分だけが返される。

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Anderson and Moore, Optimal filtering, Prentice-Hall (1979)

関連事項

dlm

例

```
dlmRandom(1, 3, 5)
```

dlmSmooth

DLM の平滑化

説明

この関数はカルマン平滑化を用いて、状態ベクトルの平滑化値とその分散共分散行列を計算する。

用法

```
dlmSmooth(y, ...)
## Default S3 method:
dlmSmooth(y, mod, ...)
## S3 method for class dlmFiltered
dlmSmooth(y, ..., debug = FALSE)
```

引数

<code>y</code>	このオブジェクトを用いてメソッドが選択される
<code>...</code>	他のメソッドに対して、もしくは他のメソッドから渡される更なる引数
<code>mod</code>	"dlm" クラスのオブジェクト
<code>debug</code>	debug=FALSE ならより高速な C のコードが使用されるが、そうでなければ計算は全て R で実行される。

詳細

デフォルトの方法では、データベクトル (もしくは行列) `y` とモデル `mod` に関する平滑化分布の平均と分散を返す。

`dlmFiltered` オブジェクト (通常 `dlmFilter` を呼び出して生成される) に基づいて同じ出力を作ることでもでき、この場合 `dlmSmooth.dlmFiltered` が使用される事になる。

ここで行われる計算は、関連する行列の特異値分解 (SVD) に基づく。分散は SVD の形で返される。

返り値

次の要素を持つリスト

<code>s</code>	状態ベクトルに関する平滑化値の時系列 (もしくは行列)。この系列は最初の観測値の一時点前から始まる。
<code>U.S</code>	次の項目を参照
<code>D.S</code>	<code>U.S</code> と組み合わせて、平滑化誤差の分散に関する SVD を構築する。

警告

`mod` における観測分散 v は非特異でなければならない。

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Zhang, Y. and Li, X.R., Fixed-interval smoothing algorithm based on singular value decomposition, *Proceedings of the 1996 IEEE International Conference on Control Applications*.

Giovanni Petris (2010), An R Package for Dynamic Linear Models. *Journal of Statistical Software*, 36(12), 1-16. <http://www.jstatsoft.org/v36/i12/>.

Petris, Petrone, and Campagnoli, *Dynamic Linear Models with R*, Springer (2009).

関連事項

`dlm` オブジェクトの説明に関しては `dlm`、SVD から分散行列を得る方法に関しては `dlmSvd2var`、カルマン・フィルタリングに関しては `dlmFilter`、最尤推定に関しては `dlmMLE`、状態ベクトルの事後分布から標本抽出を行う方法に関しては `dlmBSample` を、それぞれ参照されたい。

例

```
s <- dlmSmooth(Nile, dlmModPoly(1, dV = 15100, dW = 1470))
plot(Nile, type = "o")
lines(dropFirst(s$s), col = "red")

## Multivariate
set.seed(2)
tmp <- dlmRandom(3, 5, 20)
obs <- tmp$y
m <- tmp$mod
rm(tmp)

f <- dlmFilter(obs, m)
s <- dlmSmooth(f)
all.equal(s, dlmSmooth(obs, m))
```

dlmSum 動的線型モデルの外部和

説明

dlmSum は、二つ以上の独立な DLM から唯一の DLM を生成する。%% は dlmSum の別名である。

用法

```
dlmSum(...)
x %% y
```

引数

...	dlm クラスのオブジェクトを任意の数だけ並べたもの、もしくはそれらのリスト
x, y	dlm クラスのオブジェクト

返り値

dlm クラスのオブジェクト（引数の外部和を表す）

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Giovanni Petris (2010), An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1-16. <http://www.jstatsoft.org/v36/i12/>.
 Petris, Petrone, and Campagnoli, Dynamic Linear Models with R, Springer (2009).

例

```
m1 <- dlmModPoly(2)
m2 <- dlmModPoly(1)
dlmSum(m1, m2)
m1 %% m2 # same thing
```

dlmSvd2var 非不定値行列の特異値分解から元の行列を計算する

説明

この関数は、非負定値行列の特異値分解から元の行列を計算する。

用法

```
dlmSvd2var(u, d)
```

引数

u	正方行列、もしくは正方行列のリスト（複数時点でベクトル化される場合）
d	ベクトル、もしくは行列（複数時点でベクトル化される場合）

詳細

非負定値の正方行列 $x (n \times n)$ のSVD は ud^2u' と書ける。ここで、 u は $n \times n$ の直交行列であり、 d は対角行列である。この関数は、単一の行列に関しては単に ud^2u' を返す。引数 d は、 d の対角要素を含むベクトルであることに注意しよう。複数時点でベクトル化されている場合、 u は正方行列のリストとなり、 d は行列となる。複数時点でベクトル化されている場合の返り値は行列のリストとなり、その i 番目の要素は `u[[i]] %*% diag(d[i,]^2) %*% t(u[[i]])` となる。

返り値

この関数が返すのは、SVD から再構築された非負定値の行列、もしくはそのような行列のリストである（詳細は上記を参照）。

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Horn and Johnson, Matrix analysis, Cambridge University Press (1985)

例

```
x <- matrix(rnorm(16),4,4)
x <- crossprod(x)
tmp <- La.svd(x)
all.equal(dlmSvd2var(tmp$u, sqrt(tmp$d)), x)
## Vectorized usage
x <- dlmFilter(Nile, dlmModPoly(1, dV=15099, dW=1469))
x$se <- sqrt(unlist(dlmSvd2var(x$U.C, x$D.C)))
## Level with 50% probability interval
plot(Nile, lty=2)
lines(dropFirst(x$m), col="blue")
lines(dropFirst(x$m - .67*x$se), lty=3, col="blue")
lines(dropFirst(x$m + .67*x$se), lty=3, col="blue")
```

dropFirst

ベクトルや行列の最初の要素を取り除く

説明

ユーティリティ関数 `dropFirst` は、ベクトルや行列の最初の要素を取り除く（引数が時系列オブジェクトの場合は、時系列の属性が適切に保持される）。

用法

```
dropFirst(x)
```

引数

`x` ベクトルもしくは行列

返り値

引数が行列の場合、この関数は `x[-1]` か `x[-1,]` を返す。引数が `ts` クラスのオブジェクトの場合、クラスの `tsp` 属性は適切に保持される。

作者

Giovanni Petris <GPetris@uark.edu>

例

```
(pres <- dropFirst(presidents))
start(presidents)
start(pres)
```

FF

dlm オブジェクトの要素

説明

`dlm` クラスのオブジェクトにおける特定の要素を、取得もしくは設定する関数群

用法

```
## S3 method for class dlm
FF(x)
## S3 replacement method for class dlm
FF(x) <- value
## S3 method for class dlm
V(x)
## S3 replacement method for class dlm
V(x) <- value
## S3 method for class dlm
GG(x)
## S3 replacement method for class dlm
GG(x) <- value
## S3 method for class dlm
W(x)
## S3 replacement method for class dlm
W(x) <- value
## S3 method for class dlm
m0(x)
## S3 replacement method for class dlm
m0(x) <- value
## S3 method for class dlm
C0(x)
## S3 replacement method for class dlm
C0(x) <- value
## S3 method for class dlm
```

```

JFF(x)
## S3 replacement method for class dlm
JFF(x) <- value
## S3 method for class dlm
JV(x)
## S3 replacement method for class dlm
JV(x) <- value
## S3 method for class dlm
JGG(x)
## S3 replacement method for class dlm
JGG(x) <- value
## S3 method for class dlm
JW(x)
## S3 replacement method for class dlm
JW(x) <- value
## S3 method for class dlm
X(x)
## S3 replacement method for class dlm
X(x) <- value

```

引数

<code>x</code>	dlm クラスのオブジェクト
<code>value</code>	数値行列 (<code>m0</code> に関しては数値ベクトル)

詳細

`value` において欠損値や無限は許容されない。`value` の次元は、`x` の特定の要素に関する現在の値と整合しなければならない。

返り値

設定の形態では、更新された dlm オブジェクトとなる。

その他の形態では、`x` の特定の要素となる。

作者

Giovanni Petris <GPetris@uark.edu>

関連事項

dlm

例

```

set.seed(222)
mod <- dlmRandom(5, 6)
all.equal( FF(mod), mod$FF )
all.equal( V(mod), mod$V )
all.equal( GG(mod), mod$GG )
all.equal( W(mod), mod$W )
all.equal( m0(mod), mod$m0 )
all.equal( C0(mod), mod$C0 )
m0(mod)
m0(mod) <- rnorm(6)

```

```

C0(mod)
C0(mod) <- rwishart(10, 6)
### A time-varying model
mod <- dlmModReg(matrix(rnorm(10), 5, 2))
JFF(mod)
X(mod)

```

mcmc

MCMC による出力を分析するためのユーティリティ関数

説明

この関数は、平均、平均に関する標準偏差、入力ベクトル（もしくは行列）に関する部分平均の系列を返す。

用法

```

mcmcMean(x, sd = TRUE)
mcmcMeans(x, sd = TRUE)
mcmcSD(x)
ergMean(x, m = 1)

```

引数

<code>x</code>	マルコフ連鎖モンテカルロシミュレーションの出力を含むベクトル、もしくは行列
<code>sd</code>	論理値: モンテカルロ標準偏差の推定値が報告されるべきか否か?
<code>m</code>	<code>m:NROW(x)</code> における各行において、エルゴード平均が計算される

詳細

通常、引数 `x` はシミュレーションからの出力となる。これが行列の場合、列方向はある目標ベクトルに関する継続的なシミュレーション結果になっていると考えられる。その場合、列毎に平均、標準偏差、エルゴード平均が返される。平均に関する標準偏差は、Sokal の方法を用いて推定される (参考文献を参照)。mcmcMeans は mcmcMean の別名である。

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

P. Green (2001). A Primer on Markov Chain Monte Carlo. In *Complex Stochastic Systems*, (Barndorff-Nielsen, Cox and Klüppelberg, eds.). Chapman and Hall/CRC.

例

```

x <- matrix(rexp(1000), nc=4)
dimnames(x) <- list(NULL, LETTERS[1:NCOL(x)])
mcmcSD(x)
mcmcMean(x)
em <- ergMean(x, m = 51)
plot(ts(em, start=51), xlab="Iteration", main="Ergodic means")

```

NelPlo

Nelson-Plosser マクロ経済時系列

説明

Nelson-Plosser データの一部

用法

```
data(NelPlo)
```

フォーマット

書式は次の通り。

```
mts [1:43, 1:2] -4.39 3.12 1.08 -1.50 3.91 ...
- attr(*, "tsp")= num [1:3] 1946 1988 1
- attr(*, "class")= chr [1:2] "mts" "ts"
- attr(*, "dimnames")=List of 2
..$ : NULL
..$ : chr [1:2] "ip" "stock.prices"
```

詳細

この系列は、1946年から1988年までの工業生産と株価(S&P500)を変換($100 \cdot \text{diff}(\log())$)した値である。

出典

完全なデータセットは、パッケージ `tseries` から取得可能である。

例

```
data(NelPlo)
plot(NelPlo)
```

residuals.dlmFiltered

一期先予測誤差

説明

この関数は、フィルタ化された動的線型モデルに関して一期先予測誤差を計算する。

用法

```
## S3 method for class dlmFiltered
residuals(object, ..., type = c("standardized", "raw"), sd = TRUE)
```

引数

object	"dlmFiltered"クラスのオブジェクト (例えば dlmFilter からの出力)
...	追加引数 (未使用)
type	生成される予測誤差が規格化されるか、もしくはそのままか?
sd	sd = TRUE の場合、標準偏差も返される。

返り値

一期先予測誤差のベクトル (多変量の場合は行列) であり、type = "standardized"であれば規格化される。元の観測ベクトル (行列) における時系列属性は、一期先予測誤差でも保持される。

sd = TRUE なら返り値はリストとなり、要素 res には一期先予測誤差、要素 sd には対応する標準偏差が入る。

注記

引数 object の要素には、データが格納された y が含まれる必要がある。なお dlmFilter を呼び出す際に simplify = TRUE とすると、得られる返り値に要素 y は含まれない。

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Giovanni Petris (2010), An R Package for Dynamic Linear Models. Journal of Statistical Software, 36(12), 1-16. <http://www.jstatsoft.org/v36/i12/>.
Petris, Petrone, and Campagnoli, Dynamic Linear Models with R, Springer (2009).
West and Harrison, Bayesian forecasting and dynamic models (2nd ed.), Springer (1997).

関連事項

dlmFilter

例

```
## diagnostic plots
nileMod <- dlmModPoly(1, dV = 15100, dW = 1468)
nileFilt <- dlmFilter(Nile, nileMod)
res <- residuals(nileFilt, sd=FALSE)
qqnorm(res)
tsdiag(nileFilt)
```

rwishart	ランダムウィシャート行列
----------	--------------

説明

ウィシャート分布から抽出した標本を生成する

用法

```
rwishart(df, p = nrow(SqrtSigma), Sigma, SqrtSigma = diag(p))
```

引数

df	自由度。整数でなければならない。
p	シミュレートする行列の次元
Sigma	ウィシャート分布のパラメータ行列 Sigma
SqrtSigma	ウィシャート分布のパラメータ行列 Sigma の平方根(行列)。Sigma は <code>crossprod(SqrtSigma)</code> と等しくなる必要がある。

詳細

ウィシャート分布は、非負定値対称行列の集合に関する分布である。その密度は次のようになる。

$$p(W) = \frac{c|W|^{(n-p-1)/2}}{|\Sigma|^{n/2}} \exp \left\{ -\frac{1}{2} \text{tr}(\Sigma^{-1}W) \right\}$$

ここで n は自由度のパラメータ `df` であり、 c は規格化定数である。ウィシャート分布の平均は $n\Sigma$ であり、要素の分散は次のようになる。

$$\text{Var}(W_{ij}) = n(\Sigma_{ij}^2 + \Sigma_{ii}\Sigma_{jj})$$

パラメータ行列は正定値対称行列であるべきで、引数 `Sigma` か `SqrtSigma` のいずれかを通じて指定できる。`Sigma` が指定された場合、`SqrtSigma` は無視される。`Sigma` が対称かつ正定値であるかについての確認は行われない。

返り値

この関数は、自由度が `df` でパラメータ行列 `Sigma` か `crossprod(SqrtSigma)` のウィシャート分布から抽出された標本を一つ返す。

警告

この関数は自由度が整数の場合のみ機能する。

注記

メーリングリスト S-news に投稿された、B.Venables による提案に基づく。

作者

Giovanni Petris <GPetris@uark.edu>

参考文献

Press (1982). Applied multivariate analysis.

例

```
rwishart(25, p = 3)
a <- matrix(rnorm(9), 3)
rwishart(30, SqrtSigma = a)
b <- crossprod(a)
rwishart(30, Sigma = b)
```

USecon

米国のマクロ経済時系列

説明

米国のマクロ経済データ

用法

```
data(USecon)
```

フォーマット

書式は次の通り。

```
mts [1:40, 1:2] 0.1364 0.0778 -0.3117 -0.5478 -1.2636 ...
- attr(*, "dimnames")=List of 2
..$: NULL
..$: chr [1:2] "M1" "GNP"
- attr(*, "tsp")= num [1:3] 1978 1988 4
- attr(*, "class")= chr [1:2] "mts" "ts"
```

詳細

この系列は、1978 年から 1987 年までの米国における季調済みの実質通貨供給量'M1' と実質 GNP を変換 ($100 * \text{diff}(\log())$) した値である。

出典

完全なデータセットは、パッケージ `tseries` から取得可能である。

例

```
data(USecon)
plot(USecon)
```

索引

* トピック array		
d1mSvd2var	25	
* トピック datagen		
d1mRandom	22	
* トピック datasets		
NelPlo	30	
USecon	33	
* トピック distribution		
arms	3	
rwishart	32	
* トピック misc		
arms	3	
ARtransPars	5	
bdiag	6	
convex.bounds	7	
d1m	7	
d1mBSample	9	
d1mFilter	10	
d1mForecast	11	
d1mGibbsDIG	13	
d1mLL	14	
d1mMLE	15	
d1mModARMA	17	
d1mModPoly	18	
d1mModReg	19	
d1mModSeas	20	
d1mModTrig	21	
d1mRandom	22	
d1mSmooth	23	
d1mSum	25	
d1mSvd2var	25	
dropFirst	26	
mcmc	29	
residuals.d1mFiltered	30	
* トピック multivariate		
arms	3	
* トピック smooth		
d1mSmooth	23	
* トピック ts		
d1mFilter	10	
d1mSmooth	23	
d1mSum	25	
dropFirst	26	
		FF
		%+%(d1mSum)
		arms
		ARtransPars
		as.d1m(d1m)
		bdiag
		C0 (FF)
		C0<-(FF)
		convex.bounds
		d1m
		d1mBSample
		d1mFilter
		d1mForecast
		d1mGibbsDIG
		d1mLL
		d1mMLE
		d1mModARMA
		d1mModPoly
		d1mModReg
		d1mModSeas
		d1mModTrig
		d1mRandom
		d1mSmooth
		d1mSum
		d1mSvd2var
		dropFirst
		ergMean (mcmc)
		FF
		FF<-(FF)
		GG (FF)
		GG<-(FF)
		is.d1m(d1m)
		JFF (FF)
		JFF<-(FF)
		JGG (FF)
		JGG<-(FF)

JV (<i>FF</i>)	27
JV<- (<i>FF</i>)	27
JW (<i>FF</i>)	27
JW<- (<i>FF</i>)	27
m0 (<i>FF</i>)	27
m0<- (<i>FF</i>)	27
mcmc	29
mcmcMean (<i>mcmc</i>)	29
mcmcMeans (<i>mcmc</i>)	29
mcmcSD (<i>mcmc</i>)	29
NelPlo	30
residuals.dlmFiltered	30
rwishart	32
USecon	33
V (<i>FF</i>)	27
V<- (<i>FF</i>)	27
W (<i>FF</i>)	27
W<- (<i>FF</i>)	27
X (<i>FF</i>)	27
X<- (<i>FF</i>)	27