Package 'Stem'

January 2, 2012

Type Package

Title Spatio-temporal models in R

Version 1.0

Date 2009-01-27

Author Michela Cameletti

Maintainer Michela Cameletti <michela.cameletti@unibg.it>

Depends R (>= 2.7.0), mvtnorm, MASS

Description Estimation of the parameters of a spatio-temporal model using the EM algorithm, estimation of the parameter standard errors using a spatio-temporal parametric bootstrap, spatial mapping.

LazyLoad yes

Repository CRAN

Date/Publication 2009-01-30 10:21:38

R topics documented:

`Stem-package` *Analysis of spatio-temporal hierarchical models*

Description

This package focuses on spatio-temporal hierarchical models. The package includes functions for maximum likelihood estimation (based on Kalman filtering and EM algorithm), for computing the parameter standard errors (using a parametric spatio-temporal bootstrap) and for spatial mapping.

Details

Package:	Stem
Type:	Package
Version:	1.0
Date:	2009-01-27
License:	GPL (>= 2)
LazyLoad:	yes

Author(s)

Michela Cameletti < `michela.cameletti@unibg.it` >

References

Amisigo, B.A., Van De Giesen, N.C. (2005) *Using a spatio-temporal dynamic state-space model with the EM algorithm to patch gaps in daily riverflow series*. Hydrology and Earth System Sciences 9, 209–224.

Fasso', A., Cameletti, M., Nicolis, O. (2007) *Air quality monitoring using heterogeneous networks*. Environmetrics 18, 245–264.

Fasso', A., Cameletti, M. (2007) *A general spatio-temporal model for environmental data*. Tech.rep. n.27 *Graspa* - The Italian Group of Environmental Statistics - http://www.graspa.org.

Fasso', A., Cameletti, M. (2009) *A unified statistical approach for simulation, modelling, analysis and mapping of environmental data*. Accepted for publication by *Simulation: transaction of the Society for Modeling and Simulation International*.

Mc Lachlan, G.J., Krishnan, T. (1997) *The EM Algorithm and Extensions*. Wiley, New York.

Shumway, R.H., Stoffer, D.S. (2006) *Time Series Analysis and Its Applications: with R Examples*. Springer, New York.

Xu, K., Wikle, C.K. (2007) *Estimation of parameterized spatio-temporal dynamic models*. Journal of Statistical Inference and Planning 137, 567–588.

pm10 *Realistic data which illustrate the usage of the package Stem*

Description

This simple data set is a list of three objects and refers to 22 spatial locations and 366 time points.

Usage

```
data(pm10)
```

Format

A list of with three objects with the following components:

`coords` the coordinates of the 22 spatial locations.

`covariates` it is a 8052 by 3 matrix referring to the following covariates: *intercept, emissions (g/s) and altitude (km)*. The first 366 rows refer to the first spatial location, the rows from 367 to 732 refer to the second spatial location and so on.

`z` it is a 366 by 22 observation matrix referring to *PM10 concentration measurements* (log scale).

Author(s)

Michela Cameletti < `michela.cameletti@unibg.it` >

References

Fasso', A., Cameletti, M. (2007) *A general spatio-temporal model for environmental data*. Tech.rep. n.27 *Graspa* - The Italian Group of Environmental Statistics - http://www.graspa.org.

Examples

```
data(pm10)
names(pm10)

#plot the coordinates
dim(pm10$coords)
plot(pm10$coords[,1],pm10$coords[,2],xlab=colnames(pm10$coords)[1],
ylab=colnames(pm10$coords)[2])

#plot the data
dim(pm10$z)

#summary by station
apply(pm10$z,2,summary)

#plot the time series for station n.22
plot(pm10$z[,22],t="l",xlab="Days",ylab="PM10 concentrations (log)")
```

```
#plot the station altitude
plot(pm10$covariates[,3],ylab=colnames(pm10$covariates)[3],xaxt="n",xlab="")
positions = seq(1,8052,366)+366/2
axis(1, at=positions, labels=rownames(pm10$coords), las=2)
```

`Stem.Bootstrap` *Parametric bootstrap*

Description

This functions performs the spatio-temporal parametric bootstrap for computing the parameter standard errors.

Usage

```
Stem.Bootstrap(StemModel, B)
```

Arguments

`StemModel`	an object of class "Stem.Model" given as output by the `Stem.Estimation` function.
`B`	number of bootstrap iterations.

Details

The spatio-temporal bootstrap is used for parameter uncertainty assessment. The resampling scheme is based on the estimated model: each bootstrap sample is drawn directly from the Gaussian distributions which define the model with the parameter vector replaced by the corresponding ML estimates. For each of the B bootstrap samples, the ML estimates are computed (using the procedure of `Stem.Estimation` function). Then the B bootstrap replications are returned in a list.

Value

The function returns a list of elements called "boot.output". **Each** element of the list is an object of class "Stem.Model" and so it is composed by the following elements:

`skeleton`	a list with components `phi`, `p`, `K` where the `phi` vector is given by the `StemModel$estimates$phi.hat` vector.
`data`	a list with components `z`, `coordinates`, `covariates`, `r`, `n` and `d` where the `z` matrix is given by the simulated data matrix.
`estimates`	A list of four objects: `phi.hat`, `y.smoothed`, `loglik`, `convergence.par` as the output of the `Stem.Estimation` function.

Author(s)

Michela Cameletti < `michela.cameletti@unibg.it` >

References

Amisigo, B.A., Van De Giesen, N.C. (2005) *Using a spatio-temporal dynamic state-space model with the EM algorithm to patch gaps in daily riverflow series*. Hydrology and Earth System Sciences 9, 209–224.

Fasso', A., Cameletti, M., Nicolis, O. (2007) *Air quality monitoring using heterogeneous networks*. Environmetrics 18, 245–264.

Fasso', A., Cameletti, M. (2007) *A general spatio-temporal model for environmental data*. Tech.rep. n.27 *Graspa* - The Italian Group of Environmental Statistics - http://www.graspa.org.

Fasso', A., Cameletti, M. (2009) *A unified statistical approach for simulation, modelling, analysis and mapping of environmental data*. Accepted for publication by *Simulation: transaction of the Society for Modeling and Simulation International*.

Xu, K., Wikle, C.K. (2007) *Estimation of parameterized spatio-temporal dynamic models*. Journal of Statistical Inference and Planning 137, 567–588.

Mc Lachlan, G.J., Krishnan, T. (1997) *The EM Algorithm and Extensions*. Wiley, New York.

Shumway, R.H., Stoffer, D.S. (2006) *Time Series Analysis and Its Applications: with R Examples*. Springer, New York.

See Also

See Also `pm10`, `Stem.Model` and `Stem.Estimation`

Examples

```
#load the data
data(pm10)

#extract the data
coordinates <- pm10$coords
covariates <- pm10$covariates
z <- pm10$z

#build the parameter list
#(the phi list is used for the algorithm starting values)
phi <- list(beta=matrix(c(3.65,0.046,-0.904),3,1),
sigma2eps=0.1,
sigma2omega=0.2,
theta=0.01,
G=matrix(0.77,1,1),
Sigmaeta=matrix(0.3,1,1),
m0=as.matrix(0),
C0=as.matrix(1))

K <- matrix(1,ncol(z),1)

mod1 <- Stem.Model(z=z,covariates=covariates,
coordinates=coordinates,phi=phi,K=K)
class(mod1)
is.Stem.Model(mod1)
```

```
#mod1 is given as output by the Stem.Model function
mod1.est <- Stem.Estimation(mod1)

#it is computer intensive
mod1.boot <- Stem.Bootstrap(StemModel=mod1.est, B=3)
names(mod1.boot)
#the first element of the output list
names(mod1.boot$boot.output[[1]])


#check if there is no convergence for some bootstrap iteration
B <- length(mod1.boot$boot.output)
n.null <- sum (unlist (lapply(mod1.boot$boot.output,
function(x) length(x$StemModel)==1)) )
pos.null <- which((unlist (lapply(mod1.boot$boot.output,
function(x) length(x$StemModel)==1)) ))
cat("------------B non null =", B - n.null,"\n")
if(length(pos.null)>0) boot1.mod = mod1.boot[-pos.null]

#put the bootstrap output in a matrix
npar <- length(unlist((mod1.boot$boot.output[[1]]$estimates$phi.hat)))-1
boot.estimates <- matrix(NA, nrow = (B - n.null), ncol = npar)

for(b in 1:(B - n.null)) {
phi.estimated <- mod1.boot$boot.output[[b]]$estimates$phi.hat
boot.estimates[b,] <-  c(phi.estimated$beta,
phi.estimated$sigma2eps,
       phi.estimated$sigma2omega,
phi.estimated$theta,
       phi.estimated$G,
phi.estimated$Sigmaeta,
phi.estimated$m0)
}

#compute the parameter standard errors
se <- sqrt(diag(var(na.omit(boot.estimates))))

#create a summary table with Estimates, Standard Errors (SE) and T-statistics.
phi.hat <- mod1.est$estimates$phi.hat
MLE <- c(phi.hat$beta, phi.hat$sigma2eps, phi.hat$sigma2omega,
phi.hat$theta, phi.hat$G, phi.hat$Sigmaeta,phi.hat$m0)
output1 <- cbind(MLE, se, MLE/se)
colnames(output1)<- c("Estimate", "SE", "T-stat.")
output1

#compute the 95% confidence intervals
IC <- matrix(NA,nrow=npar,ncol=2)
for(i in 1 : npar) {
      IC[i,] <- c(quantile(boot.estimates[,i],0.025),
quantile(boot.estimates[,i],0.975))
}
```

```
#create a summary table with Estimates, Standard Errors (SE)
#and T-statistics and confidence intervals.
output2 <- cbind(output1,IC)
colnames(output2) <- c("Estimate", "SE", "T-stat.", "IC_inf", "IC_sup")
output2
```

Stem.Estimation *ML Estimation*

Description

The function computes the maximum likelihood estimates of the unknown parameters of a hierarchical spatio-temporal model of class "Stem.Model". The estimates are obtained using Kalman filtering and EM algorithm.

Usage

```
Stem.Estimation(StemModel, precision = 0.01, max.iter = 50,
flag.Gdiag = TRUE, flag.Sigmaetadiag = TRUE, cov.spat = Sigmastar.exp)
```

Arguments

StemModel	an object of class "Stem.Model" given as output by the Stem.Model function.
precision	a small positive number used for the algorithm convergence. Default is equal to 0.01. See DETAILS below.
max.iter	maximum number of iterations for the EM algorithm. Default is equal to 50.
flag.Gdiag	logical, indicating whether the transition matrix G is diagonal.
flag.Sigmaetadiag	logical, indicating whether the variance-covariance matrix of the state equation Σ_η is diagonal.
cov.spat	type of spatial covariance function. For the moment only the *exponential* function is implemented.

Details

This function estimates the vector parameter phi of the hierarchical spatio-temporal model of class "Stem.Model" using Kalman filtering and EM algorithm. The algorithm details and formulas are given in Fasso' and Cameletti (2007, 2009). Note that some parameters (beta, sigma2omega, G, Sigmaeta and m0) are updated using closed form solutions while theta and sigma2epsilon using the Newton-Raphson algorithm.

For initializing the algorithm the values contained in StemModel$skeleton$phi are used as initial values. The algorithm converges when the following convergence criteria (named in the output as conv.par and conv.log respectively) are jointly met

$$\frac{\left\|\phi^{(i+1)} - \phi^{(i)}\right\|}{\left\|\phi^{(i)}\right\|} < \pi$$

$$\frac{\left\| \log L\left(\phi^{(i+1)}\right) - \log L\left(\phi^{(i)}\right) \right\|}{\left\| \log L\left(\phi^{(i)}\right) \right\|} < \pi$$

where π is given by the `precision` option and i is the number of iteration. The use of these relative criteria instead of some other absolute ones makes it possible to correct for the different parameter scales.

Value

The function returns an object of class "Stem.Model" which is a list given by:

`skeleton` As the `skeleton` component of the `StemModel` object given in input.

`data` As the `data` component of the `StemModel` object given in input.

`estimates` A list of four objects: `phi.hat`, `y.smoothed`, `loglik`, `convergence.par` here described.

`phi.hat` is a list with the parameter ML estimates (`sigma2omega`, `beta`, `G`, `Sigmaeta`, `m0`, `C0`, `theta`, `sigma2eps`). `y.smoothed` is a `ts` object (n by p) which is the output of the Kalman filtering procedure. `loglik` is the log-likehood value. `convergence.par` is a list of 4 objects with some information about the convergence of the algorithm: `conv.log` and `conv.par` are logical values for the two convergence criteria described above; `iterEM` is the number of iterations for the EM algorithm and `iterNR` is the number of Newton-Raphson iterations for each EM algorithm iteration.

Author(s)

Michela Cameletti < `michela.cameletti@unibg.it` >

References

Amisigo, B.A., Van De Giesen, N.C. (2005) *Using a spatio-temporal dynamic state-space model with the EM algorithm to patch gaps in daily riverflow series*. Hydrology and Earth System Sciences 9, 209–224.

Fasso, A., Cameletti, M., Nicolis, O. (2007) *Air quality monitoring using heterogeneous networks*. Environmetrics 18, 245–264.

Fasso', A., Cameletti, M. (2007) *A general spatio-temporal model for environmental data*. Tech.rep. n.27 *Graspa* - The Italian Group of Environmental Statistics - http://www.graspa.org.

Fasso', A., Cameletti, M. (2009) *A unified statistical approach for simulation, modelling, analysis and mapping of environmental data*. Accepted for publication by *Simulation: transaction of the Society for Modeling and Simulation International*.

Mc Lachlan, G.J., Krishnan, T. (1997) *The EM Algorithm and Extensions*. Wiley, New York.

Shumway, R.H., Stoffer, D.S. (2006) *Time Series Analysis and Its Applications: with R Examples*. Springer, New York.

Xu, K., Wikle, C.K. (2007) *Estimation of parameterized spatio-temporal dynamic models*. Journal of Statistical Inference and Planning 137, 567–588.

See Also

See Also Stem.Model and pm10

Examples

```
#load the data
data(pm10)

#extract the data
coordinates <- pm10$coords
covariates <- pm10$covariates
z <- pm10$z

#build the parameter list
#(the phi list is used for the algorithm starting values)
phi <- list(beta=matrix(c(3.65,0.046,-0.904),3,1),
sigma2eps=0.1,
sigma2omega=0.2,
theta=0.01,
G=matrix(0.77,1,1),
Sigmaeta=matrix(0.3,1,1),
m0=as.matrix(0),
C0=as.matrix(1))

K <-matrix(1,ncol(z),1)

mod1 <- Stem.Model(z=z,covariates=covariates,
coordinates=coordinates,phi=phi,K=K)
class(mod1)
is.Stem.Model(mod1)

#mod1 is given as output by the Stem.Model function
mod1.est <- Stem.Estimation(mod1)
phi.estimates <- unlist(mod1.est$estimates$phi.hat)
```

Stem.Kriging *Dynamical spatial mapping*

Description

This functions performs spatial prediction in a set of new S spatial locations for a fixed time point.

Usage

```
Stem.Kriging(StemModel, coord.newlocations, covariates.newlocations,
K.newlocations, time.point, cov.spat = Sigmastar.exp)
```

Arguments

- `StemModel` — an object of class "Stem.Model" given as output by the `Stem.Estimation` function.
- `coord.newlocations` — a matrix or a data frame of dimension S by 2.
- `covariates.newlocations` — a matrix of dimension S by r, where r is the number of covariates as in the `StemModel` object.
  It has the same structure of the `StemModel$data$covariates` object in the sense that each station data set is stacked under the others; see the `DETAILS` of the `Stem.Model` function.
- `K.newlocations` — a loading matrix of dimension S by p.
- `time.point` — the time point between 1 and n for which the spatial prediction is performed.
- `cov.spat` — type of spatial covariance function. For the moment only the *exponential* function is implemented.

Details

Given the observation matrix and using the multivariate Normal distribution standard theory, the predictor in the new generic spatial location s_0 at time t is an univariate Gaussian distribution with mean $z(s_0, t)$ and variance $\sigma^2(s_0)$ given by:

$$z(s_0, t) = X(s_0, t)\beta + K(s_0)y_t + \Omega'\Sigma_e^{-1}(z_t - X_t\beta + Ky_t)$$

$$\sigma^2(s_0) = \sigma^2\omega - \Omega'\Sigma_e^{-1}\Omega$$

where Ω is the $d \times 1$ constant in time covariance vector, whose i-th generic element ($i = 1, ..., d$) is $Cov(z(s_i, t), z(s_0, t))$. Moreover, $X(s_0, t)$ is the $1 \times r$ vector of covariates for the new site s_0 and $K(s_0)$ is the $1 \times p$ loading vector. Note that all the parameters in the previous formula are ML estimates and the latent process y_t is the output of the Kalman filtering procedures for each time point t.

Value

The function returns a list which is given by:

- `data.newlocations` — a list of five objects related to the new spatial locations: the coordinates (`coordinates`), the covariates (`covariates`), the `K` matrix, the predictions `z` and the prediction standard errors (`se.pred`).
- `time.point` — the time point for which the spatial prediction is performed.

Author(s)

Michela Cameletti < `michela.cameletti@unibg.it` >

References

Amisigo, B.A., Van De Giesen, N.C. (2005) *Using a spatio-temporal dynamic state-space model with the EM algorithm to patch gaps in daily riverflow series*. Hydrology and Earth System Sciences 9, 209–224.

Fasso', A., Cameletti, M., Nicolis, O. (2007) *Air quality monitoring using heterogeneous networks*. Environmetrics 18, 245–264.

Fasso', A., Cameletti, M. (2007) *A general spatio-temporal model for environmental data*. Tech.rep. n.27 *Graspa* - The Italian Group of Environmental Statistics - http://www.graspa.org.

Fasso', A., Cameletti, M. (2009) *A unified statistical approach for simulation, modelling, analysis and mapping of environmental data*. Accepted for publication by *Simulation: transaction of the Society for Modeling and Simulation International*.

Mc Lachlan, G.J., Krishnan, T. (1997) *The EM Algorithm and Extensions*. Wiley, New York.

Shumway, R.H., Stoffer, D.S. (2006) *Time Series Analysis and Its Applications: with R Examples*. Springer, New York.

Xu, K., Wikle, C.K. (2007) *Estimation of parameterized spatio-temporal dynamic models*. Journal of Statistical Inference and Planning 137, 567–588.

See Also

See Also pm10, Stem.Model and Stem.Estimation

Examples

```
#load the data
data(pm10)

#extract the data
coordinates <- pm10$coords
covariates <- pm10$covariates
z <- pm10$z

#build the parameter list
#(the phi list is used for the algorithm starting values)
phi <- list(beta=matrix(c(3.65,0.046,-0.904),3,1),
sigma2eps=0.1,
sigma2omega=0.2,
theta=0.01,
G=matrix(0.77,1,1),
Sigmaeta=matrix(0.3,1,1),
m0=as.matrix(0),
C0=as.matrix(1))

K<-matrix(1,ncol(z),1)

mod1 <- Stem.Model(z=z,covariates=covariates,coordinates=coordinates,phi=phi,K=K)
class(mod1)
is.Stem.Model(mod1)

#mod1 is given as output by the Stem.Model function
```

```
mod1.est <- Stem.Estimation(mod1)

#coordinates of the 25 new points displaced in a regular grid (S=25)
xxx  <- seq(400,470,length=5)
yyy <- seq(5000,5070,length=5)
coord.new <- expand.grid(x=xxx,y=yyy)

#plot of the spatial locations
plot(pm10$coords[,1],pm10$coords[,2],xlab=colnames(pm10$coords)[1],
ylab=colnames(pm10$coords)[2])
points(coord.new[,1],coord.new[,2],col=2,pch=19)
legend("topleft",col=c(1,2),lty=c(0,0), pch=c(21,19),
legend=c("Original spatial locations","New spatial locations"))

#the covariates matrix for the new 25 spatial locations for the 10th time point
covariates.new <- cbind(rep(1,25),
c(37.98348, 18.14824, 15.32287, 11.00458, 6.67696,
29.120820, 10.487590, 2.401088, 26.112971, 1.683525,
19.211907, 31.363448, 3.629172, 10.352472, 48.289624,
7.199692, 3.524810, 25.546621, 19.598600, 10.521586,
0.004736363, 0.365510044, 0.975484255, 25.523642458, 4.671496566),
c(0.227688, 0.173037, 0.139985, 0.116392, 0.102476,
0.278325, 0.256422, 0.168136, 0.129460, 0.121040,
0.722656, 0.238780, 0.202586, 0.166547, 0.154638,
0.733208, 1.467990, 0.380001, 0.251896, 0.240350,
 2.292299 ,2.275844 ,1.382322, 0.300729, 0.208798))

K.new<-matrix(1,25,1)

#dynamical spatial prediction (10th day)
mod1.pred <-Stem.Kriging(StemModel=mod1.est,coord.newlocations=coord.new,
covariates.newlocations=covariates.new,
K.newlocations<-K.new,time.point=10)

#post-processing: build an image map
image(x=xxx,y=yyy,z=matrix(mod1.pred$data.newlocations$z,
length(xxx),length(yyy)),
xlab=colnames(pm10$coords)[1],ylab=colnames(pm10$coords)[2],
xlim=range(mod1.est$data$coordinates[,1])+5,
ylim=range(mod1.est$data$coordinates[,2]+5))
points(pm10$coords[,1],pm10$coords[,2])
points(coord.new[,1],coord.new[,2],col=2,pch=19)

byline <- min((range(xxx)[2]-range(xxx)[1])/4,(range(yyy)[2]-range(yyy)[1])/4)
abline(v=seq(range(xxx)[1],range(xxx)[2],by=byline),col="grey",lty=2)
abline(h=seq(range(yyy)[1],range(yyy)[2],by=byline),col="grey",lty=2)
```

Stem.Model *Create a Stem Model*

Description

The function `Stem.Model` is used to create an object of class "Stem.Model".

Usage

```
Stem.Model(...)

is.Stem.Model(x)
```

Arguments

`...`	List with named elements: `phi`, `K`, `z`, `coordinates`, `covariates` and, optionally, `p` (default equal to 1). See the model details and notation below.
`x`	an object of class Stem.Model

Details

The hierarchical spatio-temporal model is given by

$$z_t = X_t\beta + Ky_t + e_t, e_t \sim N(0, \Sigma_e)$$

$$y_t = Gy_{t-1} + \eta_t, \eta_t \sim N(0, \Sigma_\eta)$$

for $t = 1, ..., n$. The initialization is given by $y_0 \sim N(m0, C0)$.
Note that z_t has dimension d by 1, where d is the number of spatial locations and y_t has dimension p by 1, where p is the dimension of the latent process. The matrix X_t is the known covariate matrix and has dimension d by r, where r is the number of covariates. Moreover, the d-dimensional square matrix Σ_e is given by $\sigma^2_\epsilon + \sigma^2_\omega$ in the diagonal (for spatial distance h equal to 0), while the off-diagonal entries are given by $\sigma^2_\omega C(h, \theta)$, where $C(h, \theta)$ is the spatial covariance function. Using the default *exponential* spatial covariance function, it is $C(h, \theta) = \exp(-\theta h)$.
So the parameter vector ϕ is composed by β, σ^2_ϵ, σ^2_ω, θ, G, Σ_η and $m0$ ($C0$ is supposed fixed).

The elements required by the function **must** have the following characteristics:

phi is a list composed by: `beta` (matrix $r \times 1$), `sigma2eps` (scalar), `sigma2omega` (scalar), `theta` (scalar), `G` (matrix $p \times p$), `Sigmaeta` (matrix $p \times p$), `m0` (matrix $p \times 1$), `C0` (matrix $p \times p$). Note that these values will be used as the true parameter values in the `Stem.Simulation` function and as initial values for the EM algorithm in the `Stem.Estimation` function.

K loading matrix d by p.

z observation matrix n by d.

coordinates matrix d by 2 with UTMX-UTMY or LAT-LON coordinates.

covariates matrix $(n \times d) \times r$. It is recommended to build the covariate matrix stacking the data by station. This means that you consider the n by r matrices related to each spatial location and stack them until you get a $(n \times d) \times r$ matrix.

Value

The function returns a list which is given by:

`skeleton`	a list with components `phi`, `p`, `K` as given in the input.
`data`	a list with components `z`, `coordinates`, `covariates`, as given in the input, `r`, `n` and `d`.

Warning

No missing values are admitted in the observation matrix `z`, in the covariates matrix `covariates` and in the `coordinates` matrix.

Author(s)

Michela Cameletti < `michela.cameletti@unibg.it` >

References

Amisigo, B.A., Van De Giesen, N.C. (2005) *Using a spatio-temporal dynamic state-space model with the EM algorithm to patch gaps in daily riverflow series*. Hydrology and Earth System Sciences 9, 209–224.

Fasso', A., Cameletti, M., Nicolis, O. (2007) *Air quality monitoring using heterogeneous networks*. Environmetrics 18, 245–264.

Fasso', A., Cameletti, M. (2007) *A general spatio-temporal model for environmental data*. Tech.rep. n.27 *Graspa* - The Italian Group of Environmental Statistics - `http://www.graspa.org`.

Fasso', A., Cameletti, M. (2009) *A unified statistical approach for simulation, modelling, analysis and mapping of environmental data*. Accepted for publication by *Simulation: transaction of the Society for Modeling and Simulation International*.

Mc Lachlan, G.J., Krishnan, T. (1997) *The EM Algorithm and Extensions*. Wiley, New York.

Shumway, R.H., Stoffer, D.S. (2006) *Time Series Analysis and Its Applications: with R Examples*. Springer, New York.

Xu, K., Wikle, C.K. (2007) *Estimation of parameterized spatio-temporal dynamic models*. Journal of Statistical Inference and Planning 137, 567–588.

See Also

`pm10`

Examples

```
#load the data
data(pm10)
names(pm10)

#extract the data
coordinates <- pm10$coords
covariates <- pm10$covariates
z <- pm10$z

#build the parameter list
phi <- list(beta=matrix(c(3.65,0.046,-0.904),3,1),
sigma2eps=0.1,
sigma2omega=0.2,
theta=0.01,
G=matrix(0.77,1,1),
Sigmaeta=matrix(0.3,1,1),
```

```
m0=as.matrix(0),
C0=as.matrix(1))

K <-matrix(1,ncol(z),1)

mod1 <- Stem.Model(z=z,covariates=covariates,
coordinates=coordinates,phi=phi,K=K)

class(mod1)
is.Stem.Model(mod1)
```

`Stem.Simulation` *Simulation of spatio-temporal data*

Description

The function `Stem.Simulation` simulates spatio-temporal data.

Usage

```
Stem.Simulation(StemModel)
```

Arguments

`StemModel`	an object of class "Stem.Model" given as output by the `Stem.Model` function.

Details

Note that the values contained in `StemModel$skeleton$phi` are used as the true values of the parameters.

Value

The functions return a $n \times d$ matrix of data.

Author(s)

Michela Cameletti < `michela.cameletti@unibg.it` >

References

Amisigo, B.A., Van De Giesen, N.C. (2005) *Using a spatio-temporal dynamic state-space model with the EM algorithm to patch gaps in daily riverflow series*. Hydrology and Earth System Sciences 9, 209–224.

Fasso', A., Cameletti, M., Nicolis, O. (2007) *Air quality monitoring using heterogeneous networks*. Environmetrics 18, 245–264.

Fasso', A., Cameletti, M. (2007) *A general spatio-temporal model for environmental data*. Tech.rep. n.27 *Graspa* - The Italian Group of Environmental Statistics - `http://www.graspa.org`.

Fasso', A., Cameletti, M. (2009) *A unified statistical approach for simulation, modelling, analysis and mapping of environmental data*. Accepted for publication by *Simulation: transaction of the Society for Modeling and Simulation International*.

Mc Lachlan, G.J., Krishnan, T. (1997) *The EM Algorithm and Extensions*. Wiley, New York.

Shumway, R.H., Stoffer, D.S. (2006) *Time Series Analysis and Its Applications: with R Examples*. Springer, New York.

Xu, K., Wikle, C.K. (2007) *Estimation of parameterized spatio-temporal dynamic models*. Journal of Statistical Inference and Planning 137, 567–588.

See Also

pm10 and Stem.Model

Examples

```
data(pm10)
names(pm10)

#extract the data
coordinates <- pm10$coords
covariates <- pm10$covariates
z <- pm10$z

#build the parameter list
phi <- list(beta=matrix(c(3.65,0.046,-0.904),3,1),
sigma2eps=0.1,
sigma2omega=0.2,
theta=0.01,
G=matrix(0.77,1,1),
Sigmaeta=matrix(0.3,1,1),
m0=as.matrix(0),
C0=as.matrix(1))

K <-matrix(1,ncol(z),1)

mod1 <- Stem.Model(z=z,covariates=covariates,
coordinates=coordinates,phi=phi,K=K)

class(mod1)
is.Stem.Model(mod1)

simulateddata = Stem.Simulation(mod1)
```

Index