Package 'spm'

September 6, 2015

Type Package

Title Stochastic Process Modeling (SPM)

Version 1.0

Date 2015-06-18

Author I. Y. Zhbannikov

Maintainer Who to complain to <zhbannikov.ilya@gmail.com>

Description Stochastic Process Modeling

License GPL

Imports Rcpp (>= 0.11.1), RcppArmadillo (>= 0.4.200.0)

LinkingTo Rcpp, RcppArmadillo

Depends deSolve,mice,sas7bdat,RcppArmadillo

R topics documented:

`spm-package` *What the package does (short line) ~~ package title ~~*

Description

More about what it does (maybe more than one line) ~~ A concise (1-5 lines) description of the package ~~

Details

Package:	spm
Type:	Package
Version:	1.0
Date:	2015-06-18
License:	What license is it under?

~~ An overview of how to use the package, including the most important functions ~~

Author(s)

Who wrote it

Maintainer: Who to complain to <yourfault@somewhere.net> ~~ The author and/or maintainer of the package ~~

References

~~ Literature or other references for background information ~~

See Also

~~ Optional links to other man pages, e.g. ~~ ~~ <pkg> ~~

Examples

```
~~ simple examples of the most important functions ~~
```

`prepare_data` *Output values include: 1). Database, prepared for (slow) continuous optimization (with integral). 2). Database, prepared for (quick) discrete optimization (which is used for parameter estimations)*

Description

Output values include: 1). Database, prepared for (slow) continuous optimization (with integral). 2). Database, prepared for (quick) discrete optimization (which is used for parameter estimations)

Usage

```
prepare_data(longdat, vitstat, interval = 1, col.status = "IsDead",
  col.id = "ID", col.age = "Age", col.age.event = "LSmort",
  covariates = c("DBP", "BMI", "DBP1", "DBP2", "Weight", "Height"),
  verbose = T)
```

Arguments

longdat	A table with longitude records.
vitstat	A table with vital statistics.
col.status	A name of column containing status variable (0/1 which indicate alive/dead).
col.id	A name of column containing patient ID.
col.age	A name of age column.
col.age.event	- A name of event column.
covariates	A list of covariates.
verbose	A verbosing output indicator, default TRUE.

Value

A list of two elements: first element contains a data table for continuous optimization and second element contains a data table for quick discrete optimization used in estimation of starting point.

Examples

```
library(spm)
#Reading longitude data:
longdat <- read.csv(system.file("data","longdat.csv",package="spm"))
# Prepare data for optimization:
vitstat <- read.csv(system.file("data","vitstat.csv",package="spm"))
# Remove unneeded NAs:
longdat.nonan <- longdat[which(is.na(longdat$Age) == F),]
vitstat.nonan <- vitstat[which(is.na(vitstat$BirthCohort) == F),]
data=prepare_data(longdat=longdat.nonan, vitstat=vitstat.nonan,interval=1, col.status="IsDead", col.id="ID"
# Parameters estimation:
pars=spm(data,k = 1)
pars
```

---

sim — *Multi-dimension simulation function It uses a, f1, Q, f, b, mu0 and theta as input parameters.*

Description

Multi-dimension simulation function It uses a, f1, Q, f, b, mu0 and theta as input parameters.

Usage

```
sim(N = 100, a = -0.05, f1 = 80, Q = 2e-08, f = 80, b = 5,
  mu0 = 1e-05, theta = 0.08, ystart, tstart = 30, tend = 105, dt = 1,
  k = 1)
```

Arguments

N	Number of individuals
a	A k by k matrix, which characterize the rate of the adaptive response.
f1	A particular state, which if a deviation from the normal (or optimal). This is a vector with length of k.
Q	A matrix k by k, which is a non-negative-definite symmetric matrix.
f	A vector-function (with length k) of the normal (or optimal) state.
b	A diffusion coefficient, k by k matrix.
mu0	mortality at start period of time.
theta	A displacement coefficient of the Gompertz function.
ystart	A vector with length equal to number of dimensions used, defines starting values of covariates.
tstart	A number that defines starting time (30 by default).
tend	A number, defines final time (105 by default).
dt	A time step (1 by default).
k	number of dimensions (k = 1 by default).

Value

A table with simulated data.

Examples

```
library(spm)
data <- sim(N=1000, ystart=c(75, 94), k=1)
head(data)
```

---

simdata *Function that simulates data using u, R, epsilon, mu0, b, Q, theta*

Description

Function that simulates data using u, R, epsilon, mu0, b, Q, theta

Usage

```
simdata(N = 100, u = 8, R = 0.95, epsilon = 5, mu0 = 2e-05, b = 10,
  Q = 2e-08, theta = 0.08, tstart = 30, ystart = 80, dt = 1,
  tmax = 105, k = 1)
```

Arguments

N	Number of individuals
u	A drift vector with length of k.
R	A k by k regression matrix.
epsilon	A time-dependent normally distributed random vector (size=k).
mu0	mortality at start period of time.
b	A diffusion coefficient, k by k matrix.
Q	A matrix k by k, which is a non-negative-definite symmetric matrix.
theta	A displacement coefficient of the Gompertz function.
tstart	A number that defines starting time (30 by default).
ystart	A vector with length equal to number of dimensions used, defines starting values of covariates.
dt	A time step (1 by default).
k	Number of dimensions (k = 1 by default).
tend	A number, defines final time (105 by default).

Value

A table with simulated data.

Examples

```
library(spm)
data <- simdata(N=1000, ystart=c(75, 94), k=1)
head(data)
```

---

spm *Stochastic Process Modelling (SPM) A main function that estimates parameters a, f1, Q, f, b, mu0, theta from given dataset.*

Description

Stochastic Process Modelling (SPM) A main function that estimates parameters a, f1, Q, f, b, mu0, theta from given dataset.

Usage

```
spm(dat, k = 2, verbose = F, tol = NULL)
```

Arguments

dat	A dataset.
k	Number of dimensions.
verbose	A verbosing output indicator.
tol	A tolerance threshold for matrix inversion.

Value

A list of (1) Estimated starting point (from quick discrete optimization) and (2) Estimated coefficients.

Examples

```
library(spm)
# Reading longitude data:
longdat <- read.csv(system.file("data","longdat.csv",package="spm"))
# Prepare data for optimization:
vitstat <- read.csv(system.file("data","vitstat.csv",package="spm"))
# Remove unneeded NAs:
longdat.nonan <- longdat[which(is.na(longdat$Age) == F),]
vitstat.nonan <- vitstat[which(is.na(vitstat$BirthCohort) == F),]
data=prepare_data(longdat=longdat.nonan, vitstat=vitstat.nonan,interval=1, col.status="IsDead", col.id="ID"
# Parameters estimation:
pars=spm(data,k = 1)
pars
```

`spm_integral_MD` *Continuous multi-dimensional optimization It is much slower that discrete but more precise and can handle time intervals with different lengths.*

Description

Continuous multi-dimensional optimization It is much slower that discrete but more precise and can handle time intervals with different lengths.

Usage

```
spm_integral_MD(dat, parameters, k, verbose = F)
```

Arguments

`dat`	A data table.
`parameters`	A starting pont (a vector).
`k`	A number of dimensions.
`verbose`	An indicator of verbosing output.
`tol`	A tolerance threshold for matrix inversion.

Value

A list of two elements: (1) parameters a, f1, Q, f, b, mu0, theta; (2) An output from "optim" function used for maximum likelihood estimation.

Examples

```
#'library(spm)
# Reading longitude data:
longdat <- read.csv(system.file("data","longdat.csv",package="spm"))
# Prepare data for optimization:
vitstat <- read.csv(system.file("data","vitstat.csv",package="spm"))
# Remove unneeded NAs:
longdat.nonan <- longdat[which(is.na(longdat$Age) == F),]
vitstat.nonan <- vitstat[which(is.na(vitstat$BirthCohort) == F),]
dat=prepare_data(longdat=longdat.nonan, vitstat=vitstat.nonan,interval=1, col.status="IsDead", col.id="ID",
# Parameters estimation:
dat<-[,1:6]
pars=spm_integral_MD(dat, parameters=c(-0.05, 80, 2e-8, 80, 5, 2e-5, 0.08), k = 1)
pars
```

`spm_quick_MD` *Discrete multi-dimensional optimization It is way much faster that continuous (but less precise) and used mainly in estimation of starting point.*

Description

Discrete multi-dimensional optimization It is way much faster that continuous (but less precise) and used mainly in estimation of starting point.

Usage

```
spm_quick_MD(dat, k = 2, theta_range = seq(0.078, 0.082, by = 1e-04),
  tol = NULL)
```

Arguments

`dat`	A data table.
`k`	A number of dimensions.
`theta_range`	A range of theta parameter (axe displacement of Gompertz function).
`tol`	A tolerance threshold for matrix inversion.

Value

A list of two elements: (1) parameters u, R, b, epsilon, Q, mu0, theta and (2) parameters a, f1, Q, f, b, mu0, theta. Note: b and mu0 from first list are different from b and mu0 from the second list.

Examples

```
#'library(spm)
# Reading longitude data:
longdat <- read.csv(system.file("data","longdat.csv",package="spm"))
# Prepare data for optimization:
vitstat <- read.csv(system.file("data","vitstat.csv",package="spm"))
# Remove unneeded NAs:
longdat.nonan <- longdat[which(is.na(longdat$Age) == F),]
vitstat.nonan <- vitstat[which(is.na(vitstat$BirthCohort) == F),]
```

```
data=prepare_data(longdat=longdat.nonan, vitstat=vitstat.nonan,interval=1, col.status="IsDead", col.id="ID"
# Parameters estimation:
pars=spm_quick_MD(data,k = 1)
pars
```

Index