Package: spMC (via r-universe)

October 5, 2026

Type Package

Title Continuous-Lag Spatial Markov Chains

Version 0.3.16

Date 2026-08-23

Author Luca Sartore [aut, cre] (ORCID = ``0000-0002-0446-1328")

Maintainer Luca Sartore <drwolf85@gmail.com>

Description A set of functions is provided for 1) the stratum lengths analysis along a chosen direction, 2) fast estimation of continuous lag spatial Markov chains model parameters and probability computing (also for large data sets), 3) transition probability maps and transiograms drawing, 4) simulation methods for categorical random fields. More details on the methodology are discussed in Sartore (2013) <doi:10.32614/RJ-2013-022> and Sartore et al. (2016) <doi:10.1016/j.cageo.2016.06.001>.

Depends R (>= 4.5.0), base, methods, datasets, utils, grDevices, graphics, stats

License GPL (>= 2)

LazyLoad yes

NeedsCompilation yes

Repository https://cran.r-universe.dev

Date/Publication 2026-09-05 21:22:17 UTC

RemoteUrl https://github.com/cran/spMC

RemoteRef HEAD

RemoteSha 35f214ab603809e301818b0003057810d34dc1e5

Contents

spMC-package 3
ACM 4
boxplot.lengths 5

contour.pemt . . . 6
density.lengths . . . 8
embed_MC . . . 9
getlen . . . 11
hist.lengths . . . 12
image.multi_tpfit . . . 13
image.pemt . . . 15
is.lengths . . . 16
is.multi_tpfit . . . 17
is.multi_transiogram . . . 18
is.pemt . . . 19
is.tpfit . . . 20
is.transiogram . . . 21
mixplot . . . 22
mlen . . . 23
multi_tpfit . . . 25
multi_tpfit_ils . . . 27
multi_tpfit_me . . . 29
multi_tpfit_ml . . . 31
pemt . . . 33
persp.multi_tpfit . . . 34
persp.pemt . . . 36
plot.density.lengths . . . 37
plot.hist.lengths . . . 39
plot.lengths . . . 40
plot.transiogram . . . 41
predict.multi_tpfit . . . 42
predict.tpfit . . . 44
print.density.lengths . . . 45
print.lengths . . . 46
print.multi_tpfit . . . 47
print.multi_transiogram . . . 48
print.summary.lengths . . . 49
print.tpfit . . . 50
print.transiogram . . . 51
quench . . . 52
setCores . . . 54
sim . . . 55
sim_ck . . . 57
sim_ik . . . 59
sim_mcs . . . 61
sim_path . . . 62
summary.lengths . . . 64
tpfit . . . 65
tpfit_ils . . . 67
tpfit_me . . . 68
tpfit_ml . . . 70
transiogram . . . 72

which_lines 73

Index **75**

`spMC-package` *Continuous Lag Spatial Markov Chains*

Description

The main goal of this package is to provide a set of functions for

1. the stratum lengths analysis along a chosen direction,
2. fast estimation of continuous lag spatial Markov chains model parameters and probability computing (also for large data sets),
3. transition probability maps and transiograms drawing,
4. simulation methods for categorical random fields.

Details

Package:	spMC
Type:	Package
Version:	0.3.16
Date:	2026-08-23
License:	GPL (>= 2)
LazyLoad:	yes

Several functions are available for the stratum lengths analysis, in particular they compute the stratum lengths for each stratum category, they compute the empirical distributions and many other tools for a graphical analysis.

Usually, the basic inputs for the most of the functions are a vector of categorical data and their location coordinates. They are used to estimate empirical transition probabilities (transiogram), to estimate model parameters (tpfit for one-dimensional Markov chains or multi_tpfit for multi-dimensional Markov chains). Once parameters are estimated, it's possible to compute theoretical transition probabilities by the use of the function predict.tpfit for one-dimensional Markov chains and predict.multi_tpfit for multidimensional ones.

The function plot.transiogram allows to plot one-dimensional transiograms, while image.multi_tpfit permit to draw transition probability maps. A powerful tool to explore graphically the anisotropy of such process is given by the functions pemt and image.pemt, which let the user to draw "quasi-empirical" transition probability maps.

Simulation methods are based on Indicator Kriging (sim_ik), Indicator Cokriging (sim_ck), Fixed or Random Path algorithms (sim_path) and Multinomial Categorical Simulation technique (sim_mcs).

Author(s)

Luca Sartore

Maintainer: Luca Sartore <drwolf85@gmail.com>

References

Allard, D., D'Or, D., Froidevaux, R. (2011) An efficient maximum entropy approach for categorical variable prediction. *European Journal of Soil Science*, **62**(3), 381-393.

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Dynkin, E. B. (1961) *Theory of Markov Processes*. Englewood Cliffs, N.J.: Prentice-Hall, Inc.

Higham, N. J. (2008) *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics.

Li, W. (2007) A Fixed-Path Markov Chain Algorithm for Conditional Simulation of Discrete Spatial Variables. *Mathematical Geology*, **39**(2), 159-176.

Li, W. (2007) Markov Chain Random Fields for Estimation of Categorical Variables. *Mathematical Geology*, **39**(June), 321-335.

Li, W. (2007) Transiograms for Characterizing Spatial Variability of Soil Classes. *Soil Science Society of America Journal*, **71**(3), 881-893.

Pickard, D. K. (1980) Unilateral Markov Fields. *Advances in Applied Probability*, **12**(3), 655-671.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

Sartore, L. (2013). spMC: Modelling Spatial Random Fields with Continuous Lag Markov Chains. *The R Journal*, **5**(2), 16-28.

Sartore, L., Fabbri, P. and Gaetan, C. (2016). spMC: an R-package for 3D lithological reconstructions based on spatial Markov chains. *Computers & Geosciences*, **94**(September), 40-47.

Weise, T. (2009) *Global Optimization Algorithms - Theory and Application*. https://www.researchgate.net/publication/200622167_Global_Optimization_Algorithm_Theory_and_Application.

`ACM` *ACM Data*

Description

The data set refers to a sampled area which is located in the province of Venice. Its sample units report the geographical position of the perforation, the depth, the ground permeability and other two categorical variables which denote the soil composition.

Usage

```
data(ACM)
```

Format

A data frame with 2321 observations on the following 6 variables.

`X` a numeric vector (longitude)

`Y` a numeric vector (latitude)

`Z` a numeric vector (depth)

`MAT5` a factor with levels `Clay`, `Gravel`, `Mix of Sand and Clay`, `Mix of Sand and Gravel` and `Sand`

`MAT3` a factor with levels `Clay`, `Gravel` and `Sand`

`PERM` a logical vector (symmetric dichotomous variable)

Source

Fabbri, P. (2010) Professor at the Geosciences Department of the University of Padua.
<paolo.fabbri@unipd.it>

References

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

Examples

```
data(ACM)
str(ACM)
summary(ACM)
```

`boxplot.lengths` *Stratum Lengths Boxplot*

Description

Produce box-and-whisker plots of the stratum lengths.

Usage

```
## S3 method for class 'lengths'
boxplot(x, ..., log = FALSE, zeros.rm = TRUE)
```

Arguments

`x`	an object of the class `lengths`, typically with the output of the function `getlen`.
`...`	other arguments to pass to the function `boxplot`.
`log`	a logical value. If `TRUE`, the logarithm of the stratum lengths will be plotted. It is `FALSE` by default.
`zeros.rm`	a logical value. If `FALSE`, the box-and-whisker will be drawn by including zero values. It is `TRUE` by default.

Details

The box-and-whisker plots give some information about the distribution of the stratum lengths for the observed categories along a given direction.

Value

An image is produced on the current graphics device. The function returns a list with the following components:

`stats`	a matrix containing the values used to plot the box-and-whisker plots.
`n`	a vector with the number of observations for each category.
`conf`	a matrix containing further values to draw the lower and upper extremes of the notch.
`out`	a vectors with the values of the outlier points.
`group`	a vector whose elements indicate to which category the outlier belongs.
`names`	a character vector with the names of each category.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

`plot.lengths`, `boxplot`, `getlen`

Examples

```
data(ACM)
direction <- c(0,0,1)

# Compute the appertaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction)

# Estimate stratum lengths
gl <- getlen(ACM$MAT3, ACM[, 1:3], loc.id, direction)

# Make the boxplot of the object gl

boxplot(gl)
```

---

contour.pemt *Display Contours with Multi-directional Transiograms*

Description

The function draws the 2-D sections contour plots of a multi-directional transiogram computed without any ellipsoidal interpolation and superpose the contour lines of the theoretical transition probabilities.

Usage

```
## S3 method for class 'pemt'
contour(x, nlevels = 10, col = c("black", "blue"), main,
     mar, ask = TRUE, ...)
```

Arguments

x	an object of class pemt.
nlevels	the number of levels to pass to the function contour.
col	a vector of two colors to pass to the function contour. The former color refers to the multi-directional transiogram, while the latter is used to draw contour lines of the theoretical transition probabilities.
main	the main title (on top) whose font and size are fixed.
mar	a scalar or a numerical vector of the form c(bottom, left, top, right) which gives the number of margin lines to be specified on the four sides of image to plot. See par(mar=.).
ask	a logical value; if TRUE, the user is asked for input, before each plot. See par(ask=.).
...	other arguments to pass to the function contour.

Details

A multidimensional transiogram is a diagram which shows the transition probabilities for a single pair of categories. The probability is computed for any lag vector h through

$$\text{expm}(\|h\|R_h),$$

where entries of R_h are not ellipsoidally interpolated, but they are estimated for the direction specified by the vector h.

The exponential matrix is evaluated by the scaling and squaring algorithm.

Value

An image is produced on the current graphics device. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Higham, N. J. (2008) *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

image.pemt, contour, plot.transiogram

Examples

```
data(ACM)

# Compute a 2-D section of a
# multi-directional transiogram
psEmpTr <- pemt(ACM$MAT3, ACM[, 1:3], 2,
                max.dist = c(200, 200, 20),
                which.dire=c(1, 3),
                mle = "avg")

# Contour plots of 2-D sections of
# multi-directional transiograms
contour(psEmpTr, mar = .7)
```

---

density.lengths *Empirical Densities Estimation of Stratum Lengths*

Description

The function estimates the empirical conditional density of the stratum lengths given the category.

Usage

```
## S3 method for class 'lengths'
density(x, ..., log = FALSE, zeros.rm = TRUE)
```

Arguments

x	an object of the class lengths, typically with the output of the function getlen.
...	other arguments to pass to the function density.default.
log	a logical value. If TRUE, the output density will be calculated for the logarithm of the lengths. It is TRUE by default.
zeros.rm	a logical value. If FALSE, the density will be estimated by including zero values. It is TRUE by default.

Details

The function estimates the empirical density of the stratum lengths for each category by the use of the kernel methodology.

Value

An object of class density.lengths is returned. It contains objects of class density, the given direction of the stratum lengths and a logical value which points out if the density is computed for the logarithm of stratum lengths.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Simonoff, J. S. (1996) *Smoothing Methods in Statistics*. Springer-Verlag.

See Also

getlen, density.default, plot.density.lengths, print.density.lengths

Examples

```
data(ACM)
direction <- c(0,0,1)

# Compute the appertaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction)

# Estimate stratum lengths
gl <- getlen(ACM$MAT3, ACM[, 1:3], loc.id, direction)

# Compute the empirical densities of stratum lengths
dgl <- density(gl)
```

---

embed_MC *Transition Probabilities Estimation for Embedded Markov Chain*

Description

The function estimates the embedded transition probabilities matrix for a 1-D spatial embedded Markov chain.

Usage

```
embed_MC(data, coords, loc.id, direction)
```

Arguments

data	a categorical data vector of length n.
coords	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
loc.id	a vector of n values which indicats the directional line of each location. It is usually the output of the function which_lines.
direction	a d-D numerical vector (or versor) which represents the chosen direction.

Details

An embedded Markov chain is probabilistic model which defines the transition probabilities between embedded occurrences.

The resulting matrix is given by normalizing a transition count matrix, which doesn't depend on the length of embedded occurrences. Self-transitions of embedded occurrences are not observable, so diagonal entries are set to be `NA`.

It's also possible to calculate the transition probabilities matrix for several directions in a d-D space through arguments `direction` and `loc.id`. If the user has no previous knowledge about `loc.id`, the function `which_lines` provides a method to compute the right values.

Value

A $K \times K$ transition probability matrix, where K denotes the number of observed categories. Another $K \times K$ matrix with the counts of transitions is attached as an attribute.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Dynkin, E. B. (1961) *Theory of Markov Processes. Englewood Cliffs*, N.J.: Prentice-Hall, Inc.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

`which_lines`, `predict.tpfit`, `predict.multi_tpfit`

Examples

```
data(ACM)
direction <- c(0, 0, 1)

# Compute the appertaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction, pi/8)

# Estimate the embedded transition probabilities
# matrix for the categorical variable MAT5
embed_MC(ACM$MAT5, ACM[, 1:3], loc.id, direction)

# Estimate the embedded transition probabilities
# matrix for the categorical variable MAT3
embed_MC(ACM$MAT3, ACM[, 1:3], loc.id, direction)

# Estimate the embedded transition probabilities
# matrix for the categorical variable PERM
embed_MC(ACM$PERM, ACM[, 1:3], loc.id, direction)
```

getlen *Estimation of Stratum Lengths for Embedded Markov Chain*

Description

The function estimates the stratum lengths for a d-D spatial embedded Markov chain for a specified direction ϕ.

Usage

```
getlen(data, coords, loc.id, direction, zero.allowed = FALSE)
```

Arguments

`data`	a categorical data vector of length n.
`coords`	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
`loc.id`	a vector of n values which indicates the directional line of each location. It is usually the output of the function `which_lines`.
`direction`	a d-D numerical vector (or versor) which represents the chosen direction.
`zero.allowed`	a logical value which allows to return zero stratum lengths. It is `FALSE` by default.

Details

Stratum lengths are the lengths occupied by the same k-th category along lines in the direction ϕ.

Value

A list containing the following components:

`length`	a numerical vector with the stratum lengths along the given direction.
`categories`	a vector with the stratum categories.
`maxcens`	a vector with the maxima estimated censored lengths for each stratum.
`directions`	a d-D numerical vector which represents the chosen direction.
`zeros`	a logical values which denotes the possible presence of zero lengths.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

mlen, which_lines

Examples

```
data(ACM)
direction <- c(0,0,1)

# Compute the appertaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction)

# Estimate stratum lengths
gl <- getlen(ACM$MAT5, ACM[, 1:3], loc.id, direction)
```

---

hist.lengths *Histograms of Stratum Lengths for Each Observed Category*

Description

The function compute the histograms of the stratum lengths for each category. If `plot = TRUE`, the resulting object of class `hist.lengths` is plotted before it is returned.

Usage

```
## S3 method for class 'lengths'
hist(x, ..., log = FALSE, zeros.rm = TRUE)
```

Arguments

`x`	an object of the class `lengths`, typically with the output of the function `getlen`.
`...`	further arguments to pass to the function `hist`.
`log`	a logical value. If `TRUE`, histograms will be calculated for the logarithm of the lengths. It is `FALSE` by default.
`zeros.rm`	a logical value. If `FALSE`, histograms will be computed by including zero values. It is `TRUE` by default.

Value

If `plot = TRUE`, an image is produced on the current graphics device. The function returns an object of class `hist.lengths`. It contains class `histogram` objects, the given direction of the stratum lengths and a logical value which points out if histograms are computed for the logarithm of stratum lengths.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

getlen, hist, density.lengths, plot.density.lengths

Examples

```
data(ACM)
direction <- c(0,0,1)

# Compute the appertaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction)

# Estimate stratum lengths
gl <- getlen(ACM$MAT3, ACM[, 1:3], loc.id, direction)

# Plot the histograms
hist(gl)
```

---

image.multi_tpfit *Images with Multidimensional Transiograms*

Description

The function plots 2-D sections of a predicted multidimensional transiograms computed through ellipsoidal interpolation.

Usage

```
## S3 method for class 'multi_tpfit'
image(x, mpoints, which.dire, max.dist, main,
      mar, ask = TRUE, ..., nlevels = 10, contour = TRUE)
```

Arguments

x	an object of the class multi_tpfit, typically with the output of the function multi_tpfit.
mpoints	the number of points per axes. It controls the accuracy of images to plot.
which.dire	a vector with two chosen axial directions. If omitted, all 2-D sections are plotted.
max.dist	a scalar or a vector of maximum length for the chosen axial directions.
main	the main title (on top) whose font and size are fixed.
mar	a scalar or a numerical vector of the form c(bottom, left, top, right) which gives the number of margin lines to be specified on the four sides of image to plot. See par(mar=.).
ask	a logical value; if TRUE, the user is asked for input, before each plot. See par(ask=.).
...	other arguments to pass to the function image.
nlevels	the number of levels to pass to the function contour.
contour	logical. If TRUE, the function contour is used to draw contour lines over the image. Defaults to TRUE.

Details

A multidimensional transiogram is a diagram which shows the transition probabilities for a single pair of categories. It is computed for any lag vector h through

$$\text{expm}(\|h\|R),$$

where entries of R are ellipsoidally interpolated (see multi_tpfit).

The exponential matrix is evaluated by the scaling and squaring algorithm.

Value

An image is produced on the current graphics device. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Higham, N. J. (2008) *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

multi_tpfit, pemt, image.pemt, image, plot.transiogram

Examples

```
data(ACM)

# Estimate model parameter
x <- multi_tpfit(ACM$MAT5, ACM[, 1:3])

# Set short names for categories 3 and 4
names(x$prop)[3:4] <- c("Clay and Sand", "Gravel and Sand")

# Plot 2-D theoretical sections of
# a multidimensional transiogram
image(x, 40, max.dist=c(200,200,20), which.dire=2:3,
    mar = .7, col=rev(heat.colors(500)),
    breaks=0:500/500, nlevels = 5)
```

image.pemt *Images with Multi-directional Transiograms*

Description

The function plots 2-D sections of a multidirectional transiogram computed without any ellipsoidal interpolation.

Usage

```
## S3 method for class 'pemt'
image(x, main, mar, ask = TRUE, ...,
      nlevels = 10, contour = TRUE)
```

Arguments

x	an object of class pemt.
main	the main title (on top) whose font and size are fixed.
mar	a scalar or a numerical vector of the form c(bottom, left, top, right) which gives the number of margin lines to be specified on the four sides of image to plot. See par(mar=.).
ask	a logical value; if TRUE, the user is asked for input, before each plot. See par(ask=.).
...	other arguments to pass to the function image.
nlevels	the number of levels to pass to the function contour.
contour	logical. If TRUE, the function contour is used to draw contour lines over images. Defaults to TRUE.

Details

A multidimensional transiogram is a diagram which shows the transition probabilities for a single pair of categories. The probability is computed for any lag vector h through

$$\text{expm}(\|h\|R_h),$$

where entries of R_h are not ellipsoidally interpolated, but they are estimated for the direction specified by the vector h.

The exponential matrix is evaluated by the scaling and squaring algorithm.

Value

An image is produced on the current graphics device. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Higham, N. J. (2008) *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

`image.multi_tpfit`, `image`, `plot.transiogram`

Examples

```
data(ACM)

# Compute a 2-D section of a
# multi-directional transiogram
psEmpTr <- pemt(ACM$MAT3, ACM[, 1:3], 2,
                max.dist = c(200, 200, 20),
                which.dire=c(1, 3),
                mle = "mlk")

# Plot 2-D sections of
# a multi-directional transiogram
image(psEmpTr, col = rev(heat.colors(500)),
      breaks = 0:500 / 500, mar = .7,
      contour = FALSE)
```

`is.lengths` *Object test for lengths class*

Description

Function to test if an object is of the class `lengths`.

Usage

```
is.lengths(object)
```

Arguments

`object`	object to be tested.

Details

The function returns `TRUE` if and only if its argument is a `lengths` object.

Value

A logical value.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

getlen

Examples

```
data(ACM)
direction <- c(0,0,1)

# Compute the appertaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction)

# Estimate stratum lengths
gl <- getlen(ACM$MAT3, ACM[, 1:3], loc.id, direction)

# Test the object gl
is.lengths(gl)
```

---

is.multi_tpfit *Object test for multi_tpfit class*

Description

Function to test if an object is of the class `multi_tpfit`.

Usage

```
is.multi_tpfit(object)
```

Arguments

`object`	object to be tested.

Details

The function returns `TRUE` if and only if its argument is a `multi_tpfit` object.

Value

A logical value.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

multi_tpfit

Examples

```
data(ACM)

# Estimate the parameters of a
# multidimensional MC models
MoPa <- multi_tpfit(ACM$MAT5, ACM[, 1:3])

# Test the object MoPa
is.multi_tpfit(MoPa)
```

---

is.multi_transiogram *Object test for multi_transiogram class*

Description

Function to test if an object is of the class multi_transiogram.

Usage

```
is.multi_transiogram(object)
```

Arguments

object	object to be tested.

Details

The function returns TRUE if and only if its argument is a multi_transiogram object.

Value

A logical value.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

predict.multi_tpfit

Examples

```
data(ACM)

# Estimate the parameters of a
# multidimensional MC model
RTm <- multi_tpfit(ACM$MAT3, ACM[, 1:3])

# Generate the matrix of
# multidimensional lags
lags <- expand.grid(X=-1:1, Y=-1:1, Z=-1:1)
lags <- as.matrix(lags)

# Compute transition probabilities
# from the multidimensional MC model
TrPr <- predict(RTm, lags)

# Test the object TrPr
is.multi_transiogram(TrPr)
```

`is.pemt` *Images with Multi-direct'ional Transiograms*

Description

The function plots 2-D sections of a multi-directional transiogram computed without any ellipsoidal interpolation.

Usage

```
is.pemt(object)
```

Arguments

`object`	object to be tested.

Details

The function returns `TRUE` if and only if its argument is a `pemt` object.

Value

A logical value.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

`pemt`, `image.pemt`

Examples

```
data(ACM)

# Compute a 2-D section of a
# multi-directional transiogram
psEmpTr <- pemt(ACM$MAT3, ACM[, 1:3], 2,
                max.dist = c(20, 10, 5),
                which.dire=c(1, 3),
                mle = TRUE)

# Test the object psEmpTr
is.pemt(psEmpTr)
```

---

is.tpfit *Object test for tpfit class*

Description

Function to test if an object is of the class `tpfit`.

Usage

```
is.tpfit(object)
```

Arguments

`object`	object to be tested.

Details

The function returns `TRUE` if and only if its argument is a `tpfit` object.

Value

A logical value.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

`tpfit`

Examples

```
data(ACM)

# Estimate the parameters of a
# one-dimensional MC model
MoPa <- tpfit(ACM$MAT5, ACM[, 1:3], c(0, 0, 1))

# Test the object MoPa
is.tpfit(MoPa)
```

---

is.transiogram *Object test for transiogram class*

Description

Function to test if an object is of the class `transiogram`.

Usage

```
is.transiogram(object)
```

Arguments

`object`	object to be tested.

Details

The function returns `TRUE` if and only if its argument is a `transiogram` object.

Value

A logical value.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

`transiogram`, `predict.tpfit`

Examples

```
data(ACM)

# Estimate the parameters of a
# one-dimensional MC model
RTm <- tpfit(ACM$MAT5, ACM[, 1:3], c(0, 0, 1))

# Compute theoretical transition probabilities
# from the one-dimensional MC model
TTPr <- predict(RTm, lags = 0:2/2)

# Compute empirical transition probabilities
ETPr <- transiogram(ACM$MAT5, ACM[, 1:3], c(0, 0, 1), 200, 20)

# Test the objects TTPr and ETPr
is.transiogram(TTPr)
is.transiogram(ETPr)
```

---

mixplot *Plot of Multiple One-dimensional Transiograms*

Description

The function makes a graphical representation of transition probabilities by the use of multiple transiograms.

Usage

```
mixplot(x, main, legend = TRUE, ...)
```

Arguments

x	a list object whose elements are of the class transiogram (typically with the output of the function transiogram or predict.tpfit).
main	the main title (on top) whose font and size are fixed.
legend	a logical value for printing the legend in the graphic. It is TRUE by default.
...	other arguments to pass to the function plot.

Details

Transiogram is a diagram which is drawn for a single pair of categories in the direction ϕ. It shows the transition probabilities in the y-axis for some specific lags in the x-axis.

This function permits a graphical approach to compare theoretical vs. empirical transition probabilities for multiple directions.

Value

An image is produced on the current graphics device. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Li, W. (2007) Transiograms for Characterizing Spatial Variability of Soil Classes. *Soil Science Society of America Journal*, **71**(3), 881-893.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

transiogram, tpfit, predict.tpfit, plot.transiogram, image.multi_tpfit, plot

Examples

```
data(ACM)

# Estimate empirical transition
# probabilities by points
ETr <- transiogram(ACM$MAT3, ACM[, 1:3], c(0, 0, 1), 100)

# Estimate the transition rate matrix
RTm <- tpfit(ACM$MAT3, ACM[, 1:3], c(0, 0, 1))

# Compute transition probabilities
# from the one-dimensional MC model
TPr <- predict(RTm, lags = ETr$lags)

# Plot empirical vs. theoretical transition probabilities
mixplot(list(ETr, TPr), type = c("p", "l"), pch = "+", col = c(3, 1))
```

---

mlen *Mean Length Estimation for Embedded Markov Chain*

Description

The function estimates the mean length for a d-D spatial embedded Markov chain for a specified direction ϕ.

Usage

```
mlen(data, coords, loc.id, direction, mle = "avg")
```

Arguments

`data`	a categorical data vector of length n.
`coords`	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
`loc.id`	a vector of n values which indicates the directional line of each location. It is usually the output of the function `which_lines`.
`direction`	a d-D numerical vector (or versor) which represents the chosen direction.
`mle`	a character value. If `"trm"`, the trimmed arithmetic average will be used to calculate the mean lengths. If `"mdn"`, the trimmed median will be considered. If `"mlk"`, the maximum likelihood mean lengths will be computed. If `"avg"`, the arithmetic mean will be performed. For backward compatibility reasons, it can accept logical values, so that `TRUE` is equivalent to `"mlk"` and `FALSE` to `"avg"`.

Details

The mean length is the total length occupied by the k-th category divided by the number of its embedded occurrences along lines in the direction ϕ. More robust methods are implemented, such as the trimmed mean and the trimmed median.

If the stratum lengths are censored, the maximum likelihood approach is more appropriate than the arithmetic mean. In this case, the stratum lengths are assumed to be independent realizations from a log-normal random variable. The quantity to maximize is

$$L(\mu_1, \ldots, \mu_K, \sigma_1, \ldots, \sigma_K) = \prod_{i=1}^{m} \prod_{k=1}^{K} \left[\int_{l_i}^{l_i+u_i} \frac{1}{x\sigma_k\sqrt{2}} \exp\left\{ -\frac{(\log x - \mu_k)^2}{2\sigma_k^2} \right\} \right]^{z_{k,i}} \mathrm{d}x,$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)^\top$ and $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_K)^\top$ are vectors of parameters, l_i is the observed stratum length, u_i denotes the upper bound of the censor and $z_{k,i}$ denotes a dummy variable which assumes value 1 if and only if the i-th stratum is referred to the k-th category.

Value

A numeric vector containing the mean length for each observed category.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

`which_lines`

Examples

```
data(ACM)
direction <- c(0,0,1)

# Compute the appartaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction)

# Estimate the mean lengths for each observed category
ml <- mlen(ACM$MAT5, ACM[, 1:3], loc.id, direction, mle = "avg")

# Equivalently
gl <- getlen(ACM$MAT5, ACM[, 1:3], loc.id, direction, zero.allowed = TRUE)
ml1 <- tapply(gl$length, gl$categories, mean)
```

`multi_tpfit` *Multidimensional Model Parameters Estimation*

Description

The function estimates the model parameters of a d-D continuous lag spatial Markov chain. Transition rates matrices along axial directions and proportions of categories are computed.

Usage

```
multi_tpfit(data, coords, method = "ml", tolerance = pi/8,
            rotation = NULL, max.it = 9000, mle = "avg", ...)
```

Arguments

`data`	a categorical data vector of length n.
`coords`	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
`method`	a character object specifying the method to estimate the transition rates. Possible choises are `"ml"` (by default) for the mean length method, `"ils"` for the iterated least squares and `"me"` for the maximum entropy method.
`tolerance`	a numerical value for the tolerance angle (in radians). It's `pi/8` by default.
`rotation`	a numerical vector of length $d - 1$ with rotation angles (in radians), in order to perform the main axes rotation. No rotation is performed by default.
`max.it`	a numerical value which denotes the maximum number of iterations to perform during the optimization phase. It is `9000` by default and used only when the method is `"me"`.
`mle`	a character value to pass to the function `tpfit`. It is `"avg"` by default and not use when the method is `"ils"`.
`...`	other arguments to pass to the functions `multi_tpfit_ml`, `multi_tpfit_ils` or `multi_tpfit_me`.

Details

A d-D continuous-lag spatial Markov chain is probabilistic model which is developed by interpolation of the transition rate matrices computed for the main directions. It defines transition probabilities $\Pr(Z(s+h) = z_k | Z(s) = z_j)$ through

$$\text{expm}(\|h\| R),$$

where h is the lag vector and the entries of R are ellipsoidally interpolated.

The ellipsoidal interpolation is given by

$$|r_{jk}| = \sqrt{\sum_{i=1}^{d} \left(\frac{h_i}{\|h\|} r_{jk,\mathbf{e}_i} \right)^2},$$

where $\mathbf{e}_i$ is a standard basis for a d-D space.

If $h_i < 0$ the respective entries $r_{jk,\mathbf{e}_i}$ are replaced by $r_{jk,-\mathbf{e}_i}$, which is computed as

$$r_{jk,-\mathbf{e}_i} = \frac{p_k}{p_j} r_{kj,\mathbf{e}_i},$$

where p_k and p_j respectively denote the proportions for the k-th and j-th categories. In so doing, the model may describe the anisotropy of the process.

Value

An object of the class `multi_tpfit` is returned. The function `print.multi_tpfit` is used to print the fitted model. The object is a list with the following components:

`coordsnames`	a character vector containing the name of each axis.
`coefficients`	a list containing the transition rates matrices computed for each axial direction.
`prop`	a vector containing the proportions of each observed category.
`tolerance`	a numerical value which denotes the tolerance angle (in radians).

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

`predict.multi_tpfit`, `print.multi_tpfit`, `image.multi_tpfit`, `tpfit`

Examples

```
data(ACM)

# Estimate transition rates matrices and
# proportions for the categorical variable MAT5
multi_tpfit(ACM$MAT5, ACM[, 1:3])

# Estimate transition rates matrices and
# proportions for the categorical variable MAT3
multi_tpfit(ACM$MAT3, ACM[, 1:3])

# Estimate transition rates matrices and
# proportions for the categorical variable PERM
multi_tpfit(ACM$PERM, ACM[, 1:3])
```

`multi_tpfit_ils` *Iterated Least Squares Method for Multidimensional Model Parameters Estimation*

Description

The function estimates the model parameters of a d-D continuous lag spatial Markov chain by the use of the iterated least squares and the bound-constrained Lagrangian methods. Transition rates matrices along axial directions and proportions of categories are computed.

Usage

```
multi_tpfit_ils(data, coords, max.dist = Inf, mpoints = 20,
                tolerance = pi/8, rotation = NULL, q = 10,
                echo = FALSE, ..., mtpfit)
```

Arguments

`data`	a categorical data vector of length n.
`coords`	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
`max.dist`	a numerical value which defines the maximum lag value. It is `Inf` by default.
`mpoints`	a numerical value which defines the number of lag intervals.
`tolerance`	a numerical value for the tolerance angle (in radians). It is `pi/8` by default.
`rotation`	a numerical vector of length $d - 1$ with rotation angles (in radians), in order to perform the main axes rotation. No rotation is performed by default.
`q`	a numerical value greater than one for a constant which controls the growth of the penalization term in the loss function. It is equal to `10` by default.
`echo`	a logical value; if `TRUE`, the function prints some information about the optimization. It is `FALSE` by default.
`...`	other arguments to pass to the function `nlminb`.
`mtpfit`	an object `multi_tpfit` to optimize. If missing, the algorithm starts with null transition rates matrices.

Details

A d-D continuous-lag spatial Markov chain is probabilistic model which is developed by interpolation of the transition rate matrices computed for the main directions. It defines transition probabilities $\Pr(Z(s+h) = z_k | Z(s) = z_j)$ through

$$\text{expm}(\|h\| R),$$

where h is the lag vector and the entries of R are ellipsoidally interpolated.

The ellipsoidal interpolation is given by

$$|r_{jk}| = \sqrt{\sum_{i=1}^{d} \left(\frac{h_i}{\|h\|} r_{jk,\mathbf{e}_i} \right)^2},$$

where $\mathbf{e}_i$ is a standard basis for a d-D space.

If $h_i < 0$ the respective entries $r_{jk,\mathbf{e}_i}$ are replaced by $r_{jk,-\mathbf{e}_i}$, which is computed as

$$r_{jk,-\mathbf{e}_i} = \frac{p_k}{p_j} r_{kj,\mathbf{e}_i},$$

where p_k and p_j respectively denote the proportions for the k-th and j-th categories. In so doing, the model may describe the anisotropy of the process.

In particular, to estimate entries of transition rate matrices computed for the main axial directions, we need to minimize the discrepancies between the empirical transiograms (see `transiogram`) and the predicted transition probabilities.

By the use of the iterated least squares, the diagonal entries of R are constrained to be negative, while the off-diagonal transition rates are constrained to be positive. Further constraints are considered in order to obtain a proper transition rates matrix.

Value

An object of the class `multi_tpfit` is returned. The function `print.multi_tpfit` is used to print the fitted model. The object is a list with the following components:

`coordsnames`	a character vector containing the name of each axis.
`coefficients`	a list containing the transition rates matrices computed for each axial direction.
`prop`	a vector containing the proportions of each observed category.
`tolerance`	a numerical value which denotes the tolerance angle (in radians).

Warning

If the process is not stationary, the optimization algorithm does not converge.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

predict.multi_tpfit, print.multi_tpfit, image.multi_tpfit, tpfit_ils, transiogram

Examples

```
data(ACM)

# Estimate the parameters of a
# multidimensional MC model
multi_tpfit_ils(ACM$MAT3, ACM[, 1:3], 100)
```

`multi_tpfit_me` *Maximum Entropy Method for Multidimensional Model Parameters Estimation*

Description

The function estimates the model parameters of a d-D continuous lag spatial Markov chain. Transition rates matrices along axial directions and proportions of categories are computed.

Usage

```
multi_tpfit_me(data, coords, tolerance = pi/8, max.it = 9000,
               rotation = NULL, mle = "avg")
```

Arguments

`data`	a categorical data vector of length n.
`coords`	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
`tolerance`	a numerical value for the tolerance angle (in radians). It is `pi/8` by default.
`max.it`	a numerical value which denotes the maximum number of iterations to perform during the optimization phase. It is `9000` by default.
`rotation`	a numerical vector of length $d - 1$ with rotation angles (in radians), in order to perform the main axes rotation. No rotation is performed by default.
`mle`	a character value to pass to the function tpfit. It is `"avg"` by default.

Details

A d-D continuous-lag spatial Markov chain is probabilistic model which is developed by interpolation of the transition rate matrices computed for the main directions by the use of the function `tpfit_me`. It defines transition probabilities $\Pr(Z(s+h) = z_k | Z(s) = z_j)$ through

$$\text{expm}(\|h\| R),$$

where h is the lag vector and the entries of R are ellipsoidally interpolated.

The ellipsoidal interpolation is given by

$$|r_{jk}| = \sqrt{\sum_{i=1}^{d} \left(\frac{h_i}{\|h\|} r_{jk,\mathbf{e}_i} \right)^2},$$

where $\mathbf{e}_i$ is a standard basis for a d-D space.

If $h_i < 0$ the respective entries $r_{jk,\mathbf{e}_i}$ are replaced by $r_{jk,-\mathbf{e}_i}$, which is computed as

$$r_{jk,-\mathbf{e}_i} = \frac{p_k}{p_j} r_{kj,\mathbf{e}_i},$$

where p_k and p_j respectively denote the proportions for the k-th and j-th categories. In so doing, the model may describe the anisotropy of the process.

When some entries of the rates matrices are not identifiable, it is suggested to vary the `tolerance` coefficient and the `rotation` angles. This problem may be also avoided if the input argument `mle` is to set to be `"mlk"`.

Value

An object of the class `multi_tpfit` is returned. The function `print.multi_tpfit` is used to print the fitted model. The object is a list with the following components:

`coordsnames`	a character vector containing the name of each axis.
`coefficients`	a list containing the transition rates matrices computed for each axial direction.
`prop`	a vector containing the proportions of each observed category.
`tolerance`	a numerical value which denotes the tolerance angle (in radians).

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

See Also

`predict.multi_tpfit`, `print.multi_tpfit`, `image.multi_tpfit`, `tpfit_me`

Examples

```
data(ACM)

# Estimate transition rates matrices and
# proportions for the categorical variable MAT5
multi_tpfit_me(ACM$MAT5, ACM[, 1:3])

# Estimate transition rates matrices and
# proportions for the categorical variable MAT3
multi_tpfit_me(ACM$MAT3, ACM[, 1:3])

# Estimate transition rates matrices and
# proportions for the categorical variable PERM
multi_tpfit_me(ACM$PERM, ACM[, 1:3])
```

`multi_tpfit_ml` *Mean Length Method for Multidimensional Model Parameters Estimation*

Description

The function estimates the model parameters of a d-D continuous lag spatial Markov chain. Transition rates matrices along axial directions and proportions of categories are computed.

Usage

```
multi_tpfit_ml(data, coords, tolerance = pi/8,
               rotation = NULL, mle = "avg")
```

Arguments

`data`	a categorical data vector of length n.
`coords`	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
`tolerance`	a numerical value for the tolerance angle (in radians). It's `pi/8` by default.
`rotation`	a numerical vector of length $d - 1$ with rotation angles (in radians), in order to perform the main axes rotation. No rotation is performed by default.
`mle`	a character value to pass to the function `tpfit`. It is `"avg"` by default.

Details

A d-D continuous-lag spatial Markov chain is probabilistic model which is developed by interpolation of the transition rate matrices computed for the main directions. It defines transition probabilities $\Pr(Z(s+h) = z_k | Z(s) = z_j)$ through

$$\text{expm}(\|h\|R),$$

where h is the lag vector and the entries of R are ellipsoidally interpolated.

The ellipsoidal interpolation is given by

$$|r_{jk}| = \sqrt{\sum_{i=1}^{d} \left(\frac{h_i}{\|h\|} r_{jk,\mathbf{e}_i} \right)^2},$$

where $\mathbf{e}_i$ is a standard basis for a d-D space.

If $h_i < 0$ the respective entries $r_{jk,\mathbf{e}_i}$ are replaced by $r_{jk,-\mathbf{e}_i}$, which is computed as

$$r_{jk,-\mathbf{e}_i} = \frac{p_k}{p_j} r_{kj,\mathbf{e}_i},$$

where p_k and p_j respectively denote the proportions for the k-th and j-th categories. In so doing, the model may describe the anisotropy of the process.

When some entries of the rates matrices are not identifiable, it is suggested to vary the `tolerance` coefficient and the `rotation` angles. This problem may be also avoided if the input argument `mle` is to set to be `"mlk"`.

Value

An object of the class `multi_tpfit` is returned. The function `print.multi_tpfit` is used to print the fitted model. The object is a list with the following components:

`coordsnames`	a character vector containing the name of each axis.
`coefficients`	a list containing the transition rates matrices computed for each axial direction.
`prop`	a vector containing the proportions of each observed category.
`tolerance`	a numerical value which denotes the tolerance angle (in radians).

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

`predict.multi_tpfit`, `print.multi_tpfit`, `image.multi_tpfit`, `tpfit_ml`

Examples

```
data(ACM)

# Estimate transition rates matrices and
# proportions for the categorical variable MAT5
multi_tpfit_ml(ACM$MAT5, ACM[, 1:3])
```

```
# Estimate transition rates matrices and
# proportions for the categorical variable MAT3
multi_tpfit_ml(ACM$MAT3, ACM[, 1:3])

# Estimate transition rates matrices and
# proportions for the categorical variable PERM
multi_tpfit_ml(ACM$PERM, ACM[, 1:3])
```

`pemt` *Multi-directional Transiograms Estimation*

Description

The function computes the multi-directional transiograms without any ellipsoidal interpolation for 2-D sections.

Usage

```
pemt(data, coords, mpoints, which.dire, max.dist,
     tolerance = pi/8, rotation = NULL, mle = "avg")
```

Arguments

`data`	a categorical data vector of length n.
`coords`	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
`mpoints`	the number of points per axes. It controls the accuracy of images to plot.
`which.dire`	a vector with two chosen axial directions. If omitted, all 2-D sections are plotted.
`max.dist`	a scalar or a vector of maximum length for the chosen axial directions.
`tolerance`	a numerical value for the tolerance angle (in radians). It's `pi/8` by default.
`rotation`	a numerical vector of length $d - 1$ with rotation angles (in radians), in order to perform the main axes rotation when multidimensional transiogram is estimated. No rotation is performed by default. See `multi_tpfit_ml`.
`mle`	a character value to pass to the function `tpfit_ml`. It is `"avg"` by default.

Details

A multidimensional transiogram is a diagram which shows the transition probabilities for a single pair of categories. The probability is computed for any lag vector h through

$$\text{expm}(\|h\| R_h),$$

where entries of R_h are not ellipsoidally interpolated, but they are estimated for the direction specified by the vector h.

In particular cases, some entries of the estimated matrix R_h might be not finite, so that the exponential matrix is computable and the resulting transition probabilities are set to be `NaN`. If `mle = "mlk"`, this problem may be partially solved.

The exponential matrix is evaluated by the scaling and squaring algorithm.

Value

An object of class `pemt` is returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Higham, N. J. (2008) *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

`multi_tpfit_ml`, `tpfit_ml`, `image.pemt`, `plot.transiogram`

Examples

```
data(ACM)

# Compute a 2-D section of a
# multi-directional transiogram
pemt(ACM$MAT3, ACM[, 1:3], 2,
     max.dist = c(200, 200, 20),
     which.dire=c(1, 3), mle = "mdn")
```

---

persp.multi_tpfit *Perspective Plots with Multidimensional Transiograms*

Description

The function draws perspective-plots the 2-D sections of a predicted multidimensional transiograms computed through ellipsoidal interpolation.

Usage

```
## S3 method for class 'multi_tpfit'
persp(x, mpoints, which.dire, max.dist, main,
      mar, ask = TRUE, col = "white", ...)
```

Arguments

x	an object of the class multi_tpfit, typically with the output of the function multi_tpfit.
mpoints	the number of points per axes. It controls the accuracy of images to plot.
which.dire	a vector with two chosen axial directions. If omitted, all 2-D sections are plotted.
max.dist	a scalar or a vector of maximum length for the chosen axial directions.
main	the main title (on top) whose font and size are fixed.
mar	a scalar or a numerical vector of the form c(bottom, left, top, right) which gives the number of margin lines to be specified on the four sides of image to plot. See par(mar=.).
ask	a logical value; if TRUE, the user is asked for input, before each plot. See par(ask=.).
col	a list of colors which is usually generated by rev(heat.colors()), or with other function for colors.
...	other arguments to pass to the function persp.

Details

A multidimensional transiogram is a diagram which shows the transition probabilities for a single pair of categories. It is computed for any lag vector h through

$$\text{expm}(\|h\|R),$$

where entries of R are ellipsoidally interpolated (see multi_tpfit).

The exponential matrix is evaluated by the scaling and squaring algorithm.

Value

An image is produced on the current graphics device. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Higham, N. J. (2008) *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

multi_tpfit, persp.multi_tpfit, persp, pemt, persp.pemt, plot.transiogram

Examples

```
data(ACM)

# Estimate model parameter
x <- multi_tpfit(ACM$MAT5, ACM[, 1:3])

# Set short names for categories 3 and 4
names(x$prop)[3:4] <- c("Clay and Sand", "Gravel and Sand")

# 3D-Plot for a 2-D theoretical sections of
# a multidimensional transiogram
persp(x, 15, max.dist = c(200, 200, 20), which.dire = 2:3,
    mar = .7, col = rainbow(500), theta = 15, phi = 45)
```

`persp.pemt` *Perspective Plots with Multi-directional Transiograms*

Description

The function draws perspective-plots the 2-D sections of a multi-directional transiogram computed without any ellipsoidal interpolation.

Usage

```
## S3 method for class 'pemt'
persp(x, main, mar, ask = TRUE, col = "white", ...)
```

Arguments

`x`	an object of the class `pemt`, typically with the output of the function `pemt`.
`main`	the main title (on top) whose font and size are fixed.
`mar`	a scalar or a numerical vector of the form `c(bottom, left, top, right)` which gives the number of margin lines to be specified on the four sides of image to plot. See `par(mar=.)`.
`ask`	a logical value; if `TRUE`, the user is asked for input, before each plot. See `par(ask=.)`.
`col`	a list of colors which is usually generated by `rev(heat.colors())`, or with other function for `colors`.
`...`	other arguments to pass to the function `persp`.

Details

A multidimensional transiogram is a diagram which shows the transition probabilities for a single pair of categories. The probability is computed for any lag vector h through

$$\text{expm}(\|h\|R_h),$$

where entries of R_h are not ellipsoidally interpolated, but they are estimated for the direction specified by the vector h.

The exponential matrix is evaluated by the scaling and squaring algorithm.

Value

An image is produced on the current graphics device. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Higham, N. J. (2008) *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

pemt, persp.multi_tpfit, persp, multi_tpfit, image.pemt, plot.transiogram

Examples

```
data(ACM)

# Compute a 2-D section of a
# multi-directional transiogram
psEmpTr <- pemt(ACM$MAT3, ACM[, 1:3], 2,
                max.dist = c(200, 200, 20),
                which.dire = c(1, 3))

# 3D-Plot for a 2-D sections of
# a multi-directional transiogram
persp(psEmpTr, col = rainbow(500), mar = .7,
      theta = 15, phi = 45)
```

---

plot.density.lengths *Plot Empirical Densities Estimates of Stratum Lengths*

Description

The function plot the empirical densities of stratum lengths computed along a given direction.

Usage

```
## S3 method for class 'density.lengths'
plot(x, main = NULL, xlab = NULL, ylab = "Density", type = "l",
     zero.line = TRUE, ...)
```

Arguments

x	an object of the class density.lengths, typically with the output of the function density.lengths.
main	an overall title for the plot.
xlab	a title for the x-axis.
ylab	a title for the y-axis.
type	plotting parameter for the type of graphic (see plot).
zero.line	logical value. If TRUE (by default), the function adds a base line at $y = 0$.
...	other plotting parameters.

Value

An image is produced on the current graphics device. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

density.default, density.lengths, plot, print.density.lengths

Examples

```
data(ACM)
direction <- c(0,0,1)

# Compute the appertaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction)

# Estimate stratum lengths
gl <- getlen(ACM$MAT3, ACM[, 1:3], loc.id, direction)

# Compute the empirical densities of stratum log-lengths
dgl <- density(gl, log = TRUE)

# Plot the empirical densities of stratum log-lengths
plot(dgl)
```

plot.hist.lengths *Plot Histograms of Stratum Lengths*

Description

The function plots objects of class `hist.lengths`.

Usage

```
## S3 method for class 'hist.lengths'
plot(x, ...)
```

Arguments

`x`	an object of the class `hist.lengths`, typically with the output of the function `hist.lengths`.
`...`	further plotting parameters.

Value

An image is produced on the current graphics device. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

`hist`, `hist.lengths`, `plot`, `print.density.lengths`

Examples

```
data(ACM)
direction <- c(0,0,1)

# Compute the appertaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction)

# Estimate stratum lengths
gl <- getlen(ACM$MAT3, ACM[, 1:3], loc.id, direction)

# Compute the histograms
hgl <- hist(gl, plot = FALSE)

# Plot the histograms
plot(hgl, col = "#efffef")
```

plot.lengths *Plot Stratum Lengths*

Description

The function makes a graphical representation of the stratum lengths.

Usage

```
## S3 method for class 'lengths'
plot(x, ..., log = FALSE, zeros.rm = TRUE)
```

Arguments

x	an object of the class lengths, typically with the output of the function getlen.
...	other arguments to pass to the function boxplot.
log	a logical value. If TRUE, the logarithm of the stratum lengths will be plotted. It is FALSE by default.
zeros.rm	a logical value. If FALSE, the image will be drawn by including zero values. It is TRUE by default.

Details

The box-and-whisker plots give some information about the distribution of the stratum lengths for the observed categories along a given direction.

Value

An image is produced on the current graphics device; by the use of boxplot.lengths, the same image is produced. The function returns a list with the following components:

stats	a matrix containing the values used to plot the box-and-whisker plots.
n	a vector with the number of observations for each category.
conf	a matrix containing further values to draw the lower and upper extremes of the notch.
out	a vectors with the values of the outlier points.
group	a vector whose elements indicate to which category the outlier belongs.
names	a character vector with the names of each category.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

boxplot.lengths, boxplot, getlen

Examples

```
data(ACM)
direction <- c(0,0,1)

# Compute the appertaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction)

# Estimate stratum lengths
gl <- getlen(ACM$MAT3, ACM[, 1:3], loc.id, direction)

# Plot the object gl

plot(gl)
```

---

plot.transiogram *Plot One-dimensional Transiograms*

Description

The function makes a graphical representation of transition probabilities by the use of transiogram.

Usage

```
## S3 method for class 'transiogram'
plot(x, ..., main, legend = FALSE, ci = NULL)
```

Arguments

x	an object of the class transiogram, typically with the output of the function transiogram or predict.tpfit.
...	other arguments to pass to the function plot.
main	the main title (on top) whose font and size are fixed.
legend	a logical value; if TRUE, the legend is plot on the bottom.
ci	a numerical value in the interval (0, 1) denoting the confidence of the interval around transition probabilities. If NULL (by default), no confidence interval is plotted.

Details

Transiogram is a diagram which is drawn for a single pair of categories in the direction ϕ. It shows the transition probabilities in the y-axis for some specific lags in the x-axis.

Confidence intervals are computed on the log odds of the transition probabilities. The approximation of the confidence bounds is based on the delta method applied on the logistic transformation.

Value

An image is produced on the current graphics device. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Li, W. (2007) Transiograms for Characterizing Spatial Variability of Soil Classes. *Soil Science Society of America Journal*, **71**(3), 881-893.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

tpfit, predict.tpfit, mixplot, image.multi_tpfit, plot

Examples

```
data(ACM)

# Estimate empirical transition
# probabilities by points
ETr <- transiogram(ACM$MAT3, ACM[, 1:3], c(0, 0, 1), 100, 100)

# Estimate the transition rate matrix
RTm <- tpfit(ACM$MAT3, ACM[, 1:3], c(0, 0, 1))

# Compute transition probabilities
# from the one-dimensional MC model
TPr <- predict(RTm, lags = ETr$lags)

# Plot empirical transition probabilities
plot(ETr, type = "l", ci = 0.99)

# Plot theoretical transition probabilities
plot(TPr, type = "l")
```

---

predict.multi_tpfit *Compute Theoretical Multidimensional Transiograms*

Description

The function computes theoretical transition probabilities of a d-D continuous-lag spatial Markov chain for a specified set of lags.

Usage

```
## S3 method for class 'multi_tpfit'
predict(object, lags, byrow = TRUE, ...)
```

Arguments

`object`	an object of the class `multi_tpfit`, typically with the output of the function `multi_tpfit`.
`lags`	a lag vector or matrix of d-D lags.
`byrow`	a logical value; if `TRUE` (by default), each row of matrix argument `lags` will be considered as a lag vector.
`...`	further arguments passed from other methods.

Details

A d-D continuous-lag spatial Markov chain is probabilistic model which is developed by interpolation of the transition rate matrices computed for the main directions. It defines the transition probability $\Pr(Z(s+h) = z_k | Z(s) = z_j)$ through the entry t_{jk} of the following matrix

$$T = \text{expm}(\|h\| R),$$

where h is the lag vector and the entries of R are ellipsoidally interpolated.

Value

An object of the class `multi_transiogram` is returned. The `print.multi_transiogram` function is used to print computed probabilities. The object is a list with the following components:

`Tmat`	a 3-D array containing the probabilities.
`lags`	a matrix containing the lag vectors.
`type`	a character string which specifies that computed probabilities are theoretical.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

`multi_tpfit`, `print.multi_tpfit`, `image.multi_tpfit`, `tpfit`, `transiogram`

Examples

```
data(ACM)

# Estimate the parameters of a
# multidimensional MC model
RTm <- multi_tpfit(ACM$MAT3, ACM[, 1:3])

# Generate the matrix of
# multidimensional lags
lags <- expand.grid(X=-1:1, Y=-1:1, Z=-1:1)
lags <- as.matrix(lags)

# Compute transition probabilities
# from the multidimensional MC model
predict(RTm, lags)
```

`predict.tpfit` *Compute Theoretical One-dimensional Transiograms*

Description

The function computes theoretical transition probabilities of a 1-D continuous-lag spatial Markov chain for a specified set of lags.

Usage

```
## S3 method for class 'tpfit'
predict(object, lags, ...)
```

Arguments

`object`	an object of the class `tpfit`, typically with the output of the function `tpfit`.
`lags`	a vector of 1-D lags.
`...`	further arguments passed from other methods.

Details

A 1-D continuous-lag spatial Markov chain is probabilistic model which involves a transition rate matrix R computed for the direction ϕ. It defines the transition probability $\Pr(Z(s+h) = z_k | Z(s) = z_j)$ through the entry t_{jk} of the following matrix

$$T = \text{expm}(hR),$$

where h is a positive lag value.

Value

An object of the class `transiogram` is returned. The function `print.transiogram` is used to print computed probabilities. The object is a list with the following components:

`Tmat`	a 3-D array containing the probabilities.
`lags`	a vector containing one-dimensional lags.
`type`	a character string which specifies that computed probabilities are theoretical.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

`tpfit`, `print.tpfit`, `plot.transiogram`, `transiogram`, `multi_tpfit`

Examples

```
data(ACM)

# Estimate the parameters of a
# one-dimensional MC model
RTm <- tpfit(ACM$MAT3, ACM[, 1:3], c(0, 0, 1))

# Compute transition probabilities
# from the one-dimensional MC model
predict(RTm, lags = 0:2/2)
```

`print.density.lengths` *Printing Empirical Densities Estimates of Stratum Lengths*

Description

he function a summary of the empirical density stratum lengths calculated by `density.lengths`.

Usage

```
## S3 method for class 'density.lengths'
print(x, digits = NULL, ...)
```

Arguments

x	an object of the class density.lengths, typically with the output of the function density.lengths.
digits	minimal number of digits, see print.default.
...	further arguments to pass to the function summary.data.frame.

Value

A summary of the empirical distributions is printed on the screen or other output devices. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

density.lengths, plot.density.lengths

Examples

```
data(ACM)
direction <- c(0,0,1)

# Compute the appartaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction)

# Estimate stratum lengths
gl <- getlen(ACM$MAT3, ACM[, 1:3], loc.id, direction)

# Compute the empirical densities of stratum lengths
dgl <- density(gl)

# Print the empirical densities of stratum lengths
print(dgl)
```

---

print.lengths *Printing Stratum Lengths for Each Observed Category*

Description

The function prints stratum lengths given by getlen.

Usage

```
## S3 method for class 'lengths'
print(x, ...)
```

Arguments

x	an object of the class lengths, typically with the output of the function getlen.
...	further arguments passed to or from other methods.

Value

Stratum lengths grouped by category are printed on the screen or other output devices. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

getlen

Examples

```
data(ACM)
direction <- c(0,0,1)

# Compute the appertaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction)

# Estimate stratum lengths
gl <- getlen(ACM$MAT3, ACM[, 1:3], loc.id, direction)

# Print stratum lengths
print(gl)
```

---

print.multi_tpfit *Printing Model Parameters for Multidimensional Continuous Lag Spatial MC*

Description

The function prints parameter estimation results given by multi_tpfit.

Usage

```
## S3 method for class 'multi_tpfit'
print(x, ...)
```

Arguments

x	an object of the class multi_tpfit, typically with the output of the function multi_tpfit.
...	further arguments passed to or from other methods.

Value

Estimation results are printed on the screen or other output devices. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

multi_tpfit

Examples

```
data(ACM)

# Estimate the parameters of a
# multidimensional MC models
MoPa <- multi_tpfit(ACM$MAT5, ACM[, 1:3])

# Print results
print(MoPa)
```

---

print.multi_transiogram

Printing Theoretical Multidimensional Transiograms

Description

The function prints theoretical transition probabilities given by predict.multi_tpfit.

Usage

```
## S3 method for class 'multi_transiogram'
print(x, ...)
```

Arguments

x	an object of the class multi_transiogram, typically with the output of the function predict.multi_tpfit.
...	further arguments passed to or from other methods.

Value

Transition probabilities are printed on the screen or other output devices. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

predict.multi_tpfit

Examples

```
data(ACM)

# Estimate the parameters of a
# multidimensional MC model
RTm <- multi_tpfit(ACM$MAT3, ACM[, 1:3])

# Generate the matrix of
# multidimensional lags
lags <- expand.grid(X=-1:1, Y=-1:1, Z=-1:1)
lags <- as.matrix(lags)

# Compute transition probabilities
# from the multidimensional MC model
TrPr <- predict(RTm, lags)

# Print results
print(TrPr)
```

---

print.summary.lengths *Printing Stratum Lengths Summary for Each Observed Category*

Description

The function prints the summary of stratum lengths given by summary.lengths.

Usage

```
## S3 method for class 'summary.lengths'
print(x, ...)
```

Arguments

x	an object of the class summary.lengths, typically with the output of the function summary.lengths.
...	further arguments passed to or from other methods.

Value

The summary of stratum lengths grouped by category is printed on the screen or other output devices. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

getlen, summary.lengths

Examples

```
data(ACM)
direction <- c(0,0,1)

# Compute the appartaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction)

# Estimate stratum lengths
gl <- getlen(ACM$MAT3, ACM[, 1:3], loc.id, direction)

# Summarize the stratum lengths
sgl <- summary(gl)

# Print the summary of stratum lengths
print(sgl)
```

---

print.tpfit *Printing Model Parameters for One-dimensional Continuous Lag Spatial MC*

Description

The function prints parameter estimation results given by tpfit.

Usage

```
## S3 method for class 'tpfit'
print(x, ...)
```

Arguments

x	an object of the class tpfit, typically with the output of the function tpfit.
...	further arguments passed to or from other methods.

Value

Estimation results are printed on the screen or other output devices. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

tpfit

Examples

```
data(ACM)

# Estimate the parameters of a
# one-dimensional MC model
MoPa <- tpfit(ACM$MAT5, ACM[, 1:3], c(0, 0, 1))

# Print results
print(MoPa)
```

`print.transiogram` *Printing Theoretical or Empirical One-dimensional Transiograms*

Description

The function prints transition probabilities given by `predict.multi_tpfit` or `transiogram`.

Usage

```
## S3 method for class 'transiogram'
print(x, ...)
```

Arguments

`x`	an object of the class `transiogram`, typically with the output of the function `predict.tpfit` or `transiogram`.
`...`	further arguments passed to or from other methods.

Value

Transition probabilities are printed on the screen or other output devices. No values are returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

`transiogram`, `predict.tpfit`

Examples

```
data(ACM)

# Estimate the parameters of a
# one-dimensional MC model
RTm <- tpfit(ACM$MAT5, ACM[, 1:3], c(0, 0, 1))
```

```
# Compute theoretical transition probabilities
# from the one-dimensional MC model
TTPr <- predict(RTm, lags = 0:2/2)

# Compute empirical transition probabilities
ETPr <- transiogram(ACM$MAT5, ACM[, 1:3], c(0, 0, 1), 200, 20)

# Print results
print(TTPr)
print(ETPr)
```

---

quench *Conditional Simulation Adjuster Via Quenching Algorithm*

Description

The function adjusts a simulated random field generated by the `sim` function.

Usage

```
quench(x, data, coords, sim, GA = FALSE, optype = c("param",
       "fullprobs", "semiprobs", "coordprobs"), max.it = 1000,
       knn = 12)
```

Arguments

`x`	an object of the class `multi_tpfit`, typically with the output of the function `multi_tpfit`.
`data`	a categorical data vector of length n.
`coords`	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
`sim`	an object of the class `spsim`, it is usually the output of the function `sim`.
`GA`	a logical value; if `TRUE`, the function performs the Genetic Algorithm instead of the Simulated Annealing.
`optype`	a character which denotes the objective function to compute when the optimization is performed.
`max.it`	a numerical value which specifies the maximum number of iterations to stop the optimization algorithm. For proper results, it should be a multiple of the number of simulation points.
`knn`	an integer value which specifies the number of k-nearest neighbours for each simulation point. An optimal number is between 4 and 12. If `NULL` all observations are considered (just for very small dataset!!). It is `12` by default.

Details

This method perform a simulated annealing or a genetic algorithm to modify the simulation results, in order to reduce artifacts effects. In practice, each simulated configuration is adjusted to reach a pattern similar to the observed sample data. There are several objective functions for this purpose, by setting optype equal to "param" the optimization is performed through parametric methods. The alternatives "fullprobs" and "semiprobs" are based on transition probabilities computed among simulation points, while the option "coordprobs" is based on transition probabilities calculated among observation and simulation points.

This procedure should be executed by setting max.it equal at least to the simulation grid size, or its multiples.

Value

A data frame containing the simulation grid, the simulated random field, predicted values and the approximated probabilities.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1996) Transition Probability-Based Indicator Geostatistics. *Mathematical Geosciences*, **28**(4), 453-476.

Carle, S. F. (1999) T-PROGS: Transition Probability Geostatistical Software. University of California, Davis.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

Weise, T. (2009) *Global Optimization Algorithms - Theory and Application*. https://www.researchgate.net/publication/200622167_Global_Optimization_Algorithm_Theory_and_Application.

See Also

sim_ck, sim_ik, sim_mcs, sim_path

Examples

```
data(ACM)

# Model parameters estimation for the
# multinomial categorical simulation
x <- multi_tpfit(ACM$MAT5, ACM[, 1:3])

# Generate the simulation grid
mygrid <- list()
mygrid$X <- seq(min(ACM$X), max(ACM$X), length = 20)
mygrid$Y <- seq(min(ACM$Y), max(ACM$Y), length = 20)
mygrid$Z <- -40 * 0:9 - 1
mygrid <- as.matrix(expand.grid(mygrid$X, mygrid$Y, mygrid$Z))
```

```
# Simulate the random field through
# Ordinary Indicator Kriging algorithm
myOIKSim <- sim_ik(x, ACM$MAT5, ACM[, 1:3], mygrid)

# Perform the quenching algorithm
# to adjust simulation
quench(x, ACM$MAT5, ACM[, 1:3], myOIKSim, optype = "coordprobs",
       max.it = 2, knn = 12)
```

`setCores` *Set the number of CPU cores for HPC*

Description

The function set the number of CPU cores for parallel computation by the use of OpenMP library (https://www.openmp.org/). If the package was not complied with the library OpenMP (>= 3.0), this function is disabled.

Usage

```
setCores(n)
```

Arguments

`n`	an integer value denoting the number of CPU cores to use; if it exceeds the total number of cores, all of them will be used. If missing, the number of CPU cores in use will be displayed.

Details

When the package is loaded, only one CPU core is used.

Value

The total number of CPU cores in use will be returned and a message will be displayed. If the package was not complied with the library OpenMP (>= 3.0), the value one will be returned.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

SunTM ONE Studio 8 (2003) *OpenMP API User's Guide*. Sun Microsystems Inc., Santa Clara, U.S.A.

Examples

```
#Display the number of CPU cores in use
setCores()

#Set 2 CPU cores for parallel computation
setCores(2)

#Set 1 CPU core for serial computation
setCores(1)
```

`sim` *Random Field Simulation*

Description

The function simulates a random field. The simulation methods available are based on Indicator Kriging techniques (IK and CK), Fixed and Random Path (PATH) and Multinomial Categorical Simulation (MCS).

Usage

```
sim(x, data, coords, grid, method = "ik", ..., entropy = FALSE)
```

Arguments

`x`	an object of the class `multi_tpfit`, typically with the output of the function `multi_tpfit`.
`data`	a categorical data vector of length n.
`coords`	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
`grid`	an $m \times d$ matrix where each row denotes the d-D coordinates in the simulation grid.
`method`	a character object specifying the method to simulate the random field. Possible choises are `"ik"` (by default) for the indicator Kriging, `"ck"` for the indicator coKriging, `"path"` for the fixed and random path and `"mcs"` for the multinomial categorical simulation method.
`...`	other arguments to pass to the functions `sim_ik`, `sim_ck`, `sim_path` or `sim_mcs`.
`entropy`	a logical value. If `TRUE`, the prediction uncertainties are computed through the entropy (and standardized entropy). The default value is `FALSE`.

Details

The methods implemented compute the approximation of posterior probabilities

$$\Pr\left(Z(\mathbf{s}_0) = z_k \middle| \bigcap_{i=1}^{n} Z(\mathbf{s}_i) = z(\mathbf{s}_i)\right).$$

Once the probabilities are calculated for all the points in the simulation grid, the predictions (based on most probable category) and simulations are returned.

Value

A data frame containing the simulation grid, the simulated random field, predicted values and the approximated probabilities is returned. Two extra columns respectively denoting the entropy and standardized entorpy are bindend to the data frame when argument `entropy = TRUE`.

References

Allard, D., D'Or, D., Froidevaux, R. (2011) An efficient maximum entropy approach for categorical variable prediction. *European Journal of Soil Science*, **62**(3), 381-393.

Carle, S. F., Fogg, G. E. (1996) Transition Probability-Based Indicator Geostatistics. *Mathematical Geosciences*, **28**(4), 453-476.

Carle, S. F. (1999) T-PROGS: Transition Probability Geostatistical Software. University of California, Davis.

Li, W. (2007) A Fixed-Path Markov Chain Algorithm for Conditional Simulation of Discrete Spatial Variables. *Mathematical Geology*, **39**(2), 159-176.

Li, W. (2007) Markov Chain Random Fields for Estimation of Categorical Variables. *Mathematical Geology*, **39**(June), 321-335.

Pickard, D. K. (1980) Unilateral Markov Fields. *Advances in Applied Probability*, **12**(3), 655-671.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

Weise, T. (2009) *Global Optimization Algorithms - Theory and Application*. https://www.researchgate.net/publication/200622167_Global_Optimization_Algorithm_Theory_and_Application.

See Also

`sim_ik`, `sim_ck`, `sim_path`, `sim_mcs`

Examples

```
data(ACM)

# Model parameters estimation for the
# multinomial categorical simulation
x <- multi_tpfit(ACM$MAT5, ACM[, 1:3])

# Generate the simulation grid
mygrid <- list()
mygrid$X <- seq(min(ACM$X), max(ACM$X), length = 20)
mygrid$Y <- seq(min(ACM$Y), max(ACM$Y), length = 20)
mygrid$Z <- -40 * 0:9 - 1
mygrid <- as.matrix(expand.grid(mygrid$X, mygrid$Y, mygrid$Z))

# Simulate the random field through
# Simple Indicator Kriging algorithm and
mySim <- sim(x, ACM$MAT5, ACM[, 1:3], mygrid)
```

sim_ck *Conditional Simulation Based on Indicator Cokriging*

Description

The function simulates a random field through the Indicator Cokriging technique.

Usage

```
sim_ck(x, data, coords, grid, knn = 12, ordinary = TRUE, entropy = FALSE)
```

Arguments

x	an object of the class multi_tpfit, typically with the output of the function multi_tpfit.
data	a categorical data vector of length n.
coords	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
grid	an $m \times d$ matrix where each row denotes the d-D coordinates in the simulation grid.
knn	an integer value which specifies the number of k-nearest neighbours for each simulation point. An optimal number is between 4 and 12. If NULL all observations are considered (just for very small dataset!!). It is 12 by default.
ordinary	a logical value; if FALSE, the probabilities are computed through the Simple coKriging technique, otherwise the Ordinary coKriging method is used.
entropy	a logical value. If TRUE, the prediction uncertainties are computed through the entropy (and standardized entropy). The default value is FALSE.

Details

This method computes an approximation of posterior probabilities

$$\Pr\left(Z(\mathbf{s}_0) = z_k \left| \bigcap_{i=1}^{n} Z(\mathbf{s}_i) = z(\mathbf{s}_i) \right. \right).$$

The probability is calculated as the weighted sum of indicator variables which denote the presence of the k-th category in observed points $\mathbf{s}_i$. Weights involved in the sum are the solution of a system of equations.

Probabilities approximated are usually truncated and normalized with respect to the probability constraints, because such probabilities might lie outside the interval $[0, 1]$. The normalization procedure is designed such that it is not possible to obtain vectors such that the sum of their probabilities is always equal to one.

When an initial configuration is simulated, it should be modified to reach a pattern similar to the sample by the use of the quench function.

Value

A data frame containing the simulation grid, the simulated random field, predicted values and the approximated probabilities is returned. Two extra columns respectively denoting the entropy and standardized entorpy are bindend to the data frame when argument `entropy = TRUE`.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1996) Transition Probability-Based Indicator Geostatistics. *Mathematical Geosciences*, **28**(4), 453-476.

Carle, S. F. (1999) T-PROGS: Transition Probability Geostatistical Software. University of California, Davis.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

Weise, T. (2009) *Global Optimization Algorithms - Theory and Application*. https://www.researchgate.net/publication/200622167_Global_Optimization_Algorithm_Theory_and_Application.

See Also

sim_ik, sim_mcs, sim_path

Examples

```
data(ACM)

# Model parameters estimation for the
# multinomial categorical simulation
x <- multi_tpfit(ACM$MAT5, ACM[, 1:3])

# Generate the simulation grid
mygrid <- list()
mygrid$X <- seq(min(ACM$X), max(ACM$X), length = 20)
mygrid$Y <- seq(min(ACM$Y), max(ACM$Y), length = 20)
mygrid$Z <- -40 * 0:9 - 1
mygrid <- as.matrix(expand.grid(mygrid$X, mygrid$Y, mygrid$Z))

# Simulate the random field through
# Simple Indicator Cokriging algorithm
mySCKSim <- sim_ck(x, ACM$MAT5, ACM[, 1:3], mygrid, ordinary = FALSE)

# Simulate the random field through
# Ordinary Indicator Cokriging algorithm
myOCKSim <- sim_ck(x, ACM$MAT5, ACM[, 1:3], mygrid)
```

sim_ik *Conditional Simulation Based on Indicator Kriging*

Description

The function simulates a random field through the Indicator Kriging technique.

Usage

```
sim_ik(x, data, coords, grid, knn = 12, ordinary = TRUE, entropy = FALSE)
```

Arguments

x	an object of the class multi_tpfit, typically with the output of the function multi_tpfit.
data	a categorical data vector of length n.
coords	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
grid	an $m \times d$ matrix where each row denotes the d-D coordinates in the simulation grid.
knn	an integer value which specifies the number of k-nearest neighbours for each simulation point. An optimal number is between 4 and 12. If NULL all observations are considered (just for very small dataset!!). It is 12 by default.
ordinary	a logical value; if FALSE, the probabilities are computed through the Simple Kriging technique, otherwise the Ordinary Kriging method is used.
entropy	a logical value. If TRUE, the prediction uncertainties are computed through the entropy (and standardized entropy). The default value is FALSE.

Details

This method computes an approximation of posterior probabilities

$$\Pr\left(Z(\mathbf{s}_0) = z_k \left| \bigcap_{i=1}^{n} Z(\mathbf{s}_i) = z(\mathbf{s}_i) \right. \right).$$

The probability is calculated as the sum of the observed proportion and the weighted sum of indicator variables which denote the presence of the k-th category in observed points $\mathbf{s}_i$. Weights involved in the sum are the solution of a system of equations.

Probabilities approximated are usually truncated and normalized with respect to the probability constraints, because such probabilities might lie outside the interval $[0, 1]$. The normalization procedure is designed such that it is not possible to obtain vectors such that the sum of their probabilities is always equal to one.

When an initial configuration is simulated, it should be modified to reach a pattern similar to the sample by the use of the quench function.

Value

A data frame containing the simulation grid, the simulated random field, predicted values and the approximated probabilities is returned. Two extra columns respectively denoting the entropy and standardized entorpy are bindend to the data frame when argument `entropy = TRUE`.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1996) Transition Probability-Based Indicator Geostatistics. *Mathematical Geosciences*, **28**(4), 453-476.

Carle, S. F. (1999) T-PROGS: Transition Probability Geostatistical Software. University of California, Davis.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

Weise, T. (2009) *Global Optimization Algorithms - Theory and Application*. https://www.researchgate.net/publication/200622167_Global_Optimization_Algorithm_Theory_and_Application.

See Also

sim_ck, sim_mcs, sim_path

Examples

```
data(ACM)

# Model parameters estimation for the
# multinomial categorical simulation
x <- multi_tpfit(ACM$MAT5, ACM[, 1:3])

# Generate the simulation grid
mygrid <- list()
mygrid$X <- seq(min(ACM$X), max(ACM$X), length = 20)
mygrid$Y <- seq(min(ACM$Y), max(ACM$Y), length = 20)
mygrid$Z <- -40 * 0:9 - 1
mygrid <- as.matrix(expand.grid(mygrid$X, mygrid$Y, mygrid$Z))

# Simulate the random field through
# Simple Indicator Kriging algorithm
mySIKSim <- sim_ik(x, ACM$MAT5, ACM[, 1:3], mygrid, ordinary = FALSE)

# Simulate the random field through
# Ordinary Indicator Kriging algorithm
myOIKSim <- sim_ik(x, ACM$MAT5, ACM[, 1:3], mygrid)
```

sim_mcs *Multinomial Categorical Simulation*

Description

The function simulates a random field through the Multinomial Categorical Simulation technique (MCS).

Usage

```
sim_mcs(x, data, coords, grid, knn = NULL, entropy = FALSE)
```

Arguments

x	an object of the class multi_tpfit, typically with the output of the function multi_tpfit.
data	a categorical data vector of length n.
coords	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
grid	an $m \times d$ matrix where each row denotes the d-D coordinates in the simulation grid.
knn	an integer value which specifies the number of k-nearest neighbours for each simulation point. If NULL (by default), all observations are considered.
entropy	a logical value. If TRUE, the prediction uncertainties are computed through the entropy (and standardized entropy). The default value is FALSE.

Details

This method computes an approximation of posterior probabilities

$$\Pr\left(Z(\mathbf{s}_0) = z_k \middle| \bigcap_{i=1}^{n} Z(\mathbf{s}_i) = z(\mathbf{s}_i)\right).$$

The algorithm is based on the Bayesian maximum entropy approach and it honours both the model structure and observed data.

Value

A data frame containing the simulation grid, the simulated random field, predicted values and the approximated probabilities is returned. Two extra columns respectively denoting the entropy and standardized entorpy are bindend to the data frame when argument entropy = TRUE.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Allard, D., D'Or, D., Froidevaux, R. (2011) An efficient maximum entropy approach for categorical variable prediction. *European Journal of Soil Science*, **62**(3), 381-393.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

sim_ck, sim_ik, sim_path

Examples

```
data(ACM)

# Model parameters estimation for the
# multinomial categorical simulation
x <- multi_tpfit(ACM$MAT5, ACM[, 1:3])

# Generate the simulation grid
mygrid <- list()
mygrid$X <- seq(min(ACM$X), max(ACM$X), length = 3)
mygrid$Y <- seq(min(ACM$Y), max(ACM$Y), length = 3)
mygrid$Z <- -40 * 0:9 - 1
mygrid <- as.matrix(expand.grid(mygrid$X, mygrid$Y, mygrid$Z))

# Simulate the random field
myMCSim <- sim_mcs(x, ACM$MAT5, ACM[, 1:3], mygrid)
```

---

sim_path *Conditional Simulation Based on Path Algorithms*

Description

The function simulates a random field through the Fixed Path algorithm or Random Path technique.

Usage

```
sim_path(x, data, coords, grid, radius, fixed = FALSE, entropy = FALSE)
```

Arguments

x	an object of the class multi_tpfit, typically with the output of the function multi_tpfit.
data	a categorical data vector of length n.
coords	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
grid	an $m \times d$ matrix where each row denotes the d-D coordinates in the simulation grid.

radius	a numerical value that specifies a proper radius to search the nearest observed points within a d-D sphere.
fixed	a logical value; if TRUE, the fixed path algorithm is performed. The random path algorithm is performed by default.
entropy	a logical value. If TRUE, the prediction uncertainties are computed through the entropy (and standardized entropy). The default value is FALSE.

Details

These methods compute an approximation of posterior probabilities

$$\Pr\left(Z(\mathbf{s}_0) = z_k \left| \bigcap_{i=1}^{n} Z(\mathbf{s}_i) = z(\mathbf{s}_i) \right.\right).$$

Path algorithms are based on Pickard random fields, so that the states of such chain at any unsampled location depends on the state of its nearest known neighbours in axial directions.

Value

A data frame containing the simulation grid, the simulated random field, predicted values and the approximated probabilities is returned. Two extra columns respectively denoting the entropy and standardized entorpy are bindend to the data frame when argument entropy = TRUE.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Li, W. (2007) A Fixed-Path Markov Chain Algorithm for Conditional Simulation of Discrete Spatial Variables. *Mathematical Geology*, **39**(2), 159-176.

Li, W. (2007) Markov Chain Random Fields for Estimation of Categorical Variables. *Mathematical Geology*, **39**(June), 321-335.

Pickard, D. K. (1980) Unilateral Markov Fields. *Advances in Applied Probability*, **12**(3), 655-671.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

sim_ck, sim_ik, sim_mcs

Examples

```
data(ACM)

# Model parameters estimation for the
# multinomial categorical simulation
x <- multi_tpfit(ACM$MAT5, ACM[, 1:3])
```

```
# Generate the simulation grid
mygrid <- list()
mygrid$X <- seq(min(ACM$X), max(ACM$X), length = 20)
mygrid$Y <- seq(min(ACM$Y), max(ACM$Y), length = 20)
mygrid$Z <- -40 * 0:9 - 1
mygrid <- as.matrix(expand.grid(mygrid$X, mygrid$Y, mygrid$Z))

# Simulate the random field through
# the fixed path algorithm
myFixPathSim <- sim_path(x, ACM$MAT5, ACM[, 1:3], mygrid,
                         radius = 50, fixed = TRUE)

# Simulate the random field through
# the random path algorithm
myRndPathSim <- sim_path(x, ACM$MAT5, ACM[, 1:3], mygrid, radius = 50)
```

`summary.lengths` *Summarizing Stratum Lengths*

Description

The function summarizes the stratum lengths for each observed category.

Usage

```
## S3 method for class 'lengths'
summary(object, ..., zeros.rm = TRUE)
```

Arguments

`object`	an object of the class `lengths`, typically with the output of the function `getlen`.
`...`	further arguments passed to or from other methods.
`zeros.rm`	a logical values. If `FALSE`, summarizing statistics will be computed by including zero values. It is `TRUE` by default.

Value

An object of class `summary.lengths` containing the minimum, the first quartile, the median, the mean, the third quartile and the maximum of the stratum lengths for each observed category.

Author(s)

Luca Sartore <drwolf85@gmail.com>

See Also

`getlen`

Examples

```
data(ACM)
direction <- c(0,0,1)

# Compute the appertaining directional line for each location
loc.id <- which_lines(ACM[, 1:3], direction)

# Estimate stratum lengths
gl <- getlen(ACM$MAT3, ACM[, 1:3], loc.id, direction)

# Summarize the stratum lengths
sgl <- summary(gl)
```

---

tpfit *One-dimensional Model Parameters Estimation*

Description

The function estimates the model parameters of a 1-D continuous lag spatial Markov chain. Transition rates matrix along a user defined direction and proportions of categories are computed.

Usage

```
tpfit(data, coords, direction, method = "ml",
      tolerance = pi/8, max.it = 9000, mle = "avg", ...)
```

Arguments

`data`	a categorical data vector of length n.
`coords`	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
`direction`	a d-D numerical vector (or versor) which represents the chosen direction.
`method`	a character object specifying the method to estimate the transition rates. Possible choises are `"ml"` (by default) for the mean length method, `"ils"` for the iterated least squares and `"me"` for the maximum entropy method.
`tolerance`	a numerical value for the tolerance angle (in radians). It's `pi/8` by default.
`max.it`	a numerical value which denotes the maximum number of iterations to perform during the optimization phase. It is `9000` by default and used only when the method is `"me"`.
`mle`	a character value to pass to the function `mlen`. It is `"avg"` by default and not use when the method is `"ils"`.
`...`	other arguments to pass to the functions `tpfit_ml`, `tpfit_ils` or `tpfit_me`.

Details

A 1-D continuous-lag spatial Markov chain is probabilistic model which involves a transition rate matrix R computed for the direction ϕ. It defines the transition probability $\Pr(Z(s+h) = z_k | Z(s) = z_j)$ through the entry t_{jk} of the following matrix

$$T = \text{expm}(hR),$$

where h is a positive lag value.

Three methods are available to calculate entries of the transition rate matrix. The mean length method is performed by the use of the function `tpfit_ml`, the iterated least squares are applied through the function `tpfit_ils`, while the function `tpfit_me` implements the maximum entropy method.

Value

An object of the class `tpfit` is returned. The function `print.tpfit` is used to print the fitted model. The object is a list with the following components:

`coefficients`	the transition rates matrix computed for the user defined direction.
`prop`	a vector containing the proportions of each observed category.
`tolerance`	a numerical value which denotes the tolerance angle (in radians).

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

`predict.tpfit`, `print.tpfit`, `multi_tpfit`, `transiogram`

Examples

```
data(ACM)

# Estimate the parameters of a
# one-dimensional MC model
tpfit(ACM$MAT5, ACM[, 1:3], c(0, 0, 1))
```

`tpfit_ils` *Iterated Least Squares Method for One-dimensional Model Parameters Estimation*

Description

The function estimates the model parameters of a 1-D continuous lag spatial Markov chain by the use of the iterated least squares and the bound-constrained Lagrangian methods. Transition rates matrix along a user defined direction and proportions of categories are computed.

Usage

```
tpfit_ils(data, coords, direction, max.dist = Inf, mpoints = 20,
          tolerance = pi/8, q = 10, echo = FALSE, ..., tpfit)
```

Arguments

`data`	a categorical data vector of length n.
`coords`	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
`direction`	a d-D numerical vector (or versor) which represents the chosen direction.
`max.dist`	a numerical value which defines the maximum lag value. It's `Inf` by default.
`mpoints`	a numerical value which defines the number of lag intervals.
`tolerance`	a numerical value for the tolerance angle (in radians). It's `pi/8` by default.
`q`	a numerical value greater than one for a constant which controls the growth of the penalization term in the loss function. It is equal to `10` by default.
`echo`	a logical value; if `TRUE`, the function prints some information about the optimization. It is `FALSE` by default.
`...`	other arguments to pass to the function `nlminb`.
`tpfit`	an object `tpfit` to optimize. If missing, the algorithm starts with a null transition rates matrix.

Details

A 1-D continuous-lag spatial Markov chain is probabilistic model which involves a transition rate matrix R computed for the direction ϕ. It defines the transition probability $\Pr(Z(s+h) = z_k | Z(s) = z_j)$ through the entry t_{jk} of the following matrix

$$T = \text{expm}(hR),$$

where h is a positive lag value.

To calculate entries of the transition rate matrix, we need to minimize the discrepancies between the empirical transiogram (see `transiogram`) and the predicted transition probabilities.

By the use of the iterated least squares, the diagonal entries of R are constrained to be negative, while the off-diagonal transition rates are constrained to be positive. Further constraints are considered in order to obtain a proper transition rates matrix.

Value

An object of the class `tpfit` is returned. The function `print.tpfit` is used to print the fitted model. The object is a list with the following components:

`coefficients`	the transition rates matrix computed for the user defined direction.
`prop`	a vector containing the proportions of each observed category.
`tolerance`	a numerical value which denotes the tolerance angle (in radians).

Warning

If the process is not stationary, the optimization algorithm does not converge.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

`predict.tpfit`, `print.tpfit`, `multi_tpfit_ils`, `transiogram`

Examples

```
data(ACM)

# Estimate the parameters of a
# one-dimensional MC model
tpfit_ils(ACM$MAT3, ACM[, 1:3], c(0,0,1), 100)
```

---

tpfit_me *Maximum Entropy Method for One-dimensional Model Parameters Estimation*

Description

The function estimates the model parameters of a 1-D continuous lag spatial Markov chain by the use of the maximum entropy method. Transition rates matrix along a user defined direction and proportions of categories are computed.

Usage

```
tpfit_me(data, coords, direction, tolerance = pi/8,
         max.it = 9000, mle = "avg")
```

Arguments

`data`	a categorical data vector of length n.
`coords`	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
`direction`	a d-D numerical vector (or versor) which represents the chosen direction.
`tolerance`	a numerical value for the tolerance angle (in radians). It is `pi/8` by default.
`max.it`	a numerical value which denotes the maximum number of iterations to perform during the optimization phase. It is `9000` by default.
`mle`	a character value to pass to the function `mlen`. It is `"avg"` by default.

Details

A 1-D continuous-lag spatial Markov chain is probabilistic model which involves a transition rate matrix R computed for the direction ϕ. It defines the transition probability $\Pr(Z(s+h) = z_k | Z(s) = z_j)$ through the entry t_{jk} of the following matrix

$$T = \text{expm}(hR),$$

where h is a positive lag value.

To calculate entries of the transition rate matrix, we need to maximize the entropy of the transition probabilities of embedded occurrences along a given direction ϕ. The entropy is defined as

$$e = -\sum_{k}^{K} \sum_{j \neq k}^{K} \tau_{jk,\phi} \log \tau_{jk,\phi},$$

where $\tau_{jk,\phi}$ are transition probabilities of embedded occurrences. It is maximized by the use of the iterative proportion fitting method.

When some entries of the matrix R are not identifiable, it is suggested to vary the `tolerance` coefficient or to set the input argument `mle` to `"mlk"`.

Value

An object of the class `tpfit` is returned. The function `print.tpfit` is used to print the fitted model. The object is a list with the following components:

`coefficients`	the transition rates matrix computed for the user defined direction.
`prop`	a vector containing the proportions of each observed category.
`tolerance`	a numerical value which denotes the tolerance angle (in radians).

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

See Also

predict.tpfit, print.tpfit, multi_tpfit_me

Examples

```
data(ACM)

# Estimate the parameters of a
# one-dimensional MC model
tpfit_me(ACM$MAT5, ACM[, 1:3], c(0,0,1))
```

---

tpfit_ml *Mean Length Method for One-dimensional Model Parameters Estimation*

Description

The function estimates the model parameters of a 1-D continuous lag spatial Markov chain by the use of the mean length method. Transition rates matrix along a user defined direction and proportions of categories are computed.

Usage

```
tpfit_ml(data, coords, direction, tolerance = pi/8, mle = "avg")
```

Arguments

data	a categorical data vector of length n.
coords	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
direction	a d-D numerical vector (or versor) which represents the chosen direction.
tolerance	a numerical value for the tolerance angle (in radians). It's pi/8 by default.
mle	a character value to pass to the function mlen. It is "avg" by default.

Details

A 1-D continuous-lag spatial Markov chain is probabilistic model which involves a transition rate matrix R computed for the direction ϕ. It defines the transition probability $\Pr(Z(s+h) = z_k | Z(s) = z_j)$ through the entry t_{jk} of the following matrix

$$T = \text{expm}(hR),$$

where h is a positive lag value.

To calculate entries of the transition rate matrix, we need to compute the mean lengths and the embedded transition probabilities.

By the use of the mean lengths, diagonal entries of R are computed as

$$\hat{r}_{kk} = \frac{1}{\bar{L}_k},$$

where $\bar{L}_k$ is the mean length of the k-th category.

The off-diagonal transition rates of the matrix R are estimated by the use of embedded transition probabilities and mean lengths:

$$\hat{r}_{jk} = \frac{\pi_{jk}}{\bar{L}_k}, \quad \forall j \neq k,$$

where π_{jk} is a specific embedded transition probability.

When some entries of the matrix R are not identifiable, it is suggested to vary the `tolerance` coefficient or to set the input argument `mle` to `"mlk"`.

Value

An object of the class `tpfit` is returned. The function `print.tpfit` is used to print the fitted model. The object is a list with the following components:

`coefficients`	the transition rates matrix computed for the user defined direction.
`prop`	a vector containing the proportions of each observed category.
`tolerance`	a numerical value which denotes the tolerance angle (in radians).

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

`predict.tpfit`, `print.tpfit`, `multi_tpfit_ml`, `transiogram`

Examples

```
data(ACM)

# Estimate the parameters of a
# one-dimensional MC model
tpfit_ml(ACM$MAT5, ACM[, 1:3], c(0, 0, 1))
```

`transiogram` *Empirical Transition Probabilities Estimation for 1-D MC*

Description

The function estimates transition probabilities matrices for a 1-D continuous lag spatial Markov chain.

Usage

```
transiogram(data, coords, direction, max.dist = Inf,
            mpoints = 20, tolerance = pi / 8, reverse = FALSE)
```

Arguments

`data`	a categorical data vector of length n.
`coords`	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
`direction`	a d-D numerical vector (or versor) which represents the chosen direction.
`max.dist`	a numerical value which defines the maximum lag value. It's `Inf` by default.
`mpoints`	a numerical value which defines the number of lag intervals.
`tolerance`	a numerical value for the tolerance angle (in radians). It's `pi/8` by default.
`reverse`	a logical value. If `TRUE` the transition probabilities of the reversible chain are also computed. It's `FALSE` by default.

Details

Empirical probabilities are estimated by counting such pairs of observations which satisfy some properties, and by normalizing the result.

A generic pair of sample points s_i and s_j, where $i \neq j$, must satisfy the following properties:

- $\|s_i - s_j\| \in [a, a + \frac{m}{n}]$, where a is a non negative real value, while m denotes the maximum lag value (`max.dist`) and n is the number of lag intervals (`mpoints`).
- the lag vector $h = s_i - s_j$ must have the same direction of the vector ϕ (`direction`) with a certain angular `tolerance`.

Value

An object of the class `transiogram` is returned. The function `print.transiogram` is used to print computed probabilities. The object is a list with the following components:

`Tmat`	a 3-D array containing the probabilities.
`LOSE`	a 3-D array containing the standard error calculated for the log odds of the transition probabilities.
`lags`	a vector containing one-dimensional lags.
`type`	a character string which specifies that computed probabilities are empirical.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Carle, S. F., Fogg, G. E. (1997) Modelling Spatial Variability with One and Multidimensional Continuous-Lag Markov Chains. *Mathematical Geology*, **29**(7), 891-918.

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

predict.tpfit, predict.tpfit, plot.transiogram

Examples

```
data(ACM)

# Estimate empirical transition
# probabilities by points
transiogram(ACM$MAT3, ACM[, 1:3], c(0, 0, 1), 200, 5)
```

---

which_lines *Points Classification through Directional Lines*

Description

The function classifies points which appertain to a same directional line.

Usage

```
which_lines(coords, direction, tolerance = pi / 8)
```

Arguments

coords	an $n \times d$ matrix where each row denotes the d-D coordinates of data locations.
direction	a d-D numerical vector (or versor) which represents the chosen direction.
tolerance	a numerical value for the tolerance angle (in radians). It's pi/8 by default.

Details

The algorithm used by this function searches the nearest points to a directional line. The function classifies such pairs of points that have the minimum distance and the same direction of the vector ϕ.

This operation is done to order points, so that it's possible to compute mean lengths (mlen) and embedded transition probabilities (embed_MC).

Value

A numerical vector containing the line number for each point.

Author(s)

Luca Sartore <drwolf85@gmail.com>

References

Sartore, L. (2010) Geostatistical models for 3-D data. M.Phil. thesis, Ca' Foscari University of Venice.

See Also

embed_MC, mlen, getlen

Examples

```
data(ACM)

direction <- c(0,0,1)

loc.id <- which_lines(ACM[, 1:3], direction)
```

Index

∗ **attribute**
is.lengths, 16
is.multi_tpfit, 17
is.multi_transiogram, 18
is.pemt, 19
is.tpfit, 20
is.transiogram, 21
∗ **classif**
which_lines, 73
∗ **datasets**
ACM, 4
∗ **distribution**
boxplot.lengths, 5
contour.pemt, 6
density.lengths, 8
embed_MC, 9
hist.lengths, 12
image.multi_tpfit, 13
image.pemt, 15
mixplot, 22
multi_tpfit, 25
multi_tpfit_ils, 27
multi_tpfit_me, 29
multi_tpfit_ml, 31
pemt, 33
persp.multi_tpfit, 34
persp.pemt, 36
plot.density.lengths, 37
plot.hist.lengths, 39
plot.lengths, 40
plot.transiogram, 41
predict.multi_tpfit, 42
predict.tpfit, 44
print.density.lengths, 45
print.summary.lengths, 49
quench, 52
sim, 55
sim_ck, 57
sim_ik, 59
sim_mcs, 61
sim_path, 62
summary.lengths, 64
tpfit, 65
tpfit_ils, 67
tpfit_me, 68
tpfit_ml, 70
transiogram, 72
∗ **hplot**
boxplot.lengths, 5
contour.pemt, 6
hist.lengths, 12
image.multi_tpfit, 13
image.pemt, 15
mixplot, 22
persp.multi_tpfit, 34
persp.pemt, 36
plot.density.lengths, 37
plot.hist.lengths, 39
plot.lengths, 40
plot.transiogram, 41
∗ **models**
embed_MC, 9
multi_tpfit, 25
multi_tpfit_ils, 27
multi_tpfit_me, 29
multi_tpfit_ml, 31
predict.multi_tpfit, 42
predict.tpfit, 44
tpfit, 65
tpfit_ils, 67
tpfit_me, 68
tpfit_ml, 70
∗ **package**
spMC-package, 3
∗ **programming**
setCores, 54
∗ **spatial**
boxplot.lengths, 5

contour.pemt, 6
density.lengths, 8
embed_MC, 9
getlen, 11
hist.lengths, 12
image.multi_tpfit, 13
image.pemt, 15
is.lengths, 16
is.multi_tpfit, 17
is.multi_transiogram, 18
is.pemt, 19
is.tpfit, 20
is.transiogram, 21
mixplot, 22
mlen, 23
multi_tpfit, 25
multi_tpfit_ils, 27
multi_tpfit_me, 29
multi_tpfit_ml, 31
pemt, 33
persp.multi_tpfit, 34
persp.pemt, 36
plot.density.lengths, 37
plot.hist.lengths, 39
plot.lengths, 40
plot.transiogram, 41
predict.multi_tpfit, 42
predict.tpfit, 44
print.density.lengths, 45
print.lengths, 46
print.multi_tpfit, 47
print.multi_transiogram, 48
print.summary.lengths, 49
print.tpfit, 50
print.transiogram, 51
quench, 52
sim, 55
sim_ck, 57
sim_ik, 59
sim_mcs, 61
sim_path, 62
spMC-package, 3
summary.lengths, 64
tpfit, 65
tpfit_ils, 67
tpfit_me, 68
tpfit_ml, 70
transiogram, 72
which_lines, 73

ACM, 4

boxplot, *5, 6*, *40*
boxplot.lengths, 5, *40*

colors, *35, 36*
contour, *7, 8*, *13*, *15*
contour.pemt, 6

density.default, *8, 9*, *38*
density.lengths, 8, *13*, *38*, *45, 46*

embed_MC, 9, *73, 74*

getlen, *5, 6*, *8, 9*, 11, *12, 13*, *17*, *40*, *46, 47*, *50*, *64*, *74*

heat.colors, *35, 36*
hist, *12, 13*, *39*
hist.lengths, 12, *39*

image, *13–16*
image.multi_tpfit, *3*, 13, *16*, *23*, *26*, *29, 30*, *32*, *42, 43*
image.pemt, *3*, *8*, *14*, 15, *19*, *34*, *37*
is.lengths, 16
is.multi_tpfit, 17
is.multi_transiogram, 18
is.pemt, 19
is.tpfit, 20
is.transiogram, 21

mixplot, 22, *42*
mlen, *12*, 23, *65*, *69, 70*, *73, 74*
multi_tpfit, *3*, *13, 14*, *18*, 25, *35*, *37*, *43*, *45*, *47, 48*, *52*, *55*, *57*, *59*, *61, 62*, *66*
multi_tpfit_ils, *25*, 27, *68*
multi_tpfit_me, *25*, 29, *70*
multi_tpfit_ml, *25*, 31, *33, 34*, *71*

nlminb, *27*, *67*

par, *7*, *13*, *15*, *35, 36*
pemt, *3*, *14*, *19*, 33, *35–37*
persp, *35–37*
persp.multi_tpfit, 34, *35*, *37*
persp.pemt, *35*, 36
plot, *22, 23*, *38, 39*, *41, 42*
plot.density.lengths, *9*, *13*, 37, *46*

plot.hist.lengths, 39
plot.lengths, *6*, 40
plot.transiogram, *3*, *8*, *14*, *16*, *23*, *34, 35*, *37*, 41, *45*, *73*
predict.multi_tpfit, *3*, *10*, *18*, *26*, *29, 30*, *32*, 42, *48, 49*, *51*
predict.tpfit, *3*, *10*, *21–23*, *41, 42*, 44, *51*, *66*, *68*, *70, 71*, *73*
print.default, *46*
print.density.lengths, *9*, *38, 39*, 45
print.lengths, 46
print.multi_tpfit, *26*, *28–30*, *32*, *43*, 47
print.multi_transiogram, *43*, 48
print.summary.lengths, 49
print.tpfit, *45*, 50, *66*, *68–71*
print.transiogram, *45*, 51, *72*

quench, 52, *57*, *59*

setCores, 54
sim, *52*, 55
sim_ck, *3*, *53*, *55, 56*, 57, *60*, *62, 63*
sim_ik, *3*, *53*, *55, 56*, *58*, 59, *62, 63*
sim_mcs, *3*, *53*, *55, 56*, *58*, *60*, 61, *63*
sim_path, *3*, *53*, *55, 56*, *58*, *60*, 62, *62*
spMC *(*spMC-package*)*, 3
spMC-package, 3
summary.data.frame, *46*
summary.lengths, *49, 50*, 64

tpfit, *3*, *20*, *23*, *25, 26*, *29*, *31*, *42–45*, *50*, 65
tpfit_ils, *29*, *65, 66*, 67
tpfit_me, *30*, *65, 66*, 68
tpfit_ml, *32–34*, *65, 66*, 70
transiogram, *3*, *21–23*, *28, 29*, *41*, *43*, *45*, *51*, *66–68*, *71*, 72

which_lines, *9–12*, *24*, 73