sInitial

August 6, 2015

---

sInitial   *Function to initialise a sInit object given a topology and input data*

Description

sInitial is supposed to initialise an object of class "sInit" given a topology and input data. As a matter of fact, it initialises the codebook matrix (in input high-dimensional space). The return object inherits the topology information (i.e., a "sTopol" object from sTopology), along with initialised codebook matrix and method used.

Usage

```
sInitial(data, sTopol, init = c("linear", "uniform", "sample"))
```

Arguments

data	a data frame or matrix of input data
sTopol	an object of class "sTopol" (see sTopology)
init	an initialisation method. It can be one of "uniform", "sample" and "linear" initialisation methods

Value

an object of class "sInit", a list with following components:

- nHex: the total number of hexagons/rectanges in the grid
- xdim: x-dimension of the grid
- ydim: y-dimension of the grid
- lattice: the grid lattice
- shape: the grid shape
- coord: a matrix of nHex x 2, with each row corresponding to the coordinates of a hexagon/rectangle in the 2D map grid
- init: an initialisation method
- codebook: a codebook matrix of nHex x ncol(data), with each row corresponding to a prototype vector in input high-dimensional space
- call: the call that produced this result

1

Note

The initialisation methods include:

- "uniform": the codebook matrix is uniformly initialised via randomly taking any values within the interval [min, max] of each column of input data

- "sample": the codebook matrix is initialised via randomly sampling/selecting input data

- "linear": the codebook matrix is linearly initialised along the first two greatest eigenvectors of input data

See Also

sTopology

Examples

```
# 1) generate an iid normal random matrix of 100x10
data <- matrix( rnorm(100*10,mean=0,sd=1), nrow=100, ncol=10)

# 2) from this input matrix, determine nHex=5*sqrt(nrow(data))=50,
# but it returns nHex=61, via "sHexGrid(nHex=50)", to make sure a supra-hexagonal grid
sTopol <- sTopology(data=data, lattice="hexa", shape="suprahex")

# 3) initialise the codebook matrix using different mehtods
# 3a) using "uniform" method
sI_uniform <- sInitial(data=data, sTopol=sTopol, init="uniform")
# 3b) using "sample" method
# sI_sample <- sInitial(data=data, sTopol=sTopol, init="sample")
# 3c) using "linear" method
# sI_linear <- sInitial(data=data, sTopol=sTopol, init="linear")
```