Package ‘Umatrix’

February 12, 2020

Type Package

Title Visualization of Structures in High-Dimensional Data

Version 3.3

Date 2020-02-02

Author Florian Lerch[aut,cre], Michael Thrun[aut], Felix Pape[ctb], Raphael Paebst[ctb], Alfred Ultsch[aut]

Maintainer Florian Lerch <lerch@mathematik.uni-marburg.de>

Description By gaining the property of emergence through self-organization, the enhancement of SOMs (self organizing maps) is called Emergent SOM (ESOM). The result of the projection by ESOM is a grid of neurons which can be visualised as a three dimensional landscape in form of the Umatrix. Further details can be found in the referenced publications (see url). This package offers tools for calculating and visualising the ESOM as well as Umatrix, Pmatrix and UStarMatrix. All the functionality is also available through graphical user interfaces implemented in 'shiny'.

Imports Rcpp, ggplot2, shiny, shinyjs, reshape2, fields, plyr, png, tools, grid, abind, deldir, geometry, pdist, AdaptGauss, pracma

Suggests rgl

LinkingTo Rcpp

Depends R (>= 3.0)

License GPL-3


NeedsCompilation yes

Repository CRAN

Date/Publication 2020-02-12 12:50:02 UTC

R topics documented:

Umatrix-package ................................................................. 2
esomTrain ................................................................. 3
Hepta ................................................................. 4
Description

The ESOM (emergent self organizing map) is an improvement of the regular SOM (self organizing map) which allows for toroid grids of neurons and is intended to be used in combination with the Umatrix. The set of neurons is referred to as weights within this package, as they represent the values within the high dimensional space. The neuron with smallest distance to a datapoint is called a Bestmatch and can be considered as projection of said datapoint. As the Umatrix is usually toroid, it is drawn four consecutive times to remove border effects. An island, or Imx, is a filter mask, which cuts out a subset of the Umatrix, which shows every point only a single time while avoiding border effects cutting through potential clusters. Finally the Pmatrix shows the density structures within the grid, by a set radius. It can be combined with the Umatrix resulting in the UStarMatrix, which is therefore a combination of density based structures as well as clearly divided ones.

References


Train an ESOM (emergent self organizing map) and project data

Description

The ESOM (emergent self organizing map) algorithm as defined by [Ultsch 1999]. A set of weights (neurons) on a two-dimensional grid get trained to adapt the given datastructure. The weights will be used to project data on a two-dimensional space, by seeking the BestMatches for every datapoint.

Arguments

Data : Data that will be used for training and projection
Lines : Height of grid
Columns : Width of grid
Epochs : Number of Epochs the ESOM will run
Toroid : If TRUE, the grid will be toroid
NeighbourhoodFunction : Type of Neighbourhood; Possible values are: "cone", "mexicanhat" and "gauss"
StartLearningRate : Initial value for LearningRate
EndLearningRate : Final value for LearningRate
StartRadius : Start value for the Radius in which will be searched for neighbours
EndRadius : End value for the Radius in which will be searched for neighbours
NeighbourhoodCooling : Cooling method for radius; "linear" is the only available option at the moment
LearningRateCooling : Cooling method for LearningRate; "linear" is the only available option at the moment
shinyProgress : Generate progress output for shiny if Progress Object is given
ShiftToHighestDensity : If True, the Umatrix will be shifted so that the point with highest density will be at the center
InitMethod : name of the method that will be used to choose initializations Valid Inputs: "uni_min_max": uniform distribution with minimum and maximum from sampleData "norm_mean_2std": normal distribution based on mean and standard deviation of sampleData
Key : Vector of numeric keys matching the datapoints. Will be added to Bestmatches
UmatrixForEsom : If TRUE, Umatrix based on resulting ESOM is calculated and returned
Details

On a toroid grid, opposing borders are connected.

Value

List with

- BestMatches: BestMatches of datapoints
- Weights: Trained weights
- Lines: Height of grid
- Columns: Width of grid
- Toroid: TRUE if grid is a toroid

References


Examples

data('Hepta')
res=esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))

Description

Dataset with 7 easily separable classes.

Usage

data("Hepta")

Details

Size 212, Dimensions 3, stored in Hepta$Data
Classes 7, stored in Hepta$Cls

References

iClassification

Examples

data(Hepta)
str(Hepta)

---

iClassification  GUI for manual classification

Description

This tool is a 'shiny' GUI that visualizes a given Umatrix and allows the user to select areas and mark them as clusters.

Arguments

- **Umatrix**: Matrix of Umatrix Heights
- **BestMatches**: Array with positions of Bestmatches
- **Cls**: Classification of the Bestmatches
- **Imx**: Matrix of an island that will be cut out of the Umatrix
- **Toroid**: Are BestMatches placed on a toroid grid? TRUE by default

Value

A vector containing the selected class ids. The order is corresponding to the given Bestmatches

References


Examples

```r
## Not run:
data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
cls = iClassification(e$Umatrix, e$BestMatches)
## End(Not run)
```
Description
Trains the ESOM and shows the Umatrix.

Arguments
Data
Matrix of Data that will be used to learn. One DataPoint per row
BestMatches
Array with positions of Bestmatches
Cls
Classification of the Bestmatches as a vector
Key
Numeric vector of keys matching the Bestmatches
Toroid
Are BestMatches placed on a toroid grid? TRUE by default

Value
List with
Umatrix
matrix with height values of the umatrix
BestMatches
matrix containing the bestmatches
Lines
number of lines of the chosen ESOM
Columns
number of columns of the chosen ESOM
Epochs
number of epochs of the chosen ESOM
Weights
List of weights
Toroid
True if a toroid grid was used
EsomDetails
Further details describing the chosen ESOM parameters

References
Description

The toroid Umatrix is usually drawn 4 times, so that connected areas on borders can be seen as a whole. An island is a manual cutout of such a tiled visualization, that is selected such that all connected areas stay intact. This ‘shiny’ tool allows the user to do this manually.

Arguments

- **Umatrix**: Matrix of Umatrix Heights
- **BestMatches**: Array with positions of BestMatches
- **Cls**: Classification of the BestMatches

Value

Boolean Matrix that represents the island within the tiled Umatrix

References


Examples

```r
## Not run:
data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
Imx = iUmapIsland(e$Umatrix, e$BestMatches)
plotMatrix(e$Umatrix, e$BestMatches, Imx = Imx$Imx)
## End(Not run)
```

Description

Calculates the Ustarmatrix by combining a Umatrix with a Pmatrix.
plotMatrix

Arguments

Weights  Weights that were trained by the ESOM algorithm
Lines    Height of the used grid
Columns  Width of the used grid
Data     Matrix of Data that was used to train the ESOM. One datapoint per row
Imx      Island mask that will be cut out from displayed Umatrix
Cls      Classification of the Bestmatches
Toroid   Are weights placed on a toroid grid?

Value

Ustarmatrix  matrix with height values of the Ustarmatrix

References


plotMatrix

Description

Draws a plot based on given Umatrix or Pmatrix.

Arguments

Matrix    Umatrix or Pmatrix to be plotted
BestMatches Positions of Bestmatches to be plotted onto the Umatrix
Cls       Class identifier for the Bestmatches
ClsColors Vector of colors that will be used to colorize the different classes
ColorStyle If "Umatrix" the colors of a Umatrix (Blue -> Green -> Brown -> White) will be used; If "Pmatrix" the colors of a Pmatrix (White -> Yellow -> Red) will be used
Toroid    Should the Umatrix be drawn 4times?
BmSize    Integer between 0.1 and 5, magnification factor of the drawn BestMatch circles
DrawLegend If TRUE, a color legend will be drawn next to the plot
FixedRatio If TRUE, the plot will be drawn with a fixed ratio of x and y axis
CutoutPol Only draws the area within given polygon
Nrlevels  Number of height levels that will be used within the Umatrix
plotMatrix

- **TransparentContours**: Use half transparent contours. Looks better but is slow
- **Imx**: Mask to cut out an island. Every value should be either 1 (stays in) or 0 (gets cut out)
- **Clean**: If TRUE, axis, margins, ... surrounding the Umatrix image will be removed
- **RemoveOcean**: If TRUE, the surrounding blue area around an island will be reduced as much as possible (while still maintaining a rectangular form)
- **TransparentOcean**: If TRUE, the surrounding blue area around an island will be transparent
- **Title**: A title that will be drawn above the plot
- **BestMatchesLabels**: Vector of strings corresponding to the order of BestMatches which will be drawn on the plot as labels
- **BestMatchesShape**: Numeric value of Shape that will be used. Responds to the usual shapes of ggplot
- **MarkDuplicatedBestMatches**: If TRUE, BestMatches that are shown more than once within an island, will be marked
- **YellowCircle**: If TRUE, a yellow circle is drawn around Bestmatches to distinct them better from background

**Details**

The heightScale (nrlevels) is set at the proportion of the 1 percent quantile against the 99 percent quantile of the matrix values.

**Value**

A `ggplot` of a Matrix

**References**


**Examples**

data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
plotMatrix(e$Umatrix, e$BestMatches)
Description

Generates a Pmatrix based on the weights of an ESOM.

Arguments

Data A \([n,k]\) matrix containing the data
Weights Weights stored as a list in a 2D matrix
Lines Number of lines of the SOM that is described by weights
Columns Number of columns of the SOM that is described by weights
Radius The radius for measuring the density within the hypersphere
PlotIt If set the Pmatrix will also be plotted
Toroid Are BestMatches placed on a toroid grid? TRUE by default

Value

UstarMatrix

References


Examples

data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
Pmatrix = pmatrixForEsom(Hepta$Data,
e$Weights,
e$Lines,
e$Columns,
e$Toroid)
plotMatrix(Pmatrix, ColorStyle = "Pmatrix")
showMatrix3D

Description

Visualizes the matrix(Umatrix/Pmatrix) in an interactive window in 3D.

Arguments

- **Matrix**: Matrix to be plotted
- **BestMatches**: Positions of BestMatches to be plotted onto the matrix
- **Cls**: Class identifier for the BestMatch at the given point
- **Imx**: a mask (island) that will be used to cut out the Umatrix
- **Toroid**: Should the Matrix be drawn 4 times (in a toroid view)
- **HeightScale**: Optional. Scaling Factor for Mountain Height
- **BmSize**: Size of drawn BestMatches
- **RemoveOcean**: Remove as much area surrounding an island as possible
- **ColorStyle**: Either "Umatrix" or "Pmatrix" respectevily for their colors
- **ShowAxis**: Draw an axis arround the drawn matrix
- **SmoothSlope**: Try to increase the island size, to get smooth slopes around the island
- **ClsColors**: Vector of colors that will be used for classes
- **FileName**: Name for a stl file to write the Matrix to

Details

The heightScale is set at the proportion of the 1 percent quantile against the 99 percent quantile of the Matrix values.

References


Examples

```r
## Not run:
data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
showMatrix3D(e$Umatrix)
## End(Not run)
```
umatrixForEsom

**Description**

Calculate the Umatrix for given ESOM projection

**Arguments**

- **Weights**
  - Weights from which the Umatrix will be calculated
- **Lines**
  - Number of lines of the SOM that is described by weights
- **Columns**
  - Number of columns of the SOM that is described by weights
- **Toroid**
  - Boolean describing if the neural grid should be borderless

**Value**

Umatrix

**References**


**Examples**

data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
umatrix = umatrixForEsom(e$Weights,
                          Lines=e$Lines,
                          Columns=e$Columns,
                          Toroid=e$Toroid)
plotMatrix(umatrix, e$BestMatches)

ustarmatrixCalc

**Description**

The UStarMatrix is a combination of the Umatrix (average distance to neighbours) and Pmatrix (density in a point). It can be used to improve the Umatrix, if the dataset contains density based structures.

**Arguments**

- **Umatrix**
  - A given Umatrix
- **Pmatrix**
  - A density matrix
Value
UStarMatrix

References

Examples
```r
data("Hepta")
e = esomTrain(Hepta$Data, Key = 1:nrow(Hepta$Data))
Pmatrix = pmatrixForEsom(Hepta$Data,
e$Weights,
e$Lines,
e$Columns,
e$Toroid)
Ustarmatrix = ustarmatrixCalc(e$Umatrix, Pmatrix)
plotMatrix(Ustarmatrix, e$BestMatches)
```
### Index

*Topic **ESOM**
  - esomTrain, 3
  - Umatrix-package, 2
*Topic **High-Dimensional Data**
  - Umatrix-package, 2
*Topic **High-dimensional**
  - Umatrix-package, 2
*Topic **High-dimensional**
  - Umatrix-package, 2
*Topic **P-Matrix**
  - pmatrixForEsom, 10
*Topic **P-matrix**
  - pmatrixForEsom, 10
*Topic **Pmatrix**
  - pmatrixForEsom, 10
*Topic **Projection**
  - Umatrix-package, 2
*Topic **SOM**
  - esomTrain, 3
  - Umatrix-package, 2
*Topic **U*-Matrix**
  - ustarmatrixCalc, 12
*Topic **U*-matrix**
  - ustarmatrixCalc, 12
*Topic **Umatrix**
  - pmatrixForEsom, 10
*Topic **U-Matrix**
  - umatrixForEsom, 12
*Topic **U-matrix**
  - umatrixForEsom, 12
*Topic **Umatrix**
  - umatrixForEsom, 12
*Topic **Ustar-Matrix**
  - ustarmatrixCalc, 12
*Topic **Ustar-matrix**
  - ustarmatrixCalc, 12
*Topic **Visualization**
  - Umatrix-package, 2
*Topic **datasets Hepta FCPS**
  - Hepta, 4
*Topic **esom**
  - esomTrain, 3
*Topic **self organizing map**
  - esomTrain, 3
*Topic **som**
  - esomTrain, 3
  - esomTrain, 3
  - Hepta, 4
  - iClassification, 5
  - iEsomTrain, 6
  - iUmapIsland, 7
  - iUstarmatrix, 7
  - plotMatrix, 8
  - pmatrixForEsom, 10
  - showMatrix3D, 11
  - Umatrix-package, 2
  - umatrixForEsom, 12
  - ustarmatrixCalc, 12