Package ‘popsom’

December 20, 2021

Version 6.0

Title An Efficient Implementation of Kohonen's Self-Organizing Maps
(SOMs) with Starburst Visualizations

Imports fields, graphics, ggplot2, hash, stats, grDevices

Suggests testthat (>= 3.0.0)

Description Kohonen's self-organizing maps with a number of distinguishing features:
(1) An efficient, single threaded, stochastic training algorithm inspired by ideas from tensor algebra. Provides significant speedups over traditional single-threaded training algorithms. No special accelerator hardware required (see <doi:10.1007/978-3-030-01057-7_60>).
(2) Automatic centroid detection and visualization using starbursts.
(3) Two models of the data: (a) a self organizing map model, (b) a centroid based clustering model.
(4) A number of easily accessible quality metrics for the self organizing map and the centroid based cluster model (see <doi:10.1007/978-3-319-28518-4_4>).

License GPL-3

URL `https://github.com/lutzhamel/popsom`

BugReports `https://github.com/lutzhamel/popsom/issues`

Config/testthat/edition 3

NeedsCompilation yes

Author Lutz Hamel [aut, cre],
Benjamin Ott [aut],
Gregory Breard [aut],
Robert Tatoian [aut],
Michael Eiger [aut],
Vishakh Gopu [aut]

Maintainer Lutz Hamel <`lutzhamel@uri.edu`>

Repository CRAN

Date/Publication 2021-12-20 20:40:02 UTC

R topics documented:

`map.build` *Build Map*

Description

Construct a self-organizing map and return an object of class 'map'

Usage

```
map.build(data,labels=NULL,xdim=10,ydim=5,
          alpha=0.3,train=1000,normalize=FALSE,
          seed=NULL,minimal=FALSE)
```

Arguments

`data`	A dataframe where each row contains an unlabeled training instance.
`labels`	A vector or dataframe with one label for each observation in data.
`xdim`	The x-dimension of the map.
`ydim`	The y-dimension of the map.
`alpha`	The learning rate, should be a value greater than zero and less or equal to one.
`train`	The number of training iterations.
`normalize`	Boolean switch indicating whether or not to normalize the data.
`seed`	A seed value for repeatablity of random initialization and selection.
`minimal`	Boolean switch indicating whether to build a 'map.minimal' or 'map' object.

Details

The function 'map.build' constructs an object of type 'map'. The object contains two models: (1) A self-organizing map model expressed through its trained neurons and its quality of fit can be ascertained by the 'convergence' (see below). (2) A cluster model expressed by the discovered centroids. The quality of these models can be ascertained by the 'map convergence', 'within cluster sum of squares', and the 'between cluster sum of squares' (see below).

Value

An object of type ‘map’. The object has the following member fields:

data Data frame contining the possibly normalized training data.

labels Vector of labels, one for each observation in data or NULL if no labels were given.

xdim The x dimension of the neuron map.

ydim The y dimension of the neuron map.

alpha The given learning rate for the neural network.

train The training iterations applied to the neural network.

neurons A list of neurons for the network. The dimensionality of this data frame is the same as the training data. The following two formulas come in handy when working with the neural data. The first set of computations provide the x and y coordinate on the map of the neuron in row ‘rowix’ of the ‘neurons’ data frame,

```
x <- (rowix-1)%%map$xdim+1
y <- (rowix-1)%/%map$xdim+1
```

The second formula computes the row of the neuron in position (x,y) on the map,

```
rowix <- x+(y-1)*map$xdim
```

heat This is the representation of the map which is the basis for the ‘starburst’ plot.

fitted.obs List of indexes of the best matching neuron for each observation. Each index is an row index into the ‘neuron’ data frame.

centroids This is a data frame of (x,y)-locations where each cell points to the the (x,y)-location on the map where the corresponding centroid is located. Centroids point to themselves.

unique.centroids A vector of actual centroid (x,y)-locations on the map. Hint: to compute the number of clusters on the map take the length of this vector.

centroid.labels A data frame where the (x,y)-locations of actual centroids have a label associated with them. All other locations are NULL. If the training data is unlabeled then popsom invents a label for each centroid.

label.to.centroid A label-to-centroid lookup table (hash). A lookup in this table will return a list of indexes into the ‘unique.centroids’ table. Note: a label can be associated with multiple centroids.

centroid.obs A vector of lists of observations per centroid indexed by the centroid number from ‘unique.centroids’. The observations on the list are row numbers of the ‘data’ data frame.

convergence A quality measure of how well the map fits the training data.

wcss The average ‘within cluster sum of squares’. This is the average distance variance within the clusters of the underlying cluster model.

bcss The ‘between cluster sum of squares’. This is the distance variance between the cluster centroids of the underlying cluster model.

Note

If the ‘minimal’ switch is set to TRUE then a ‘map.minimal’ object is returned which only contains the trained neurons together with the training parameters. Observe that none of the POPSOM interface functions will work with this kind of object.

Note

If your training data is unlabeled popsom will automatically generate a label for each of the centroids it discovers.

Author(s)

Lutz Hamel, Benjamin Ott, Gregory Breard

References

VSOM: Efficient, Stochastic Self-Organizing Map Training, Lutz Hamel, Intelligent Systems Conference (IntelliSys) 2018, K. Arai et al. (Eds.): Intelligent Systems and Applications, Advances in Intelligent Systems and Computing 869, pp 805-821, Springer, 2018.

Self-Organizing Map Convergence, Robert Tatoian and Lutz Hamel. Proceedings of the 2016 International Conference on Data Mining (DMIN'16), pp92-98, July 25-28, 2016, Las Vegas, Nevada, USA, ISBN: 1-60132-431-6, CSREA Press.

Evaluating Self-Organizing Map Quality Measures as Convergence Criteria, Gregory Breard and Lutz Hamel, Proceedings of the 2018 International Conference on Data Science (ICDATA'18), Robert Stahlbock, Gary M. Weiss, Mahmoud Abou-Nasr (Eds.), ISBN: 1-60132-481-2, pp 86-92, CSREA Press, 2018.

SOM Quality Measures: An Efficient Statistical Approach, Lutz Hamel, Proceedings of the 11th International Workshop WSOM 2016, Houston, Texas USA, E. Merenyi et al. (eds.), Advances in Self-Organizing Maps and Learning Vector Quantization, Advances in Intelligent Systems and Computing 428, Springer, pp 49-59, DOI 10.1007/978-3-319-28518-4_4, 2016.

Examples

```
# training data
data(iris)
df <- subset(iris,select=-Species)
labels <- subset(iris,select=Species)

# build a map
m <- map.build(df,labels,xdim=15,ydim=10,train=10000,seed=42)

# look at the characteristics of the maps
map.summary(m)

# plot the map
map.starburst(m)
```

`map.convergence` *SOM Quality Assessment*

Description

Evaluate the quality of a SOM using embedding accuracy and estimated topographical accuracy.

Usage

```
map.convergence(map,conf.int=.95,k=50,verb=TRUE,ks=TRUE)
```

Arguments

`map`	an object of type 'map'.
`conf.int`	is the confidence interval of the quality assessment.
`k`	number of samples to use in the computation of the estimated topographical accuracy.
`verb`	if true reports the two convergence components separately, otherwise it will report a linear combination of the two indices.
`ks`	if true uses the Kolmogorov-Smirnov convergence test otherwise a convergence test based on variance and means is performed.

Value

A single value or a pair of values: 1) embedding accuracy 2) estimated topographic accuracy. The structure of the return value depends on the 'verb' switch.

Author(s)

Lutz Hamel

References

"SOM Quality Measures: A Statistical Approach," Lutz Hamel, WSOM16, 2016.

Examples

```
data(iris)

## set data frame and labels
df <- subset(iris,select=-Species)
labels <- subset(iris,select=Species)

## build a map
m <- map.build(df, labels, xdim=15, ydim=10, train=1000)

## map quality
map.convergence(m)
```

map.fitted *Fit Observations*

Description

Computes a vector of labels assigned to each of the observations in the training data through the constructed cluster model. If the training data is unlabeled then machine-generated labels are used.

Usage

```
map.fitted(map)
```

Arguments

`map` An object of type 'map'.

Value

A vector of predicted labels, one for each observations in the training data.

Author(s)

Lutz Hamel

Examples

```
data(iris)

df <- subset(iris,select=-Species)
labels <- subset(iris,select=Species)

m <- map.build(df,labels,xdim=15,ydim=10,train=10000)

map.fitted(m)
```

map.marginal *Plot Marginal Distribution*

Description

Generate a plot that shows the marginal probability distribution of the neurons and data.

Usage

```
map.marginal(map,marginal)
```

Arguments

`map`	An object of type 'map'.
`marginal`	The name of a training data dimension or index.

Author(s)

Lutz Hamel, Robert Tatoian

Examples

```
data(iris)

## set data frame and labels
df <- subset(iris,select=-Species)
labels <- subset(iris,select=Species)

## build a map
m <- map.build(df,labels,xdim=15,ydim=10,train=10000)

## display marginal distribution of dimension 1
map.marginal(m,1)
```

`map.position` *Compute Map Positions for Given Points*

Description

Compute the (x,y)-positions of points on the map.

Usage

```
map.position(map,points)
```

Arguments

`map`	An object of type 'map'.
`points`	A data frame of points to be mapped.

Value

A data frame with (x,y)-positions. The data frame has two columns:

x-dim The x-position of the corresponding point in the 'points' data frame.

y-dim The y-position of the corresponding point in the 'points' data frame.

Author(s)

Lutz Hamel

Examples

```
data(iris)

df <- subset(iris,select=-Species)
labels <- subset(iris,select=Species)

m <- map.build(df,labels,xdim=15,ydim=10,train=10000)

map.position(m,df)
```

map.predict *Compute Classification Labels for Given Points*

Description

Compute classification labels for points in a given data frame using the underlying clustering model. If the training data is unlabeled then machine-generated labels are used.

Usage

```
map.predict(map,points)
```

Arguments

`map`	An object of type 'map'.
`points`	A data frame of points to be classified.

Value

A data frame with classification results. The data frame has two columns:

Label The assigned label to the observation at the same row in the 'points' data frame.

Confidence A confidence value assigned to the label prediction.

Author(s)

Lutz Hamel

Examples

```
data(iris)

df <- subset(iris,select=-Species)
labels <- subset(iris,select=Species)

m <- map.build(df,labels,xdim=15,ydim=10,train=10000)

map.predict(m,df)
```

map.significance *Compute Significance of Features*

Description

Computes the relative significance of each feature and plots it.

Usage

```
map.significance(map,graphics=TRUE,feature.labels=TRUE)
```

Arguments

`map`	An object of type 'map'.
`graphics`	A switch that controls whether a plot is generated or not.
`feature.labels`	A switch to allow the plotting of feature names vs feature indices.

Value

If graphics=FALSE a vector containing the significance for each feature is returned.

Note

We use a Bayesian approach to compute the relative significance of features based on variance.

Author(s)

Lutz Hamel

References

Bayesian Probability Approach to Feature Significance for Infrared Spectra of Bacteria, Lutz Hamel, Chris W. Brown, Applied Spectroscopy, Volume 66, Number 1, 2012.

Examples

```
data(iris)

df <- subset(iris,select=-Species)
labels <- subset(iris,select=Species)

m <- map.build(df,labels,xdim=15,ydim=10,train=10000)

## show the relative feature significance for each feature
df <- data.frame(names(df),map.significance(m,graphics=FALSE))
names(df) <- c("Features","Significance")
df

## display the relative feature significance graphically
map.significance(m)
```

map.starburst *Generate Starburst For Map*

Description

Generate a starburst representation of the clusters on the heat map for the self-organizing map model.

Usage

```
map.starburst(map)
```

Arguments

map	An object of type 'map'

Author(s)

Lutz Hamel, Benjamin Ott, Gregory Breard, Robert Tatoian, Vishakh Gopu

References

Improved Interpretability of the Unified Distance Matrix with Connected Components, Lutz Hamel and Chris W. Brown. Proceeding of the 7th International Conference on Data Mining (DMIN'11), July 18-21, 2011, Las Vegas Nevada, USA, ISBN: 1-60132-168-6, pp338-343, CSREA Press, 2011.

Examples

```
data(iris)

df <- subset(iris,select=-Species)
labels <- subset(iris,select=Species)

m <- map.build(df,labels,xdim=15,ydim=10,train=100000)

map.starburst(m)
```

map.summary *Summary Object for Map*

Description

Generate a summary object for 'map' objects.

Usage

```
map.summary(map,verb=TRUE)
```

Arguments

`map` An object of type 'map'.

`verb` A switch controlling the output.

Value

An object of type 'summary.map' which contains two structures:

training.parameters A dataframe containing the parameters the map was trained with.

quality.assessments A dataframe containing the quality assessments of the map. In particular, it contains the 'convergence' of the map which is a linear combination of variance capture and topographic fidelity of the map. A value close to 1 means a converged map (for more details see the reference below). Furthermore, it contains the 'separation' of the clusters. This is computed by the formula,

```
 1 - wcss/bcss
```

In general, a value close to 1 means well separated clusters.

If 'verb' is TRUE the summar.map object will be formatted and printed to the screen, otherwise it will be returned as a data structure.

Author(s)

Lutz Hamel

References

Self-Organizing Map Convergence, Robert Tatoian and Lutz Hamel. Proceedings of the 2016 International Conference on Data Mining (DMIN'16), pp92-98, July 25-28, 2016, Las Vegas, Nevada, USA, ISBN: 1-60132-431-6, CSREA Press.

Examples

```
data(iris)

## set data frame and labels
df <- subset(iris,select=-Species)
labels <- subset(iris,select=Species)

## build a map
m <- map.build(df,labels,xdim=15,ydim=10,train=10000)

## compute a summary object and display it
s <- map.summary(m,verb=FALSE)
s
```

Index