

# Package ‘popsom’

June 3, 2019

**Version** 4.2.1

**Date** 2017-05-31

**Title** Functions for Constructing and Evaluating Self-Organizing Maps

**Maintainer** Lutz Hamel <hamel@cs.uri.edu>

**Imports** som,class,fields, graphics, ggplot2

**Description** State of the art functions for constructing and evaluating self-organizing maps.

**License** GPL

**URL** <http://homepage.cs.uri.edu/faculty/hamel/>

**NeedsCompilation** yes

**Author** Lutz Hamel [aut, cre],  
Benjamin Ott [aut],  
Gregory Breard [aut],  
Robert Tatroian [aut],  
Vishakh Gopu [aut]

**Repository** CRAN

**Date/Publication** 2019-06-03 10:13:10 UTC

## R topics documented:

map.build . . . . .	2
map.convergence . . . . .	3
map.embed . . . . .	4
map.marginal . . . . .	5
map.neuron . . . . .	6
map.projection . . . . .	7
map.significance . . . . .	7
map.starburst . . . . .	8
map.topo . . . . .	9

<b>Index</b>	<b>11</b>
--------------	-----------

---

`map.build`*Build Map*

---

**Description**

Constructs a SOM, returns an object of class 'map'.

**Usage**

```
map.build(data, labels=NULL, xdim=10, ydim=5, alpha=0.3, train=1000, algorithm="vsom")
```

**Arguments**

<code>data</code>	a dataframe where each row contains an unlabeled training instance.
<code>labels</code>	a vector or dataframe with one label for each observation in data.
<code>xdim</code>	the x-dimension of the map. (default=10)
<code>ydim</code>	the y-dimension of the map. (default=5)
<code>alpha</code>	the learning rate, should be a positive non-zero real number. (default=0.3)
<code>train</code>	the number of training iterations. (default=1000)
<code>algorithm</code>	training algorithm selection switch. (default="vsom")

**Value**

object of type 'map'.

**Note**

You have a choice of training algorithms:

- "vsom" - vectorized stochastic learning, this is a highly optimized version of stochastic training written in FORTRAN 9X.
- "som" - the traditional stochastic learning algorithm written in C++/TNT.
- "batchsom" - batch version of the SOM training algorithm, written in C.
- "experimental" - experimental implementation of vectorized stochastic learning implemented directly in R.

**Note**

If your training data does not have any labels you can construct a simple label vector as follows: `labels <- 1:nrow(training.data)`. If you let the labels default to the NULL value then no labels will be shown in the map visualization.

**Author(s)**

Lutz Hamel, Benjamin Ott, Gregory Breard

**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.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=FALSE, ks=FALSE)
```

**Arguments**

map	an object of type 'map'.
conf.int	is the confidence interval of the quality assessment (default 95%)
k	number of samples to use in the computation of the estimated topographical accuracy (default=50)
verb	if true reports the two convergence components separately, otherwise it will report a linear combination of the two indices (default=FALSE)
ks	if true uses the Kolmogorov-Smirnov convergence test otherwise a convergence test based on variance and means is performed (default=FALSE)

**Value**

A single value or a pair of values: 1) embedding accuracy 2) estimated topographic accuracy.

**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.embed

*Map Embedding Accuracy*

---

## Description

Evaluates how well a map models the underlying training data distribution.

## Usage

```
map.embed(map, conf.int = 0.95, verb=FALSE, ks=FALSE)
```

## Arguments

map	an object of type 'map'.
conf.int	the confidence interval of the embedding test (default 95 percent).
verb	a switch controlling the structure of the output value (default=FALSE)
ks	if true uses the Kolmogorov-Smirnov convergence test otherwise a convergence test based on variance and means is performed (default=FALSE)

## Value

The embedding accuracy of the map. If the switch verb=TRUE then a vector of the individual feature embedding accuracies are returned.

## Author(s)

Lutz Hamel, Benjamin Ott, Gregory Breard, Robert Tatioian

## References

"A Population Based Convergence Criterion for Self-Organizing Maps," Lutz Hamel and Benjamin Ott. Proceeding of the 2012 International Conference on Data Mining (DMIN'12), pp98-104, July 16-19, 2012, Las Vegas Nevada, USA.

**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)

## display the embedding accuracy of the map
map.embed(m)

## display the embedding accuracies of the individual features
data.frame(names(df),map.embed(m,verb=TRUE))
```

---

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                is the name of a training data frame 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=1000)

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

---

map.neuron	<i>Return a Map Neuron</i>
------------	----------------------------

---

### Description

Returns the contents of a neuron at (x,y) on the map as a vector.

### Usage

```
map.neuron(map, x, y)
```

### Arguments

map	an object of type 'map'.
x	map x-coordinate of neuron.
y	map y-coordinate of neuron.

### Value

A vector representing the neuron.

### Author(s)

Lutz Hamel

### 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)

## display the neuron at position (1,1)
map.neuron(m,1,1)
```

---

map.projection	<i>Map Projection</i>
----------------	-----------------------

---

**Description**

Prints the association of labels with map elements.

**Usage**

```
map.projection(map)
```

**Arguments**

map                    an object of type 'map'.

**Value**

a dataframe containing the projection onto the map for each training observation.

**Author(s)**

Lutz Hamel, Benjamin Ott, Gregory Breard

**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)

## display the label association for the map
map.projection(m)
```

---

map.significance	<i>Compute Significance Of Features</i>
------------------	---

---

**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, Benjamin Ott, Gregory Breard

**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)

## 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)

## show the relative feature significance for each feature
data.frame(names(df),map.significance(m,graphics=FALSE))

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

---

map.starburst

*Generate Starburst For Map*

---

**Description**

Computes and displays the starburst representation of clusters on a map.

**Usage**

```
map.starburst(map, explicit = FALSE, smoothing = 2, merge.clusters=TRUE, merge.range=.25)
```



**Arguments**

map	an object of type 'map'.
explicit	controls the shape of the connected components.
smoothing	controls the smoothing level of the map display (NULL, 0, >0).
merge.clusters	is a switch that controls if the starburst clusters are merged together
merge.range	is a range that is used as a percentage of a certain distance in the code to determine whether components are closer to their centroids or centroids closer to each other.

**Author(s)**

Lutz Hamel, Benjamin Ott, Gregory Breard, Robert Tatioian, 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)

## 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)

## display the starburst for the map
map.starburst(m)
```

---

map.topo

*Estimated Topographical Accuracy*

---

**Description**

Evaluate the topological quality of a SOM using the estimated topographical accuracy.

**Usage**

```
map.topo(map,k=50,conf.int = 0.95,verb=FALSE,interval=TRUE)
```

**Arguments**

map	an object of type 'map'.
k	number of samples to use in the computation of the estimated topographical accuracy (default=50)
conf.int	the confidence interval of the estimated topographical accuracy (default 95 percent).
verb	a switch controlling the structure of the output value (default=FALSE)
interval	a switch that controls whether the confidence interval is computed (default=TRUE)

**Value**

1) The value of the estimated topographical accuracy. 2) The low value of the confidence interval and the high value of the confidence interval 'conf.int' if interval=TRUE. 3) If verb=TRUE then map.accuracy will return a vector with the accuracies of the individual k samples.

**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)

## display estimated topographical accuracy of the map
map.topo(m)
```

# Index

map.build, 2  
map.convergence, 3  
map.embed, 4  
map.marginal, 5  
map.neuron, 6  
map.projection, 7  
map.significance, 7  
map.starburst, 8  
map.topo, 9