Package 'mmtsne'

July 28, 2017

Type Package

Title Multiple Maps t-SNE

Author Benjamin J. Radford

Maintainer Benjamin J. Radford <benjamin.radford@gmail.com>

Version 0.1.0

Description An implementation of multiple maps t-distributed stochastic neighbor embedding (t-SNE). Multiple maps t-SNE is a method for projecting high-dimensional data into several low-dimensional maps such that non-metric space properties are better preserved than they would be by a single map. Multiple maps t-SNE with only one map is equivalent to standard t-SNE. When projecting onto more than one map, multiple maps t-SNE estimates a set of latent weights that allow each point to contribute to one or more maps depending on similarity relationships in the original data. This implementation is a port of the original 'Matlab' library by Laurens van der Maaten.
See Van der Maaten and Hinton (2012) <doi:10.1007/s10994-011-5273-4>.
This material is based upon work supported by the United States Air Force and Defense Advanced Research Project Agency (DARPA) under Contract No. FA8750-17-C-0020.
Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force and Defense Advanced Research Projects Agency.
Distribution Statement A: Approved for Public Release; Distribution Unlimited.

License FreeBSD | file LICENSE

LazyData TRUE

RoxygenNote 6.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2017-07-28 09:09:42 UTC

R topics documented:

`hbeta` *Estimate perplexity and probability values*

Description

`hbeta` returns the perplexity and probability values for a row of data `D`.

Usage

```
hbeta(D, beta = 1)
```

Arguments

`D`	A distance vector.
`beta`	A constant scalar.

`mmtsne` *Multiple maps t-SNE*

Description

`mmtsne` estimates a multiple maps t-distributed stochastic neighbor embedding (multiple maps t-SNE) model.

Usage

```
mmtsne(X, no_maps = 1, no_dims = 2, perplexity = 30, max_iter = 500,
  momentum = 0.5, final_momentum = 0.8, mom_switch_iter = 250,
  eps = 1e-07)
```

Arguments

`X`	A dataframe or matrix of N rows and D columns.
`no_maps`	The number of maps (positive whole number) to be estimated.
`no_dims`	The number of dimensions per map. Typical values are 2 or 3.
`perplexity`	The target perplexity for probability matrix construction. Commonly recommended values range from 5 to 30. Perplexity roughly corresponds to the expected number of neighbors per data point.

`max_iter`	The number of iterations to run.
`momentum`	Constant scaling factor for update momentum in gradient descent algorithm.
`final_momentum`	Constant scaling factor for update momentum in gradient descent algorithm after the momentum switch point.
`mom_switch_iter`	The iteration at which momentum switches from `momentum` to `final_momentum`.
`eps`	A small positive value near zero.

Details

`mmtsne` is a wrapper that performs multiple maps t-SNE on an input dataset, `X`. The function will pre-process `X`, an N by D matrix or dataframe, then call `mmtsneP`. The pre-processing steps include calls to `x2p` and `p2sp` to convert `X` into an N by N symmetrical joint probability matrix.

The `mmtnseP` code is an almost direct port of the original multiple maps t-SNE Matlab code by van der Maaten and Hinton (2012). `mmtsne` estimates a multidimensional array of `N x no_dims x no_maps`. Each map is an `N x no_dims` matrix of estimated t-SNE coordinates. When `no_maps=1`, multiple maps t-SNE reduces to standard t-SNE.

Value

A list that includes the following objects:

Y An `N x no_dims x no_maps` array of predicted coordinates.

weights An `N x no_maps` matrix of unscaled weights. A high weight on entry i, j indicates a greater contribution of point i on map j.

proportions An `N x no_maps` matrix of scaled weights. A high weight on entry i, j indicates a greater contribution of point i on map j.

References

L.J.P. van der Maaten and G.E. Hinton. "Visualizing Non-Metric Similarities in Multiple Maps." *Machine Learning* 87(1):33-55, 2012. PDF.

Examples

```
# Load the iris dataset
data("iris")

# Estimate a mmtsne model with 2 maps, 2 dimensions each
model <- mmtsne(iris[,1:4], no_maps=2, max_iter=100)

# Plot the results side-by-side for inspection
# Points scaled by map proportion weights plus constant factor
par(mfrow=c(1,2))
plot(model$Y[,,1], col=iris$Species, cex=model$proportions[,1] + .2)
plot(model$Y[,,2], col=iris$Species, cex=model$proportions[,2] + .2)
par(mfrow=c(1,1))
```

`mmtsneP` *Multiple maps t-SNE with symmetric probability matrix*

Description

`mmtsneP` estimates a multiple maps t-distributed stochastic neighbor embedding (multiple maps t-SNE) model.

Usage

```
mmtsneP(P, no_maps, no_dims = 2, max_iter = 500, momentum = 0.5,
  final_momentum = 0.8, mom_switch_iter = 250, eps = 1e-07)
```

Arguments

`P`	An NxN symmetric joint probability distribution matrix. These can be constructed from an N by D matrix with `x2p` and `p2sp`. Alternatively, the wrapper function `mmtsne` will wrap the matrix construction and multiple maps t-SNE model estimation into a single step.
`no_maps`	The number of maps (positive whole number) to be estimated.
`no_dims`	The number of dimensions per map. Typical values are 2 or 3.
`max_iter`	The number of iterations to run.
`momentum`	Constant scaling factor for update momentum in gradient descent algorithm.
`final_momentum`	Constant scaling factor for update momentum in gradient descent algorithm after the momentum switch point.
`mom_switch_iter`	The iteration at which momentum switches from `momentum` to `final_momentum`.
`eps`	A small positive value near zero.

Details

This code is an almost direct port of the original multiple maps t-SNE Matlab code by van der Maaten and Hinton (2012). `mmtsne` estimates a multidimensional array of `N x no_dims x no_maps`. Each map is an `N x no_dims` matrix of estimated t-SNE coordinates. When `no_maps=1`, multiple maps t-SNE reduces to standard t-SNE.

Value

A list that includes the following objects:

Y An `N x no_dims x no_maps` array of predicted coordinates.

weights An `N x no_maps` matrix of unscaled weights. A high weight on entry i, j indicates a greater contribution of point i on map j.

proportions An `N x no_maps` matrix of scaled weights. A high weight on entry i, j indicates a greater contribution of point i on map j.

References

L.J.P. van der Maaten and G.E. Hinton. "Visualizing Non-Metric Similarities in Multiple Maps." *Machine Learning* 87(1):33-55, 2012. PDF.

Examples

```
# Load the iris dataset
data("iris")

# Produce a symmetric joint probability matrix
prob_matrix <- p2sp(x2p(as.matrix(iris[,1:4])))

# Estimate a mmtsne model with 2 maps, 2 dimensions each
model <- mmtsneP(prob_matrix, no_maps=2, max_iter=100)

# Plot the results side-by-side for inspection
# Points scaled by map proportion weights plus constant factor
par(mfrow=c(1,2))
plot(model$Y[,,1], col=iris$Species, cex=model$proportions[,1] + 0.2)
plot(model$Y[,,2], col=iris$Species, cex=model$proportions[,2] + 0.2)
par(mfrow=c(1,1))
```

`p2sp` *Probability matrix to symmetric probability matrix*

Description

`p2sp` returns a symmetrical pair-wise joint probability matrix given an input probability matrix *P*.

Usage

```
p2sp(P)
```

Arguments

`P`	An `N x N` probability matrix, like those produced by `x2p`

Value

An `N x N` symmetrical matrix of pair-wise probabilities.

x2p *Data to probability matrix*

Description

x2p returns a pair-wise conditional probability matrix given an input matrix *X*.

Usage

```
x2p(X, perplexity = 30, tol = 1e-05)
```

Arguments

X	A data matrix with N rows.
perplexity	The target perplexity. Values between 5 and 50 are generally considered appropriate. Loosely translates into the expected number of neighbors per point.
tol	A small positive value.

Details

This function is an almost direct port of the original Python implementation by van der Maaten and Hinton (2008). It uses a binary search to estimate probability values for all pairwise-elements of X. The conditional Gaussian distributions should all be of equal perplexity.

Value

An N x N matrix of pair-wise probabilities.

References

L.J.P. van der Maaten and G.E. Hinton. "Visualizing High-Dimensional Data Using t-SNE." *Journal of Machine Learning Research* 9(Nov):2579-2605, 2008. PDF.

Index