Package ‘tsne’

March 19, 2026

Type Package

Title T-Distributed Stochastic Neighbor Embedding for R (t-SNE)

Version 0.2-0

Date 2026-03-18

Description A ``pure R'' implementation of the t-SNE algorithm.

License GPL

LazyLoad yes

NeedsCompilation no

URL https://github.com/jdonaldson/rtsne/

BugReports https://github.com/jdonaldson/rtsne/issues

Author Justin Donaldson [aut, cre]

Maintainer Justin Donaldson <jdonaldson@gmail.com>

Repository CRAN

Date/Publication 2026-03-19 07:30:02 UTC

Contents

`tsne-package` *The tsne-package for multidimensional scaling*

Description

This package contains one function called tsne which contains all the functionality.

Details

Package:	tsne
Type:	Package
Version:	0.1
Date:	2010-02-19
License:	GPL
LazyLoad:	yes

Author(s)

Justin Donaldson https://github.com/jdonaldson/rtsne Maintainer: Justin Donaldson (jdonaldson@gmail.com)

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.

L.J.P. van der Maaten. Learning a Parametric Embedding by Preserving Local Structure. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics* (AISTATS), JMLR W&CP 5:384-391, 2009.

tsne *The t-SNE method for dimensionality reduction*

Description

Provides a simple function interface for specifying t-SNE dimensionality reduction on R matrices or "dist" objects.

Usage

```
tsne(X, initial_config = NULL, k = 2, initial_dims = 30, perplexity = 30,
     max_iter = 1000, min_cost = 0, epoch_callback = NULL, whiten = TRUE,
     epoch=100)
```

Arguments

`X`	The R matrix or "dist" object
`initial_config`	an argument providing a matrix specifying the initial embedding for X. See Details.
`k`	the dimension of the resulting embedding.
`initial_dims`	The number of dimensions to use in reduction method.
`perplexity`	Perplexity parameter. (optimal number of neighbors)

`max_iter` Maximum number of iterations to perform.

`min_cost` The minimum cost value (error) to halt iteration.

`epoch_callback` A callback function used after each epoch (an epoch here means a set number of iterations)

`whiten` A boolean value indicating whether the matrix data should be whitened.

`epoch` The number of iterations in between update messages.

Details

When the initial_config argument is specified, the algorithm will automatically enter the *final momentum* stage. This stage has less large scale adjustment to the embedding, and is intended for small scale tweaking of positioning. This can greatly speed up the generation of embeddings for various similar X datasets, while also preserving overall embedding orientation.

Value

An R object containing a *ydata* embedding matrix, as well as a the matrix of probabilities *P*

Author(s)

Justin Donaldson (jdonaldson@gmail.com)

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.

L.J.P. van der Maaten. Learning a Parametric Embedding by Preserving Local Structure. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics* (AISTATS), JMLR W&CP 5:384-391, 2009.

See Also

dist

Examples

```
## Not run:
colors = rainbow(length(unique(iris$Species)))
names(colors) = unique(iris$Species)
ecb = function(x,y){ plot(x,t='n'); text(x,labels=iris$Species, col=colors[iris$Species]) }
tsne_iris = tsne(iris[,1:4], epoch_callback = ecb, perplexity=50)

# compare to PCA
dev.new()
pca_iris = princomp(iris[,1:4])$scores[,1:2]
plot(pca_iris, t='n')
text(pca_iris, labels=iris$Species,col=colors[iris$Species])

## End(Not run)
```

Index