Package ‘PearsonICA’

January 20, 2025

Type Package

Title Independent Component Analysis using Score Functions from the Pearson System

Version 1.2-5

Date 2022-02-19

Author Juha Karvanen

Maintainer Juha Karvanen <juha.karvanen@iki.fi>

Description The Pearson-ICA algorithm is a mutual information-based method for blind separation of statistically independent source signals. It has been shown that the minimization of mutual information leads to iterative use of score functions, i.e. derivatives of log densities. The Pearson system allows adaptive modeling of score functions. The flexibility of the Pearson system makes it possible to model a wide range of source distributions including asymmetric distributions. The algorithm is designed especially for problems with asymmetric sources but it works for symmetric sources as well.

License AGPL-3

Imports grDevices, graphics, stats

NeedsCompilation no

Repository CRAN

Date/Publication 2022-02-21 09:00:28 UTC

Contents

PearsonICA *Pearson-ICA Algorithm for Independent Component Analysis (ICA)*

Description

The Pearson-ICA algorithm is a mutual information-based method for blind separation of statistically independent source signals. It has been shown that the minimization of mutual information leads to iterative use of score functions, i.e. derivatives of log densities. The Pearson system allows adaptive modeling of score functions. The flexibility of the Pearson system makes it possible to model a wide range of source distributions including asymmetric distributions. The algorithm is designed especially for problems with asymmetric sources but it works for symmetric sources as well.

Usage

```
PearsonICA(X, n.comp = 0, row.norm = FALSE, maxit = 200, tol = 1e-04,
           border.base = c(2.6, 4), border.slope = c(0, 1), verbose = FALSE,
   w.init = NULL, na.rm = FALSE, whitening.only = FALSE, PCA.only = FALSE)
```

Arguments

`X`	input data. Each column contains one signal.
`n.comp`	number of components to be extracted.
`row.norm`	a logical value indicating whether rows of the data matrix 'X' should be standardized beforehand.
`maxit`	maximum number of iterations to perform
`tol`	a positive scalar giving the tolerance at which the un-mixing matrix is considered to have converged.
`border.base`	intercept terms for the tanh boundaries. See details.
`border.slope`	slope terms for the tanh boundaries. See details.
`verbose`	a logical value indicating the level of output as the algorithm runs.
`w.init`	initial un-mixing matrix of dimension (n.comp,n.comp). If NULL (default) then a matrix of normal r.v.'s is used.
`na.rm`	should the rows with missing values be removed.
`whitening.only`	perform only whitening.
`PCA.only`	perform only principal component analysis.

Details

The data matrix X is considered to be a linear combination of statistically independent components, i.e. X = SA where A is a linear mixing and matrix the columns of S contain the independent components of which at most one has Gaussian distribution. The goal of ICA is to find a matrix W such that the output Y = XW is an estimate of possibly scaled and permutated source matrix S.

In order to extract the independent components/sources we search for a demixing matrix W that minimizes the mutual information of the sources. The minimization of mutual information leads to iterative use of score functions, i.e. derivatives of log densities. Pearson-ICA uses the Pearson system to model the score functions of the output Y. The parameters of the Pearson system are estimated by method of moments. To speed up the algorithm, tanh nonlinearity is used when the distribution is far from Gaussian. The parameters 'border.base' and 'border.slope' define the boundaries of the tanh area in the skewness-kurtosis plane. See Figure 2 in (Karvanen, Eriksson and Koivunen, 2000) for an illustration.

Value

A list containing the following components

`X`	input data
`whitemat`	whitening matrix
`W`	estimated demixing matrix
`A`	estimated mixing matrix
`S`	separated (estimated) source signals
`Xmu`	component means
`w.init`	starting value of W
`maxit`	maximum number of iterations allowed
`tol`	convergence limit
`it`	number of iterations used

Warning

The definition of W is different from that of `fastICA` algorithm (version 1.1-8).

Note

The R code is based on the MATLAB code by Juha Karvanen, Jan Eriksson and Visa Koivunen. Parts of the R code and documentation are taken from the `fastICA` R package.

Author(s)

Juha Karvanen

References

Karvanen J., Koivunen V. 2002. Blind separation methods based on Pearson system and its extensions, *Signal Processing* **82**(4), 663–673.

Karvanen J., Eriksson J.,Koivunen V. 2000, Pearson system based method for blind separation, *Proceedings of Second International Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, Helsinki, Finland, 585–590.

See Also

`PearsonICAdemo`

Examples

```
S<-matrix(runif(5000),1000,5);
X<-S+S[,c(2,3,4,5,1)];
icaresults<-PearsonICA(X,verbose=TRUE)
print(icaresults$A)
```

PearsonICAdemo *Demonstration of the Pearson-ICA Algorithm*

Description

Displays source signals, mixed signals and signals separated by Pearson-ICA.

Usage

```
PearsonICAdemo(numsig = 4, signal_length = 5000)
```

Arguments

`numsig`	number of source signals
`signal_length`	length of signal

Value

Displays a demonstration.

Note

The R code is based on the MATLAB code by Juha Karvanen, Jan Eriksson and Visa Koivunen.

Author(s)

Juha Karvanen

References

Karvanen J., Koivunen V. 2002. Blind separation methods based on Pearson system and its extensions, *Signal Processing* **82**(4), 663–673.

Karvanen J., Eriksson J.,Koivunen V. 2000, Pearson system based method for blind separation, *Proceedings of Second International Workshop on Independent Component Analysis and Blind Signal Separation (ICA2000)*, Helsinki, Finland, 585–590.

See Also

`PearsonICA`

Examples

```
PearsonICAdemo()
```

Index