

Package ‘SFAmix’

October 29, 2017

Type Package

Title SFAmix

Version 0.0-0

Date 2017-05-12

Author Chuan Gao <chuan.gao.cornell@gmail.com>

Maintainer Chuan Gao <chuan.gao.cornell@gmail.com>

Description SFAmix is a bayesian statistical method that decomposes a normally distributed matrix into the product of a sparse loading matrix, and a dense factor matrix. The method employs a three parameter beta distribution that act as a spike and slab prior to induce sparsity to the loading matrix. The method further consider the loading to be drawn from a mixture of sparse and dense component, which allows it to model the hidden confounding (in its dense component). The method is especially suitable for deriving context informed functional gene modules, where the context can be inferred from the factor matrix, and the sparse gene modules can be found on the corresponding loadings. Please refer to our paper for more details (<https://arxiv.org/abs/1310.4792>).

License GPL (>= 2)

Depends R (>= 2.14)

Collate SFAmixWrapper.R

NeedsCompilation yes

RoxygenNote 6.0.1

R topics documented:

gen_SFAmix_data	1
SFAmixR	2

Index	4
--------------	----------

gen_SFAmix_data	<i>Simulate matrix with dimension of 500 x 200, number of factors is set to 15, where 10 of them being sparse. The sparse loading matrix cotains mostly zeros, and random blocks of nonzero values generated from $N(0, std)$. The dense loading matrix is generated from $N(0, std)$, the factor matrix and the error matrix are generated from $N(0, 1)$.</i>
-----------------	--

Description

Simulate matrix with dimension of 500 x 200, number of factors is set to 15, where 10 of them being sparse. The sparse loading matrix contains mostly zeros, and random blocks of nonzero values generated from $N(0, \text{std})$. The dense loading matrix is generated from $N(0, \text{std})$, the factor matrix and the error matrix are generated from $N(0, 1)$.

Usage

```
gen_SFAmix_data(std = 1)
```

Arguments

std standard deviation for the normal distribution

Value

a list containing the following

- lams: the sparse loading matrix
- lamd: the dense loading matrix
- ex: the factors matrix
- y: the y matrix calculated as $y = \text{lam} * \text{ex} + \text{err}$

SFAmixR

An algorithm for decomposing a high dimensional matrix into the product of a sparse loading matrix, and a dense factor matrix.

Description

An algorithm for decomposing a high dimensional matrix into the product of a sparse loading matrix, and a dense factor matrix.

Usage

```
SFAmixR(y = y, nf = 50, a = 0.5, b = 0.5, itr = 500, out_itr = NULL,
        out_dir = NULL)
```

Arguments

y matrix to be decomposed, no missing values are allowed

nf the number of factors for the algorithm to start with, will be shrunk to a smaller number reflecting the number of factors needed to explain the variance, default to 50

a parameter one for the three parameter beta distribution, default to 0.5 to recapitulate horseshoe

b parameter two for the three parameter beta distribution, default to 0.5 to recapitulate horseshoe

itr The maximum number of iterations the algorithm is allowed to run, default to 500

out_itr	An iteration number at which the algorithm will output results into a specified directory (see below)
out_dir	The directory where temporary results at specified iteration (see above) are stored

Value

lam: the sparse loading matrix

ex: the factor matrix

z: a vector indicating whether the corresponding loading is sparse (value of 1)

nf: the number of factors learned by the model

exx: the expected value of the covariance matrix, $E(XX^T)$

Author(s)

Chuan Gao <chuan.gao.cornell@gmail.com>

References

<https://arxiv.org/abs/1310.4792>

Examples

```
library(SFAmix)
## simulate data
data = gen_SFAmix_data(std=2)
## run algorithm on the simulated data
result = SFAmixR(data$y,nf=50,a=0.5,b=0.5,itr=1000)
## calculate a correlation matrix of the estimated loading matrix
## and the true loading matrix. Ideally, there should be one and
## only one big correlation value for a given row and column of the
## correlation matrix
cor.est.real = cor(result$lam[,result$z==1],data$lams)
## visualize the correlation matrix
image(cor.est.real)
```

Index

gen_SFamix_data, [1](#)

SFamixR, [2](#)