Package ‘TensorComplete’

May 18, 2021

Type Package

Title Tensor Noise Reduction and Completion Methods

Version 0.1.0

Author Anonymous

Maintainer Anonymous <anonymous@anonymous.com>

Imports pracma, methods, utils, tensorregress, MASS

Description Efficient algorithms for nonparametric tensor completion via sign series. This package provides the nonparametric completion tool for estimating tensor signals from noisy, possibly incomplete observations. The algorithms employ the alternating optimization. The detailed algorithm description can be found in the main paper.

License GPL(>=2)

Encoding UTF-8

RoxygenNote 7.1.1

R topics documented:

`Altopt` *Alternating optimization of the weighted classification loss*

Description

Optimize the weighted classification loss given a weight tensor, an observed data tensor, and a large margin loss. This function is used as a subroutine in the main function `fit_nonparaT`.

Usage

```
Altopt(Ybar,W,r,type = c("logistic","hinge"),start = "linear")
```

Arguments

`Ybar` A given (possibly noisy and incomplete) data tensor.

`W` A weight tensor used in the weighted classification loss.

`r` A rank to be fitted (CP rank).

`type` A large margin loss to be used. Logistic or hinge loss is available.

`start` Choice of initialization method. Use random initialization if `start` = "random"; Use the initialization based on low rank approximation if `start` = "linear". Linear initialization is default.

Value

The returned object is a list of components.

`binary_obj` - Trajectory of binary loss values over iterations.

`obj` - Trajectory of weighted classification loss values over iterations.

`iter` - The number of iterations.

`error` - Trajectory of errors over iterations.

`fitted` - A tensor that optimizes the weighted classification loss.

Examples

```
library(tensorregress)
indices = c(2,3,4)
noise = rand_tensor(indices)@data
Theta = array(runif(prod(indices),min=-3,max = 3),indices)

# The signal plus noise model
Y = Theta + noise

# Optimize the weighted classification for given a sign tensor sign(Y) and a weight tensor abs(Y)
result = Altopt(sign(Y),abs(Y),r = 3,type = "hinge",start = "linear")
signTheta = sign(result$fitted)
```

fit_continuous_cp *Signal tensor estimation from a noisy and incomplete data tensor based on CP low rank tensor method.*

Description

Estimate a signal tensor from a noisy and incomplete data tensor using CP low rank tensor method.

Usage

```
fit_continuous_cp(data,r)
```

Arguments

`data` A given (possibly noisy and incomplete) data tensor.

`r` A rank to be fitted (CP rank).

Value

The returned object is a list of components.

`est` - An estimated signal tensor based on CP low rank tensor method.

`U` - A list of factor matrices.

`lambda` - A vector of tensor singular values.

Examples

```
library(tensorregress)
indices = c(2,3,4)
noise = rand_tensor(indices)@data
Theta = array(runif(prod(indices),min=-3,max = 3),indices)

# The signal plus noise model
Y = Theta + noise

# Estimate Theta from CP low rank tensor method
hatTheta = fit_continuous_cp(Y,3)
print(hatTheta$est)
```

`fit_nonparaT` *Main function for nonparametric tensor estimation and completion based on low sign rank model.*

Description

Estimate a signal tensor from a noisy and incomplete data tensor using nonparametric tensor method via sign series.

Usage

```
fit_nonparaT(Y,truer,H,Lmin,Lmax,option = 2)
```

Arguments

`Y`	A given (possibly noisy and incomplete) data tensor. The function allows both continuous- and binary-valued tensors. Missing value should be encoded as `NA`.
`truer`	Sign rank of the signal tensor.
`H`	Resolution parameter.
`Lmin`	Minimum value of the signal tensor (or minimum value of the tensor Y).
`Lmax`	Maximum value of the signal tensor (or maximum value of the tensor Y).
`option`	A large margin loss to be used. Use logistic loss if `option` = 1, hinge loss if `option` = 2. Hinge loss is default.

Value

The returned object is a list of components.

`fitted` - A series of optimizers that minimize the weighted classification loss at each level.

`est` - An estimated signal tensor based on nonparametic tensor method via sign series.

Examples

```
library(tensorregress)
indices = c(2,2,2)
noise = rand_tensor(indices)@data
Theta = array(runif(prod(indices),min=-2,max = 2),indices)

# The signal plus noise model
Y = Theta + noise

# Estimate Theta from nonparametic completion method via sign series
hatTheta = fit_nonparaT(Y,truer = 2,H = 2,Lmin = -2,Lmax = 2, option =2)
print(hatTheta$est)
```

Index