

# Package ‘tensorsign’

February 2, 2021

**Type** Package

**Title** Nonparametric Tensor Completion Via Sign Series

**Version** 0.1.0

**Author** Chanwoo Lee <chanwoo.lee@wisc.edu>, Miaoyan Wang <miaoyan.wang@wisc.edu>

**Maintainer** Chanwoo Lee <chanwoo.lee@wisc.edu>

**Imports** rTensor,  
Matrix,  
pracma

**Description** Efficient algorithm for nonparametric tensor completion ~~method~~ via sign series. The algorithm employs the alternating optimization approach to solve the weighted classification problem. The detailed algorithm description can be found in Lee and Wang (2021) <arXiv:2102.00384>.

**URL** <https://arxiv.org/abs/2102.00384>

**License** GPL(>=2)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

## R topics documented:

Alt . . . . .	1
fit_continuous . . . . .	2
SignT . . . . .	3
<b>Index</b>	<b>5</b>

---

Alt	<i>Alternating optimization of the weighted classification loss</i>
-----	---------------------------------------------------------------------

---

## Description

Optimize the weighted classification loss given a weight tensor, an observed data tensor, and a large margin loss.

## Usage

```
Alt(Ybar,W,r,type = c("logistic","hinge"),start = "random")
```

pay attention to the order. If parentheses are \*part of\* the sentence, the sentence should read well even without the parentheses. *fit\_continuous*

### Arguments

Ybar A given ~~data tensor~~ (possibly noisy and incomplete). data tensor  
W A weight tensor ~~of~~ the weighted classification loss. used in  
r ~~A rank to be fitted.~~  
type tensor rank to be fitted 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

of  
The returned value is a list ~~with~~ components.  
trajectory of binary loss values over iterations.  
binary\_obj ~~A series of binary loss values at each iteration.~~  
obj A series of the weighted classification loss values at each iteration.  
iter The number of iterations.  
error A series of errors at each iteration.  
fitted A tensor that optimizes the weighted classification loss.

### References

Lee, C., & Wang, M. (2021). Beyond the Signs: Nonparametric Tensor Completion via Sign Series. *arXiv preprint arXiv:2102.00384*. `a series of` allows non-ordered list.

### Examples

I intentionally use ``a series of sign tensors`` but not ``a sequence of sign tensors`` in our paper — think about why

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

In your case, "a sequence of", or ``trajector``

```
# The signal plus noise model
Y = Theta + noise
```

given a sign tensor

```
# Optimize the weighted classification for a given sign tensor sign(Y) and a weight tensor abs(Y)
result = Alt(sign(Y),abs(Y),r = 2,type = "logistic",start = "random")
signTheta = sign(result$fitted)
```

Revise the code to allow r >= d\_min.  
Current random initialization seems to require r <= min d.  
in the final documentation, use r=3 and start = "linear".

### Signal tensor estimation

---

fit\_continuous ~~Estimating a signal tensor~~ 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(data,r)
```

**Arguments**

data A given data tensor (possibly noisy and incomplete).  
 r ~~A~~ rank of the signal tensor.

**Value**

object

The returned ~~value~~ is a list with 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(rTensor)
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(Y,3)
print(hatTheta$est)
```

**Signal tensor estimation**

SignT

~~Estimation of a signal tensor~~ from a noisy and incomplete data tensor based on nonparametric tensor method via sign series.

**Description**

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

**Usage**

```
SignT(Y,truer,H,Lmin,Lmax,option = 1)
```

**Arguments**

Y A given data tensor (possibly noisy and incomplete).  
 truer ~~A~~ sign rank of the signal tensor.  
 H ~~A~~ resolution parameter.  
 Lmin ~~A~~ minimum value of the signal tensor (or a minimum value of the tensor Y).  
 Lmax ~~A~~ maximum value of the signal tensor (or a 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.

**Value** **object**

The returned **value** is a list with components.

**fitted** A series of optimizers that minimize the weighted classification loss at **each** ~~a given level~~  $\pi$ .

**est** An estimated signal tensor based on nonparametric tensor method via sign series.

**References**

Lee, C., & Wang, M. (2021). Beyond the Signs: Nonparametric Tensor Completion via Sign Series. *arXiv preprint arXiv:2102.00384*.

**Examples**

```
library(rTensor)
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 nonparametric completion method via sign series
hatTheta = SignT(Y,truer = 3,H = 5,Lmin = -3,Lmax = 3, option =1)
print(hatTheta$est)
```

# Index

Alt, [1](#)

fit\_continuous, [2](#)

SignT, [3](#)