Package ‘SmoothTensor’

November 4, 2021

Type Package

Title A collection of smooth tensor estimation methods

Version 0.1.0

Author Anonymous authors

Maintainer Anonymous <anonymous@anonymous.com>

Imports methods, Matrix, rTensor

Description Efficient algorithms for smooth tensor estimation with unknown permutation.

License GPL-3

Depends R (>= 3.5.0)

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

R topics documented:

Borda_count 1
LSE 2
ltns 4
mode_info 4
simulation 5
simulation_asym 6
simulation_bin 7
Spectral 8

Index **9**

`Borda_count` *Borda count algorithm for nonparametric tensor estimation with unknown permutation.*

Description

Estimate a signal tensor and permutation from a noisy and incomplete data tensor using Borda count estimation method.

Usage

```
Borda_count(A, l, kvec, sym = FALSE)
```

Arguments

`A`	A given (possibly noisy and incomplete) data tensor. Missing value should be encoded as `NA`.
`l`	Degree of polynomial approximation.
`kvec`	A vector of the number of groups for each mode.
`sym`	Boolean variables representing symmetricity of the signal tensor. Non-symmetric tensor (`sym = FALSE`) is default.

Value

The returned object is a list of components.

`Theta` - An estimated signal tensor based on Borda count estimation.

`permutation` - An estimated permutation based on Borda count estimation.

References

Anonymous authors. Smooth tensor estimation with unknown permutations. 2021.

Examples

```
# Generate the noisy observation from smooth tensor and permutation
d = 20
sim1 = simulation(d,mode = 1)
signal_T = sim1$signal
observe_T = sim1$observe
permutation = sim1$permutation

# Estimate signal tensor and permutation
kvec = c(3,3,3)
result = Borda_count(observe_T,2,kvec,sym = TRUE)

# Calculate MSE
hatTheta = result$Theta
mean((hatTheta-signal_T)^2)
```

LSE *The least squares estimation for nonparametric tensor estimation with unknown permutation.*

Description

Estimate a permuted signal tensor from a noisy data tensor based on the least squares estimation with constant block approximation.

Usage

```
LSE(A, kvec, sym = FALSE, mode = 3)
```

Arguments

`A` A given noisy data tensor.

`kvec` A vector of the number of groups for each mode.

`sym` Boolean variables representing symmetricity of the signal tensor. Non-symmetric tensor (`sym = FALSE`) is default.

`mode` An integer from 1 to 3 representing a type of methods for estimating the clustering functions. Higher-order spectral clustering method is default.
`mode = 1`: k-means algorithm applied on unfolded matrices.
`mode = 2`: k-means algorithm for community detection in stocahstic block model (only availble on binary observation).
`mode = 3`: higher-order spectral clustering algorithm.

Value

An estimated permuted signal tensor based on the least squares estimation.

References

Gao, C., Y. Lu, and H. H. Zhou. Rate-optimal graphon estimation. The Annals of Statistics, 2015.
Balasubramanian, K. Nonparametric modeling of higher-order interactions via hypergraphons. Journal of Machine Learning Research, 2021.
Han, R., Y. Luo, M. Wang, and A. R. Zhang. Exact clustering in tensor block model: Statistical optimality and computational limit. arXiv:2012.09996, 2020.

Examples

```
# Generate the noisy observation from smooth tensor and permutation
d = 20
sim1 = simulation(d, mode = 1)
signal_T = sim1$signal
observe_T = sim1$observe
permutation = sim1$permutation
psignal_T = signal_T[permutation,permutation,permutation]

# Estimate permuted signal tensor
kvec = c(10,10,10)
hatpTheta = LSE(observe_T,kvec,sym = TRUE)

# Calculate MSE
mean((hatpTheta-psignal_T)^2)
```

ltns *Chicago crime tensor dataset*

Description

Chicago crime dataset consists of crime counts reported in the city of Chicago, ranging from January 1st, 2001 to December 11th, 2017.

Usage

```
ltns
```

Format

An order-3 tensor with entries representing the log counts of crimes from 24 hours, 77 community areas, and 32 crime types.

Source

http://frostt.io/tensors/chicago-crime/

mode_info *A list of mode information of the Chicago crime tensor dataset*

Description

A list of mode information of order-3 tensor dataset ltns.

Usage

```
mode_info
```

Format

A list consisting of crime areas, crime hours, and crime types:

hour_map	24 hours of crimes
area_map	77 areas of crimes
crimetype_map	32 types of crimes

Source

http://frostt.io/tensors/chicago-crime/

simulation *Generate a symmetric tensor observation from the smooth signal tensor, Gaussian noise tensor, and permutation.*

Description

Generate a symmetric tensor observation from the smooth signal tensor, Gaussian noise tensor, and permutation. Users can select one of 5 different smooth signal tensors generated from functions specified in Table 4 of the reference given below.

Usage

```
simulation(d, mode = 1, sigma = 0.5, signal_level=5)
```

Arguments

`d`	Dimension of a tensor to be generated.
`mode`	An integer from 1 to 5 corresponding to models specified. Default model is 1.
`sigma`	Standard deviation of the Gaussian noise tensor. Default value is 0.5.
`signal_level`	A scale of the magnitude of the signal tensor to be generated.

Value

The returned object is a list of components.

`signal` - A true signal tensor generated from a function specified.

`observe` - A noisy observation generated from the smooth signal tensor, Gaussian noise tensor, and permutation.

`permutation` - A true permutation.

References

Anonymous authors. Smooth tensor estimation with unknown permutations. 2021.

Examples

```
d = 20
# Generate 20 by 20 by 20 observed tesnor generated from model 1
sim1 = simulation(d,mode = 1)
observed_tensor = sim1$observe
signal_tensor = sim1$signal
permutation = sim1$permutation
```

simulation_asym — *Generate a non-symmetric tensor observation from the smooth signal tensor, Gaussian noise tensor, and permutation.*

Description

Generate a non-symmetric tensor observation from the smooth signal tensor, Gaussian noise tensor, and permutation. Users can select one of 5 different smooth signal tensors generated from functions specified in Table 5 of the reference given below.

Usage

```
simulation_asym(d, mode = 1, sigma = 0.5, signal_level=5)
```

Arguments

`d`	A vector of dimensions of a tensor to be generated.
`mode`	An integer from 1 to 5 corresponding to models specified. Default model is 1.
`sigma`	Standard deviation of the Gaussian noise tensor. Default value is 0.5.
`signal_level`	A scale of the magnitude of the signal tensor to be generated.

Value

The returned object is a list of components.

`signal` - A true non-symmetric signal tensor generated from a function specified.

`observe` - A noisy observation generated from the smooth signal tensor, Gaussian noise tensor, and permutation.

`permutation` - A list of true permutation for each mode.

References

Anonymous authors. Smooth tensor estimation with unknown permutations. 2021.

Examples

```
d = c(10,20,30)
# Generate 10 by 20 by 30 observed tesnor generated from model 1
sim1 = simulation_asym(d,mode = 1)
observed_tensor = sim1$observe
signal_tensor = sim1$signal
permutation = sim1$permutation
```

simulation_bin *Generate a symmetric binary tensor from the probability tensor and permutation.*

Description

Generate a symmetric binary tensor from the probability tensor and permutation. Users can select one of 5 different smooth probability tensor generated from functions specified in Table 4 of the reference given below.

Usage

```
simulation_bin(d, mode = 1)
```

Arguments

d	Dimension of a tensor to be generated.
mode	An integer from 1 to 5 corresponding to models specified. Default model is 1.

Value

The returned object is a list of components.

`signal` - A true probability tensor generated from a function specified.

`observe` - A binary tensor generated by Bernoulli trials given the probability tensor and permutation.

`permutation` - A true permutation.

References

Anonymous authors. Smooth tensor estimation with unknown permutations. 2021.

Examples

```
d = 20
# Generate 20 by 20 by 20 binary-valued tensor generated from model 1
sim1 = simulation_bin(d, mode = 1)
observed_tensor = sim1$observe
signal_tensor = sim1$signal
permutation = sim1$permutation
```

Spectral *Spectral method for nonparametric tensor estimation with unknown permutation.*

Description

Estimate a permuted signal tensor from a noisy data tensor using spectral method, which performs universal singualr value thresholding on the unfolded tensor.

Usage

```
Spectral(A, row_idx, col_idx, threshold = NULL)
```

Arguments

A	A given noisy data tensor.
row_idx	The indices of the modes that map onto the row space
col_idx	The indices of the modes that map onto the column space
threshold	A threshold to disregard singular values. Default value is the square root of unfolded matrix dimension.

Value

An estimated permuted signal tensor based on Spectral method.

References

Xu, J. Rates of convergence of spectral methods for graphon estimation. International Conference on Machine Learning, 2018.
Anonymous authors. Smooth tensor estimation with unknown permutations. 2021.

Examples

```
# Generate the noisy observation from smooth tensor and permutation
d = 20
sim1 = simulation(d,mode = 1)
signal_T = sim1$signal
observe_T = sim1$observe
permutation = sim1$permutation
psignal_T = signal_T[permutation,permutation,permutation]

# Estimate permuted signal tensor
hatpTheta = Spectral(observe_T,1,c(2,3))

# Calculate MSE
mean((hatpTheta-psignal_T)^2)
```

Index

∗ **datasets**
    ltns, 4
    mode_info, 4

Borda_count, 1

LSE, 2
ltns, 4

mode_info, 4

simulation, 5
simulation_asym, 6
simulation_bin, 7
Spectral, 8