Package ‘treelet’

February 20, 2015

Type Package

Title An Adaptive Multi-Scale Basis for High-Dimensional, Sparse and Unordered Data

Version 1.1

Date 2015-02-10

Author Di Liu and Trent Gaugler

Maintainer Trent Gaugler <gauglert@lafayette.edu>

Description Treelets provides a novel construction of multi-scale bases that extends wavelets to non-smooth signals. It returns a multi-scale orthonormal basis, where the final computed basis functions are supported on nested clusters in a hierarchical tree. Both the tree and the basis, which are constructed simultaneously, reflect the internal structure of the data.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation no

Repository CRAN

Date/Publication 2015-02-11 00:37:29

R topics documented:

`treelet-package` *Treelets - an adaptive multi-scale basis for high-dimensional,sparse and unordered data*

Description

Treelets provides a novel construction of multi-scale bases that extends wavelets to non-smooth signals. It returns a multi-scale orthonormal basis, where the final computed basis functions are supported on nested clusters in a hierarchical tree. Both the tree and the basis, which are constructed simultaneously, reflect the internal structure of the data.

Details

Package:	treelet
Type:	Package
Version:	1.1
Date:	2015-02-10
License:	GPL (>= 2)

Author(s)

Di Liu and Trent Gaugler

Maintainer: Trent Gaugler <gauglert@lafayette.edu>

`Ahat` *Estimated additive genetic relationship matrix for 50 simulated individuals*

Description

Using the simulated data from the family pedigrees in the Crossett et al arXiv paper, five families were randomly selected, and 10 individuals from the 15 related individuals were randomly selected independently in each family. The $\hat{A}$ matrix was then calculated via the GCTA software of Yang et al. In the resulting 50 x 50 relationship matrix, element $\hat{A}_{ij}$ represents the estimated kinship coefficient between individuals i and j.

Usage

```
data(Ahat)
```

Source

arXiv:1208.2253v1 [stat.AP]

References

arXiv:1208.2253v1 [stat.AP]

Lee, AB, Nadler, B, Wasserman, L (2008). Treelets - an adaptive multi-scale basis for sparse unordered data. The Annals of Applied Statistics 2: 435-471. `http://www.stat.cmu.edu/~annlee/AOAS137.pdf`

Yang J, Lee SH, Goddard ME and Visscher PM. GCTA: a tool for Genome-wide Complex Trait Analysis. Am J Hum Genet. 2011 Jan 88(1): 76-82. [PubMed ID: 21167468]

`Build_JTree` *Constructs the hierarchical tree for the Treelet algorithm*

Description

Returns information on the construction of the treelet hierarchical tree on which the basis functions are supported.

Usage

```
Build_JTree(C, cc, maxlev, whichsave)
```

Arguments

`C` the covariance matrix of the data. For example, if using this function on genetics data to improve estimates of heritability, as in the Crossett et al arXiv paper, this argument will be the estimated additive genetic relationship matrix $\hat{A}$.

`cc` the correlation matrix of the data.

`maxlev` the maximum height of the tree. This must be an integer between 1 and `nrow(X)`-1.

`whichsave` a vector containing the levels of the tree, specified as integers between 1 and `maxlev`, for which you want to save the basis functions and the covariance matrix.

Value

a list with components

`Zpos` A matrix of dimension `maxlev` x 2. Each row records which two nodes/clusters of the tree were combined at each step in its construction.

`T` This is a list with `maxlev` elements, where each element is a 2x2 Jacobi rotation matrix for each step of the treelet algorithm.

PCidx	A matrix of dimension maxlev x 2, where each row is a permutation of $(1, 2)$ indicating which of the two nodes/clusters merged at that step is the sum variable (value of 1) and which is the difference (value of 2).
all_nodes	A matrix of dimension maxlev x nrow(X) giving node/cluster labels at each step of the treelet algorithm. A label of zero indicates a node/cluster that was merged with another node/cluster and was the difference variable.
TreeCovs	This is a list with maxlev elements. Only those elements that are specified in the whichsave argument will be non-null entries in the list. For the non-null entries, this is the covariance matrix calculated at that level of the tree. The covariances in this matrix are those between the weights (orthogonal projections onto local basis vectors) in the basis expansion of the data vector.

Author(s)

Trent Gaugler <gauglert@lafayette.edu>

References

arXiv:1208.2253v1 [stat.AP]

Lee, AB, Nadler, B, Wasserman, L (2008). Treelets - an adaptive multi-scale basis for sparse unordered data. The Annals of Applied Statistics 2: 435-471. http://www.stat.cmu.edu/~annlee/AOAS137.pdf

See Also

Run_JTree, JTree_Basis, TCS

JTree_Basis *Constructs the orthonormal basis for the Treelet algorithm*

Description

Returns information on the construction of the treelet orthonormal basis at each level of the tree up to the maximum tree height specified.

Usage

```
JTree_Basis(Zpos, T, PCidx, maxlev, all_nodes, whichsave)
```

Arguments

Zpos	A matrix of dimension maxlev x 2. Each row records which two nodes/clusters of the tree were combined at each step in its construction.
T	This is a list with maxlev elements, where each element is a 2x2 Jacobi rotation matrix for each step of the treelet algorithm.

`PCidx` A matrix of dimension `maxlev` x 2, where each row is a permutation of $(1, 2)$ indicating which of the two nodes/clusters merged at that step is the sum variable (value of 1) and which is the difference (value of 2).

`maxlev` The maximum height of the tree. This must be an integer between 1 and `nrow(X)`-1.

`all_nodes` A matrix of dimension `maxlev` x `nrow(X)` giving node/cluster labels at each step of the treelet algorithm. A label of zero indicates a node/cluster that was merged with another node/cluster and was the difference variable.

`whichsave` A vector containing the levels of the tree, specified as integers between 1 and `maxlev`, for which you want to save the basis functions and the covariance matrix.

Details

Note that with the exception of `maxlev` and `whichsave`, the arguments provided to this function are outputs from the `Build_JTree` function. This function is therefore not intended to be used as a stand-alone function, but in conjunction with `Build_JTree`. The wrapper function `Run_JTree` performs this connection.

Value

`basis` This is a list with `maxlev` elements. Only those elements that are specified in the `whichsave` argument will be non-null entries in the list. For the non-null entries, this is the orthonormal treelet basis calculated at that level of the tree.

Author(s)

Trent Gaugler <`gauglert@lafayette.edu`>

References

arXiv:1208.2253v1 [stat.AP]

Lee, AB, Nadler, B, Wasserman, L (2008). Treelets - an adaptive multi-scale basis for sparse unordered data. The Annals of Applied Statistics 2: 435-471. `http://www.stat.cmu.edu/~annlee/AOAS137.pdf`

See Also

`Build_JTree`, `Run_JTree`, `TCS`

Run_JTree *Treelet basis/hierarchical tree construction*

Description

Returns information on the simultaneous construction of the treelet orthonormal basis and hierarchical tree, including which nodes were merged at each step and the basis at each specified step of the construction.

Usage

```
Run_JTree(X, maxlev, whichsave)
```

Arguments

`X` the covariance matrix of the data. For example, if using this function on genetics data to improve estimates of heritability, as in the Crossett et al arXiv paper, this argument will be the estimated additive genetic relationship matrix $\hat{A}$.

`maxlev` the maximum height of the tree. This must be an integer between 1 and `nrow(X)`-1.

`whichsave` a vector containing the levels of the tree, specified as integers between 1 and `maxlev`, for which you want to save the basis functions and the covariance matrix.

Details

This function serves as a wrapper for the functions `Build_JTree` and `JTree_Basis`, which build the hierarchical tree and calculate the basis and covariance matrix at each level, respectively.

Value

a list with components

`basis` This is a list with `maxlev` elements. Only those elements that are specified in the `whichsave` argument will be non-null entries in the list. For the non-null entries, this is the orthonormal treelet basis calculated at that level of the tree.

`Zpos` A matrix of dimension `maxlev` x 2. Each row records which two nodes/clusters of the tree were combined at each step in its construction.

`T` This is a list with `maxlev` elements, where each element is a 2x2 Jacobi rotation matrix for each step of the treelet algorithm.

`PCidx` A matrix of dimension `maxlev` x 2, where each row is a permutation of $(1, 2)$ indicating which of the two nodes/clusters merged at that step is the sum variable (value of 1) and which is the difference (value of 2).

`all_nodes` A matrix of dimension `maxlev` x `nrow(X)` giving node/cluster labels at each step of the treelet algorithm. A label of zero indicates a node/cluster that was merged with another node/cluster and was the difference variable.

`TreeCovs` This is a list with `maxlev` elements. Only those elements that are specified in the `whichsave` argument will be non-null entries in the list. For the non-null entries, this is the covariance matrix calculated at that level of the tree. The covariances in this matrix are those between the weights (orthogonal projections onto local basis vectors) in the basis expansion of the data vector.

Author(s)

Trent Gaugler `<gauglert@lafayette.edu>`

References

arXiv:1208.2253v1 [stat.AP]

Lee, AB, Nadler, B, Wasserman, L (2008). Treelets - an adaptive multi-scale basis for sparse unordered data. The Annals of Applied Statistics 2: 435-471. `http://www.stat.cmu.edu/~annlee/AOAS137.pdf`

See Also

`Build_JTree`, `JTree_Basis`, `TCS`

Examples

```
data(Ahat)
out=Run_JTree(Ahat,49,49)
#The information in out$Zpos[1,] and out$all_nodes[1,]
#both show which two individuals were the first merged
#in the tree.  The remaining rows give information
#on subsequent merges in the tree.

basis=out$basis[[49]]
cov=out$TreeCovs[[49]]
temp=basis
#This is how you can use the basis and cov output
#to reconstruct the estimated relationship matrix.
#See how close temp and the original Ahat are:
Ahat1=round(Ahat,14)
temp1=round(temp,14)
sum(Ahat1!=temp1)
#In this example, we do start seeing discrepancies in the 15th digit and beyond.
```

TCS *Treelet Covariance Smoothing*

Description

This function thresholds values in the treelet estimated covariance and returns a smoothed estimate of a covariance matrix.

Usage

```
TCS(basis, cov, lambda)
```

Arguments

`basis` the orthonormal treelet basis calculated at a specific level ℓ of the tree.

`cov` the corresponding covariance matrix calculated at level ℓ of the tree. The covariances in this matrix are those between the weights (orthogonal projections onto local basis vectors) in the basis expansion of the data vector.

`lambda` a positive thresholding coefficient. Any element of the matrix `cov` that is less than this coefficient in absolute value will be set to zero.

Details

This function implements the TCS method presented in the Crossett et al arXiv paper. The arguments `basis` and `cov` should be obtained from the `Run_JTree` function. The `TCS` function is written so that it does not calculate the treelet basis within the function but asks for it as an argument so that the subsampling method presented in the arXiv paper, or another method to obtain a reasonable value of lambda, can be implemented.

Value

`smooth` the smoothed estimate of the covariance matrix.

Author(s)

Trent Gaugler <`gauglert@lafayette.edu`>

References

arXiv:1208.2253v1 [stat.AP]

Lee, AB, Nadler, B, Wasserman, L (2008). Treelets - an adaptive multi-scale basis for sparse unordered data. The Annals of Applied Statistics 2: 435-471. `http://www.stat.cmu.edu/~annlee/AOAS137.pdf`

See Also

`Build_JTree`, `JTree_Basis`, `Run_JTree`

Examples

```
data(Ahat)
out=Run_JTree(Ahat,49,49)
basis=out$basis[[49]]
cov=out$TreeCovs[[49]]

temp=TCS(basis,cov,.04)
#The value .04 above is arbitrary, and the user
#should carefully select this value.  One approach
```

```
#is the subsampling method outlined in the Crossett et al
#arXiv paper.  The value in 'temp' is the smoothed estimate
#of the relationship matrix.
```

Index