Package: LPGraph (via r-universe)

September 18, 2024

Type Package

Title Nonparametric Smoothing of Laplacian Graph Spectra

Version 2.1

Date 2020-01-29

Author Subhadeep Mukhopadhyay, Kaijun Wang

Maintainer Kaijun Wang <kaijun.wang@temple.edu>

Description A nonparametric method to approximate Laplacian graph spectra of a network with ordered vertices. This provides a computationally efficient algorithm for obtaining an accurate and smooth estimate of the graph Laplacian basis. The approximation results can then be used for tasks like change point detection, k-sample testing, and so on. The primary reference is Mukhopadhyay, S. and Wang, K. (2018, Technical Report).

Depends R (>= 3.5.0), stats, car, PMA

License GPL-2

NeedsCompilation no

Repository CRAN

Date/Publication 2020-01-30 20:20:02 UTC

Contents

LPGraph-package 2
LP.basis 2
LP.struct.test 3
LPSpectral 5
senate 6

Index **8**

LPGraph-package *Nonparametric Smoothing of Laplacian Graph Spectra*

Description

This package provides an algorithm to approximate and smooth the Laplacian graph spectra of a network with ordered vertices using LP-nonparametric basis. The underlying theory yields a reduced order (compressive) characterization of spectral graph properties.

Author(s)

Mukhopadhyay, S. and Wang, K.

Maintainer: Kaijun Wang <kaijun.wang@temple.edu>

References

Mukhopadhyay, S. and Wang, K. (2018), "Graph Spectral Compression via Smoothing".

Mukhopadhyay, S. (2017+), "Unified Statistical Theory of Spectral Graph Analysis".

Mukhopadhyay, S. and Parzen, E. (2014), "LP Approach to Statistical Modeling", arXiv:1405.2601.

LP.basis *Computes LP basis function of a discrete distribution*

Description

This function computes `m` LP basis functions for the given discrete distribution `p.dist`

Usage

```
LP.basis(p.dist, m)
```

Arguments

`p.dist`	A vector of n probabilities of a discrete distribution
`m`	An integer denoting the number of required LP basis functions

Value

A matrix of dimension $n \times m$.

Author(s)

Mukhopadhyay, S. and Wang, K.

References

Mukhopadhyay, S. and Parzen, E. (2014), "LP Approach to Statistical Modeling", arXiv:1405.2601.

Examples

```
  ##1.toy example:
  ##simulate a two sample locational difference normal data:
   X1<-matrix(rnorm(250,mean=0,sd=1),10,25)
   X2<-matrix(rnorm(250,mean=0.5,sd=1),10,25)
   X<-rbind(X1,X2)
  ## Adjacency matrix:
   dmat<-dist(X)
   W   <-exp(-as.matrix(dmat)^2/(2*quantile(dmat,.5)^2))
  ## getting the basis
   pp<- rowSums(W)/sum(W)
   T<-LP.basis(pp,m=4)
  #plot the j-th LP basis for the two sample data (here we use j=1).
   j=1
   plot(cumsum(pp),T[,j],type='s',xlab='',ylab='')

  ##2.Senate data
  ## Not run:
   data(senate)
   attach(senate)
  #create W matrix: (long computation)
   require(psych)
   W <- matrix(0,nrow(X),nrow(X))
   for(i in 1:(nrow(X)-1)){
 for(j in (i+1):nrow(X)) {
  W[i,j] <- psych::phi(table(X[i,],X[j,]))
 }
   }
   W = W + t(W)
   diag(W)<-0
  #getting the basis:
   pp<- rowSums(W)/sum(W)
   T<-LP.basis(pp,m=4)
  #plot the j-th LP basis for senate data (here we use j=1).
   j=1
   plot(cumsum(pp),T[,j],type='s',xlab='',ylab='')

## End(Not run)
```

`LP.struct.test` *Detection of structures in an ordered-network.*

Description

Given adjacency matrix `W`, this function perform a graph based test to determine whether there are different communities present in a graph of ordered vertices.

Usage

```
LP.struct.test(W, m = NULL, n.iter = 50)
```

Arguments

W	A n-by-n weighted-adjacency matrix.
m	Number of LP-nonparametric basis used for generating the test statistic, set to NULL to use original Laplacian.
n.iter	Iterations used for small sample correction, default is 50.

Value

A list containing the following items:

stat	The test statistic, which asymptotically follows a normal distribution with mean and variance mentioned in the reference.
pval	P-value for the test, small p-value means different communities may be present.

Author(s)

Mukhopadhyay, S. and Wang, K.

References

Mukhopadhyay, S. and Wang, K. (2018), "Graph Spectral Compression via Smoothing".

Examples

```
##1.example: null case
##simulate a normal data with mean 0 and variance 1:
 X  <-matrix(rnorm(500,mean=0,sd=1),20,25)
## Generate adjacency matrix:
 dmat<-dist(X)
 W   <-exp(-as.matrix(dmat)^2/(2*quantile(dmat,.5)^2))
## test of structure:
 h0.test<-LP.struct.test(W, m = 4 , n.iter = 50)
###extract p-value:
 h0.test$pval

##2.example: two sample location alternative
##simulate a two sample locational difference normal data:
 X1<-matrix(rnorm(250,mean=0,sd=1),10,25)
 X2<-matrix(rnorm(250,mean=0.5,sd=1),10,25)
 X<-rbind(X1,X2)
## Generate adjacency matrix:
 dmat<-dist(X)
 W   <-exp(-as.matrix(dmat)^2/(2*quantile(dmat,.5)^2))
## test of structure:
 h1.test<-LP.struct.test(W, m = 4 , n.iter = 50)
###extract p-value:
 h1.test$pval
```

LPSpectral *Nonparametric smooth approximation of the Laplacian graph spectra*

Description

This function provides nonparametric smooth approximation of the Laplacian graph spectra given a weighted-adjacency matrix W.

Usage

```
LPSpectral(W, k, m=8,sparse=TRUE)
```

Arguments

W	A n-by-n weighted-adjacency matrix.
k	Number of approximated singular vectors and singular values to return, where $k \leq m$.
m	Number of LP-nonparametric basis used for approximation, where $m \leq n$. By default, $m = 8$.
sparse	Set to TRUE to make coefficients for LP basis sparse, thus allowing for further smoothing.

Value

A list containing the following items:

LP	m-by-m LP Spectral graph matrix.
Phi	A n-by-k matrix of LP-approximated singular vectors.
sval	A vector of length k containing top k approximated singular values.

Author(s)

Mukhopadhyay, S. and Wang, K.

References

Mukhopadhyay, S. and Wang, K. (2018), "Graph Spectral Compression via Smoothing".

Examples

```
##1.toy example:
##simulate a two sample locational difference normal data:
 X1<-matrix(rnorm(250,mean=0,sd=1),10,25)
 X2<-matrix(rnorm(250,mean=0.5,sd=1),10,25)
 X<-rbind(X1,X2)
## Adjacency matrix:
 dmat<-dist(X)
```

```
  W <-exp(-as.matrix(dmat)^2/(2*quantile(dmat,.5)^2))
 ## Obtain top 10 approximated nontrivial singular values:
  data_sval<-LPSpectral(W, k=10)$sval
## Obtain approximated singular vector corresponding to the top nontrivial singular value:
  data_phi1<-LPSpectral(W, k=1)$Phi
 ## plot the results:
  par(mfrow=c(1,2))
  plot(data_sval,type='b')
  plot(data_phi1)

 ##2.Senate Data
## Not run:
  data(senate)
  attach(senate)
 ##creating W (long computation)
  require(psych)
  W <- matrix(0,nrow(X),nrow(X))
  for(i in 1:(nrow(X)-1)){
for(j in (i+1):nrow(X)) {
 W[i,j] <- psych::phi(table(X[i,],X[j,]))
}
  }
  W = W + t(W)
  diag(W)<-0
 ## Obtain top 10 approximated nontrivial singular values:
  senate_sval<-LPSpectral(W, k=10, m=15)$sval
## Obtain approximated singular vector corresponding to the top nontrivial singular value:
  senate_phi1<-LPSpectral(W, k=1, m=15)$Phi
 ## plot the results:
  par(mfrow=c(1,2))
  plot(senate_sval,type='b')
  plot(senate_phi1)

## End(Not run)
```

`senate` *Senate Vote Data*

Description

The senate vote data contain 2508 observations ordered by time from 1989 to 2001, with a dimension of 100. The change point occurred on January 17th, 1995, at the beginning of the tenure of the 104th Congress when the Republican Party captured the US House of Representatives for the first time after 1956.

Usage

```
data("senate")
```

Format

A list containing the following items:

`year`: A vector of length 2508, time labels.

`X`: A matrix of dimension 2508 by 100, original data of senate votes.

Details

To generate the weighted adjacency matrix `W`, one simple take the matrix `X` and Compute: $W_{ij} = \phi(X_i, X_j) = \frac{p_{00} - p_{+0}p_{0+}}{\sqrt{p_{0+}p_{1+}p_{+0}p_{+1}}}$ Where $p_{rs} = \sum_i \sum_j I(X_i = r, X_j = s), r \in \{0, 1\}; s \in \{0, 1\}$. And $p_{+s} = \sum_r p_{rs}$; $p_{r+} = \sum_s p_{rs}$. See the examples in main function `LPSpectral` for the codes.

References

Roy, S., Atchade, Y., and Michailidis, G. (2017). "Change point estimation in high dimensional Markov random-field models". Journal of the Royal Statistical Society: Series B (Statistical Methodology), 79(4), 1187-1206.

Index

∗ **Main functions**
LP.basis, 2
LP.struct.test, 3
LPSpectral, 5
∗ **datasets**
senate, 6
∗ **package**
LPGraph-package, 2

Laplacian *(*LP.struct.test*)*, 3
LP.basis, 2
LP.struct.test, 3
LPGraph *(*LPGraph-package*)*, 2
LPGraph-package, 2
LPSpectral, 5

senate, 6

wt.mean *(*LP.basis*)*, 2