Package ‘TDAvec’

October 31, 2022

Type Package

Title Vector Summaries of Persistence Diagrams

Version 0.1.3

Description Tools for computing various vector summaries of persistence diagrams studied in Topological Data Analysis. For improved computational efficiency, all code for the vector summaries is written in 'C++' using the 'Rcpp' package.

License GPL (>= 2)

Encoding UTF-8

Imports Rcpp (>= 1.0.9), TDA, microbenchmark

LinkingTo Rcpp

Suggests knitr

VignetteBuilder knitr

NeedsCompilation yes

RoxygenNote 7.2.1

Author Umar Islambekov [aut],
Alexey Luchinsky [aut, cre],
Hasani Pathirana [ctb]

Maintainer Alexey Luchinsky <aluchi@bgsu.edu>

Repository CRAN

Date/Publication 2022-10-31 16:00:23 UTC

R topics documented:

computeECC *A Vector Summary of the Euler Characteristic Curve*

Description

Vectorizes the Euler characteristic curve

$$\chi(t)=\sum_{k=0}^{d}(-1)^k\beta_k(t),$$

where $\beta_0,\beta_1,\dots,\beta_d$ are the Betti curves corresponding to persistence diagrams $D_0,D_1,\dots,D_d$ of dimeansions $0,1,\dots,d$ respectively, all computed from the same filtration

Usage

```
computeECC(D, maxhomDim, scaleSeq)
```

Arguments

`D`	matrix with three columns containing the dimension, birth and death values respectively
`maxhomDim`	maximum homological dimension considered (0 for H_0, 1 for H_1, etc.)
`scaleSeq`	numeric vector of increasing scale values used for vectorization

Value

A numeric vector whose elements are the average values of the Euler characteristic curve computed between each pair of consecutive scale points of `scaleSeq`=$\{t_1,t_2,\dots,t_n\}$:

$$\Big(\frac{1}{\Delta t_1}\int_{t_1}^{t_2}\chi(t)dt,\frac{1}{\Delta t_2}\int_{t_2}^{t_3}\chi(t)dt,\dots,\frac{1}{\Delta t_{n-1}}\int_{t_{n-1}}^{t_n}\chi(t)dt\Big),$$

where $\Delta t_k=t_{k+1}-t_k$

Author(s)

Umar Islambekov

References

1. Richardson, E., & Werman, M. (2014). Efficient classification using the Euler characteristic. Pattern Recognition Letters, 49, 99-106.

Examples

```
N <- 100
set.seed(123)
# sample N points uniformly from unit circle and add Gaussian noise
X <- TDA::circleUnif(N,r=1) + rnorm(2*N,mean = 0,sd = 0.2)

# compute a persistence diagram using the Rips filtration built on top of X
D <- TDA::ripsDiag(X,maxdimension = 1,maxscale = 2)$diagram

scaleSeq = seq(0,2,length.out=11) # sequence of scale values

# compute ECC
computeECC(D,maxhomDim=1,scaleSeq)
```

`computeNL` *A Vector Summary of the Normalized Life Curve*

Description

For a given persistence diagram $D=\{(b_i,d_i)\}_{i=1}^N$, `computeNL()` vectorizes the normalized life (NL) curve

$$sl(t)=\sum_{i=1}^N \frac{d_i-b_i}{L}\mathbf{1}_{[b_i,d_i)}(t),$$

where $L=\sum_{i=1}^N (d_i-b_i)$. Points of D with infinite death value are ignored

Usage

```
computeNL(D, homDim, scaleSeq)
```

Arguments

`D`	matrix with three columns containing the dimension, birth and death values respectively
`homDim`	homological dimension (0 for H_0, 1 for H_1, etc.)
`scaleSeq`	numeric vector of increasing scale values used for vectorization

Value

A numeric vector whose elements are the average values of the persistent entropy summary function computed between each pair of consecutive scale points of `scaleSeq=`$\{t_1,t_2,\ldots,t_n\}$:

$$\Big(\frac{1}{\Delta t_1}\int_{t_1}^{t_2} sl(t)dt, \frac{1}{\Delta t_2}\int_{t_2}^{t_3} sl(t)dt,\ldots,\frac{1}{\Delta t_{n-1}}\int_{t_{n-1}}^{t_n} sl(t)dt\Big),$$

where $\Delta t_k = t_{k+1}-t_k$

Author(s)

Umar Islambekov

References

Chung, Y. M., & Lawson, A. (2022). Persistence curves: A canonical framework for summarizing persistence diagrams. Advances in Computational Mathematics, 48(1), 1-42.

Examples

```
N <- 100
set.seed(123)
# sample N points uniformly from unit circle and add Gaussian noise
X <- TDA::circleUnif(N,r=1) + rnorm(2*N,mean = 0,sd = 0.2)

# compute a persistence diagram using the Rips filtration built on top of X
D <- TDA::ripsDiag(X,maxdimension = 1,maxscale = 2)$diagram

scaleSeq = seq(0,2,length.out=11) # sequence of scale values

# compute NL for homological dimension H_0
computeNL(D,homDim=0,scaleSeq)

# compute NL for homological dimension H_1
computeNL(D,homDim=1,scaleSeq)
```

computePES *A Vector Summary of the Persistent Entropy Summary Function*

Description

For a given persistence diagram $D = \{(b_i, d_i)\}_{i=1}^N$, `computePES()` vectorizes the persistent entropy summary (PES) function

$$S(t) = -\sum_{i=1}^{N} \frac{l_i}{L} \log_2 (\frac{l_i}{L}) \mathbf{1}_{[b_i,d_i]}(t),$$

where $l_i = d_i - b_i$ and $L = \sum_{i=1}^N l_i$. Points of D with infinite death value are ignored

Usage

```
computePES(D, homDim, scaleSeq)
```

Arguments

`D` matrix with three columns containing the dimension, birth and death values respectively

`homDim` homological dimension (0 for H_0, 1 for H_1, etc.)

`scaleSeq` numeric vector of increasing scale values used for vectorization

Value

A numeric vector whose elements are the average values of the persistent entropy summary function computed between each pair of consecutive scale points of `scaleSeq`=$\{t_1, t_2, \ldots, t_n\}$:

$$\Big(\frac{1}{\Delta t_1}\int_{t_1}^{t_2} S(t)dt, \frac{1}{\Delta t_2}\int_{t_2}^{t_3} S(t)dt, \ldots, \frac{1}{\Delta t_{n-1}}\int_{t_{n-1}}^{t_n} S(t)dt\Big),$$

where $\Delta t_k = t_{k+1} - t_k$

Author(s)

Umar Islambekov

References

1. Atienza, N., Gonzalez-Díaz, R., & Soriano-Trigueros, M. (2020). On the stability of persistent entropy and new summary functions for topological data analysis. Pattern Recognition, 107, 107509.

Examples

```
N <- 100
set.seed(123)
# sample N points uniformly from unit circle and add Gaussian noise
X <- TDA::circleUnif(N,r=1) + rnorm(2*N,mean = 0,sd = 0.2)

# compute a persistence diagram using the Rips filtration built on top of X
D <- TDA::ripsDiag(X,maxdimension = 1,maxscale = 2)$diagram

scaleSeq = seq(0,2,length.out=11) # sequence of scale values

# compute PES for homological dimension H_0
computePES(D,homDim=0,scaleSeq)

# compute PES for homological dimension H_1
computePES(D,homDim=1,scaleSeq)
```

`computePI` *A Vector Summary of the Persistence Surface*

Description

For a given persistence diagram $D = \{(b_i, p_i)\}_{i=1}^N$, `computePI()` computes the persistence image (PI) - a vector summary of the persistence surface:

$$\rho(x, y) = \sum_{i=1}^{N} f(b_i, p_i)\phi_{(b_i,p_i)}(x, y),$$

where $\phi_{(b_i,p_i)}(x,y)$ is the Gaussian distribution with mean (b_i,p_i) and covariance matrix $\sigma^2 I_{2\times 2}$ and

$$f(b,p) = w(p) = \begin{cases} 0 & y \leq 0 \\ p/p_{max} & 0 < p < p_{max} \\ 1 & y \geq p_{max} \end{cases}$$

is the weighting function with p_{max} being the maximum persistence value among all persistence diagrams considered in the experiment. Points of D with infinite persistence value are ignored

Usage

```
computePI(D, homDim, xSeq, ySeq, sigma)
```

Arguments

`D`	matrix with three columns containing the dimension, birth and persistence values respectively
`homDim`	homological dimension (0 for H_0, 1 for H_1, etc.)
`xSeq`	numeric vector of increasing x (birth) values used for vectorization
`ySeq`	numeric vector of increasing y (persistence) values used for vectorization
`sigma`	standard deviation of the Gaussian

Value

A numeric vector whose elements are the average values of the persistence surface computed over each cell of the two-dimensional grid constructred from `xSeq`=$\{x_1,x_2,\ldots,x_n\}$ and `ySeq`=$\{y_1,y_2,\ldots,y_m\}$:

$$\Big(\frac{1}{\Delta x_1 \Delta y_1}\int_{[x_1,x_2]\times[y_1,y_2]}\rho(x,y)dA,\ldots,\frac{1}{\Delta x_{n-1}\Delta y_{m-1}}\int_{[x_{n-1},x_n]\times[y_{m-1},y_m]}\rho(x,y)dA\Big),$$

where $dA = dxdy$, $\Delta x_k = x_{k+1} - x_k$ and $\Delta y_j = y_{j+1} - y_j$

Author(s)

Umar Islambekov

References

1. Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., ... & Ziegelmeier, L. (2017). Persistence images: A stable vector representation of persistent homology. Journal of Machine Learning Research, 18.

Examples

```
N <- 100
set.seed(123)
# sample N points uniformly from unit circle and add Gaussian noise
X <- TDA::circleUnif(N,r=1) + rnorm(2*N,mean = 0,sd = 0.2)

# compute a persistence diagram using the Rips filtration built on top of X
```

```
D <- TDA::ripsDiag(X,maxdimension = 1,maxscale = 2)$diagram

# switch from the birth-death to the birth-persistence coordinates
D[,3] <- D[,3] - D[,2]
colnames(D)[3] <- "Persistence"

resB <- 5 # resolution (or grid size) along the birth axis
resP <- 5 # resolution (or grid size) along the persistence axis

# compute PI for homological dimension H_0
minPH0 <- min(D[D[,1]==0,3]); maxPH0 <- max(D[D[,1]==0,3])
ySeqH0 <- seq(minPH0,maxPH0,length.out=resP+1)
sigma <- 0.5*(maxPH0-minPH0)/resP
computePI(D,homDim=0,xSeq=NA,ySeqH0,sigma)

# compute PI for homological dimension H_1
minBH1 <- min(D[D[,1]==1,2]); maxBH1 <- max(D[D[,1]==1,2])
minPH1 <- min(D[D[,1]==1,3]); maxPH1 <- max(D[D[,1]==1,3])
xSeqH1 <- seq(minBH1,maxBH1,length.out=resB+1)
ySeqH1 <- seq(minPH1,maxPH1,length.out=resP+1)
sigma <- 0.5*(maxPH1-minPH1)/resP
computePI(D,homDim=1,xSeqH1,ySeqH1,sigma)
```

`computePL` *A Vector Summary of the Persistence Landscape Function*

Description

Vectorizes the persistence landscape (PL) function constructed from a given persistence diagram. The kth order landscape function of a persistence diagram $D = \{(b_i, d_i)\}_{i=1}^N$ is defined as

$$\lambda_k(t) = k\mathrm{max}_{1\leq i\leq N}\Lambda_i(t), \quad k \in N,$$

where $k$max returns the kth largest value and

$$\Lambda_i(t) = \begin{cases} t - b_i & t \in [b_i, \frac{b_i+d_i}{2}] \\ d_i - t & t \in (\frac{b_i+d_i}{2}, d_i] \\ 0 & \text{otherwise} \end{cases}$$

Usage

```
computePL(D, homDim, scaleSeq, k=1)
```

Arguments

`D` matrix with three columns containing the dimension, birth and death values respectively

`homDim` homological dimension (0 for H_0, 1 for H_1, etc.)

`scaleSeq` numeric vector of increasing scale values used for vectorization

`k` order of landscape function. By default, k=1

Value

A numeric vector whose elements are the values of the kth order landscape function evaluated at each point of `scaleSeq`=$\{t_1, t_2, \ldots, t_n\}$:

$$(\lambda_k(t_1), \lambda_k(t_2), \ldots, \lambda_k(t_n))$$

Author(s)

Umar Islambekov

References

1. Bubenik, P. (2015). Statistical topological data analysis using persistence landscapes. Journal of Machine Learning Research, 16(1), 77-102.

2. Chazal, F., Fasy, B. T., Lecci, F., Rinaldo, A., & Wasserman, L. (2014, June). Stochastic convergence of persistence landscapes and silhouettes. In Proceedings of the thirtieth annual symposium on Computational geometry (pp. 474-483).

Examples

```
N <- 100
set.seed(123)
# sample N points uniformly from unit circle and add Gaussian noise
X <- TDA::circleUnif(N,r=1) + rnorm(2*N,mean = 0,sd = 0.2)

# compute a persistence diagram using the Rips filtration built on top of X
D <- TDA::ripsDiag(X,maxdimension = 1,maxscale = 2)$diagram

scaleSeq = seq(0,2,length.out=11) # sequence of scale values

# compute persistence landscape (PL) for homological dimension H_0 with order of landscape k=1
computePL(D,homDim=0,scaleSeq,k=1)

# compute persistence landscape (PL) for homological dimension H_1 with order of landscape k=1
computePL(D,homDim=1,scaleSeq,k=1)
```

---

computePS *A Vector Summary of the Persistence Silhouette Function*

Description

Vectorizes the persistence silhouette (PS) function constructed from a given persistence diagram. The pth power silhouette function of a persistence diagram $D = \{(b_i, d_i)\}_{i=1}^N$ is defined as

$$\phi_p(t) = \frac{\sum_{i=1}^N |d_i - b_i|^p \Lambda_i(t)}{\sum_{i=1}^N |d_i - b_i|^p},$$

where

$$\Lambda_i(t) = \begin{cases} t - b_i & t \in [b_i, \frac{b_i+d_i}{2}] \\ d_i - t & t \in (\frac{b_i+d_i}{2}, d_i] \\ 0 & \text{otherwise} \end{cases}$$

Points of D with infinite death value are ignored

Usage

```
computePS(D, homDim, scaleSeq, p=1)
```

Arguments

`D`	matrix with three columns containing the dimension, birth and death values respectively
`homDim`	homological dimension (0 for H_0, 1 for H_1, etc.)
`scaleSeq`	numeric vector of increasing scale values used for vectorization
`p`	power of the weights for the silhouette function. By default, `p=1`

Value

A numeric vector whose elements are the average values of the pth power silhouette function computed between each pair of consecutive scale points of `scaleSeq=`$\{t_1, t_2, \ldots, t_n\}$:

$$\Big(\frac{1}{\Delta t_1}\int_{t_1}^{t_2} \phi_p(t)dt, \frac{1}{\Delta t_2}\int_{t_2}^{t_3} \phi_p(t)dt, \ldots, \frac{1}{\Delta t_{n-1}}\int_{t_{n-1}}^{t_n} \phi_p(t)dt\Big),$$

where $\Delta t_k = t_{k+1} - t_k$

Author(s)

Umar Islambekov

References

1. Chazal, F., Fasy, B. T., Lecci, F., Rinaldo, A., & Wasserman, L. (2014). Stochastic convergence of persistence landscapes and silhouettes. In Proceedings of the thirtieth annual symposium on Computational geometry (pp. 474-483).

Examples

```
N <- 100
set.seed(123)
# sample N points uniformly from unit circle and add Gaussian noise
X <- TDA::circleUnif(N,r=1) + rnorm(2*N,mean = 0,sd = 0.2)

# compute a persistence diagram using the Rips filtration built on top of X
D <- TDA::ripsDiag(X,maxdimension = 1,maxscale = 2)$diagram

scaleSeq = seq(0,2,length.out=11) # sequence of scale values
```

```
# compute persistence silhouette (PS) for homological dimension H_0
computePS(D,homDim=0,scaleSeq,p=1)

# compute persistence silhouette (PS) for homological dimension H_1
computePS(D,homDim=1,scaleSeq,p=1)
```

`computeVAB` *A Vector Summary of the Betti Curve*

Description

For a given persistence diagram $D=\{(b_i,d_i)\}_{i=1}^N$, `computeVAB()` vectorizes the Betti Curve

$$\beta(t)=\sum_{i=1}^N w(b_i,d_i)\mathbf{1}_{[b_i,d_i)}(t),$$

where the weight function $w(b,d)\equiv 1$

Usage

```
computeVAB(D, homDim, scaleSeq)
```

Arguments

`D`	matrix with three columns containing the dimension, birth and death values respectively
`homDim`	homological dimension (0 for H_0, 1 for H_1, etc.)
`scaleSeq`	numeric vector of increasing scale values used for vectorization

Value

A numeric vector whose elements are the average values of the Betti curve computed between each pair of consecutive scale points of `scaleSeq=`$\{t_1,t_2,\dots,t_n\}$:

$$\Big(\frac{1}{\Delta t_1}\int_{t_1}^{t_2}\beta(t)dt,\frac{1}{\Delta t_2}\int_{t_2}^{t_3}\beta(t)dt,\dots,\frac{1}{\Delta t_{n-1}}\int_{t_{n-1}}^{t_n}\beta(t)dt\Big),$$

where $\Delta t_k=t_{k+1}-t_k$

Author(s)

Umar Islambekov, Hasani Pathirana

References

1. Chazal, F., & Michel, B. (2021). An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists. Frontiers in Artificial Intelligence, 108.

2. Chung, Y. M., & Lawson, A. (2022). Persistence curves: A canonical framework for summarizing persistence diagrams. Advances in Computational Mathematics, 48(1), 1-42.

Examples

```
N <- 100
set.seed(123)
# sample N points uniformly from unit circle and add Gaussian noise
X <- TDA::circleUnif(N,r=1) + rnorm(2*N,mean = 0,sd = 0.2)

# compute a persistence diagram using the Rips filtration built on top of X
D <- TDA::ripsDiag(X,maxdimension = 1,maxscale = 2)$diagram

scaleSeq = seq(0,2,length.out=11) # sequence of scale values

# compute vector of averaged Bettis (VAB) for homological dimension H_0
computeVAB(D,homDim=0,scaleSeq)

# compute vector of averaged Bettis (VAB) for homological dimension H_1
computeVAB(D,homDim=1,scaleSeq)
```

`computeVPB` *A Vector Summary of the Persistence Block*

Description

For a given persistence diagram $D = \{(b_i, p_i)\}_{i=1}^N$, `computeVPB()` vectorizes the persistence block

$$f(x,y) = \sum_{i=1}^{N} \mathbf{1}_{E(b_i,p_i)}(x,y),$$

where $E(b_i,p_i) = [b_i - \frac{\lambda_i}{2}, b_i + \frac{\lambda_i}{2}] \times [p_i - \frac{\lambda_i}{2}, p_i + \frac{\lambda_i}{2}]$ and $\lambda_i = 2\tau p_i$ with $\tau \in (0,1]$. Points of D with infinite persistence value are ignored

Usage

```
computeVPB(D, homDim, xSeq, ySeq, tau)
```

Arguments

`D`	matrix with three columns containing the dimension, birth and persistence values respectively
`homDim`	homological dimension (0 for H_0, 1 for H_1, etc.)
`xSeq`	numeric vector of increasing x (birth) values used for vectorization
`ySeq`	numeric vector of increasing y (persistence) values used for vectorization
`tau`	parameter (between 0 and 1) controlling block size. By default, `tau=0.3`

Value

A numeric vector whose elements are the weighted averages of the persistence block computed over each cell of the two-dimensional grid constructred from xSeq=$\{x_1, x_2, \ldots, x_n\}$ and ySeq=$\{y_1, y_2, \ldots, y_m\}$:

$$\Big(\frac{1}{\Delta x_1 \Delta y_1}\int_{[x_1,x_2]\times[y_1,y_2]} f(x,y)wdA, \ldots, \frac{1}{\Delta x_{n-1}\Delta y_{m-1}}\int_{[x_{n-1},x_n]\times[y_{m-1},y_m]} f(x,y)wdA\Big),$$

where $wdA = (x+y)dxdy$, $\Delta x_k = x_{k+1} - x_k$ and $\Delta y_j = y_{j+1} - y_j$

Author(s)

Umar Islambekov, Aleksei Luchinsky

References

1. Chan, K. C., Islambekov, U., Luchinsky, A., & Sanders, R. (2022). A computationally efficient framework for vector representation of persistence diagrams. Journal of Machine Learning Research 23, 1-33.

Examples

```
N <- 100
set.seed(123)
# sample N points uniformly from unit circle and add Gaussian noise
X <- TDA::circleUnif(N,r=1) + rnorm(2*N,mean = 0,sd = 0.2)

# compute a persistence diagram using the Rips filtration built on top of X
D <- TDA::ripsDiag(X,maxdimension = 1,maxscale = 2)$diagram

# switch from the birth-death to the birth-persistence coordinates
D[,3] <- D[,3] - D[,2]
colnames(D)[3] <- "Persistence"

# construct one-dimensional grid of scale values
ySeqH0 <- unique(quantile(D[D[,1]==0,3],probs = seq(0,1,by=0.2)))
tau <- 0.3 # parameter in [0,1] which controls the size of blocks around each point of the diagram
# compute VPB for homological dimension H_0
computeVPB(D,homDim = 0,xSeq=NA,ySeqH0,tau)

xSeqH1 <- unique(quantile(D[D[,1]==1,2],probs = seq(0,1,by=0.2)))
ySeqH1 <- unique(quantile(D[D[,1]==1,3],probs = seq(0,1,by=0.2)))
# compute VPB for homological dimension H_1
computeVPB(D,homDim = 1,xSeqH1,ySeqH1,tau)
```

Index