

R Package TDA for Statistical Inference on Topological Data Analysis

Jisu KIM

INRIA Saclay

2019-05-18

Installation

R Package TDA: Statistical Tools for Topological Data Analysis

Homology and Persistent Homology

Sample on manifolds, Distance Functions, and Density Estimators

Persistent Homology and Landscape

Statistical Inference on Persistence Homology and Landscape

For Windows and Mac, TDA can be easily installed.

```
if (!require(package = "TDA")) {  
  install.packages(pkgs = "TDA")  
}
```

For Linux, you need to install several libraries first, and then install TDA.

- ▶ You need to install libraries gmp and mpfr.
- ▶ Then you need to install required R package FNN, igraph, and scales.
- ▶ Then you can install R package TDA.

```
if (!require(package = "FNN")) {  
  install.packages(pkgs = "FNN")  
}  
if (!require(package = "igraph")) {  
  install.packages(pkgs = "igraph")  
}  
if (!require(package = "scales")) {  
  install.packages(pkgs = "scales")  
}  
if (!require(package = "TDA")) {  
  install.packages(pkgs = "TDA")  
}
```

Installation

R Package TDA: Statistical Tools for Topological Data Analysis

Homology and Persistent Homology

Sample on manifolds, Distance Functions, and Density Estimators

Persistent Homology and Landscape

Statistical Inference on Persistence Homology and Landscape

R is ideal for educational purpose.

- ▶ R is a programming language for statistical computing and graphics.
- ▶ Many packages for statistical computing.
- ▶ Easy to make (interactive) plots.
- ▶ Easy to install and use.
- ▶ Platform independent.
- ▶
- ▶ ... but slow.

R Package TDA provides an R interface for C++ libraries for Topological Data Analysis.

- ▶ website:
<https://cran.r-project.org/web/packages/TDA/index.html>
- ▶ Author: Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, Clément Maria, David Milman, and Vincent Rouvreau.
- ▶ R has short development time, while C/C++ has short execution time.
- ▶ R package TDA provides an R interface for C++ library GUDHI/Dionysus/PHAT, which are for Topological Data Analysis.

Installation

R Package TDA: Statistical Tools for Topological Data Analysis

Homology and Persistent Homology

Sample on manifolds, Distance Functions, and Density Estimators

Persistent Homology and Landscape

Statistical Inference on Persistence Homology and Landscape

Number of holes is used to summarize Geometrical features.

- ▶ Geometrical objects :

- ▶ A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z,

- ▶ 가, 字, あ

- ▶ Number of holes of different dimensions is considered.

1. β_0 = # of connected components



2. β_1 = # of loops (holes inside 1-dim sphere)



3. β_2 = # of voids (holes inside 2-dim sphere) : if $dim \geq 3$



Example : Objects are classified by homologies.

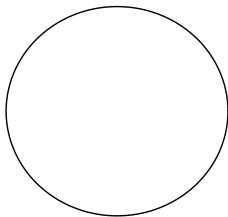
1. $\beta_0 = \#$ of connected components ●

2. $\beta_1 = \#$ of loops ○

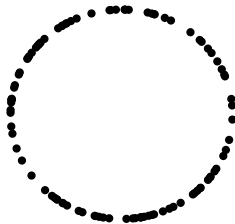
$\beta_0 \setminus \beta_1$	0	1	2
1	C, G, I, J, L, M, N, S, U, V, W, Z, E, F, T, Y, H, K, X	A, R, D, O, P, Q	B, あ
2	가, 字		

When analyzing data, we prefer robust features where features of the underlying manifold can be inferred from features of finite samples.

Underlying circle



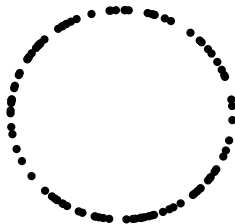
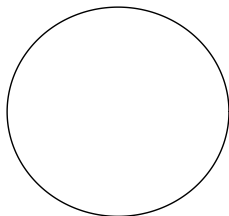
100 samples



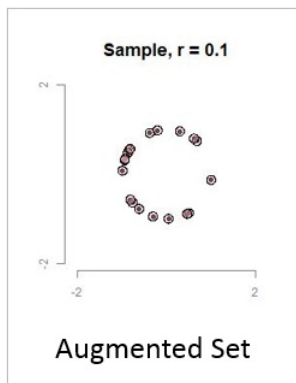
Homology of finite sample is different from homology of underlying manifold, hence it cannot be directly used for the inference.

Underlying circle: $\beta_0 = 1, \beta_1 = 1$

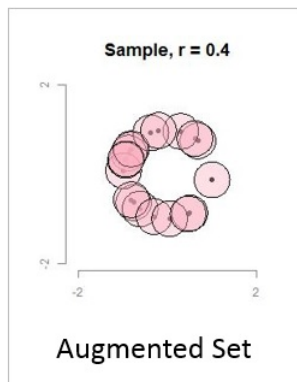
100 samples: $\beta_0 = 100, \beta_1 = 0$



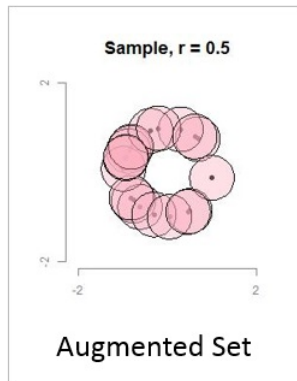
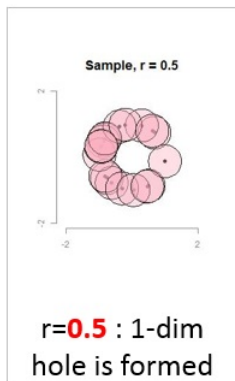
Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.



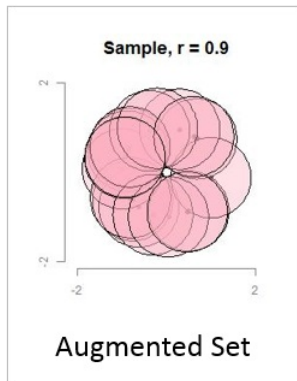
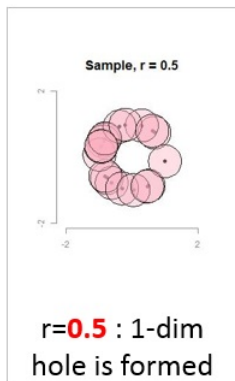
Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.



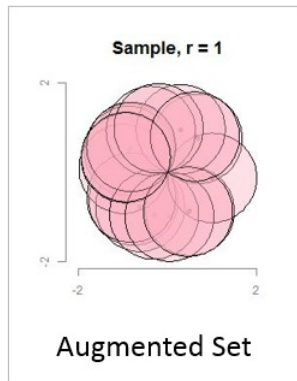
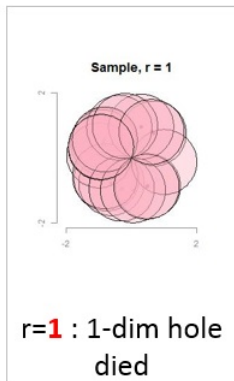
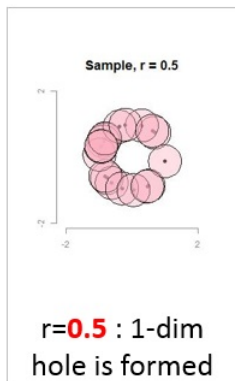
Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.



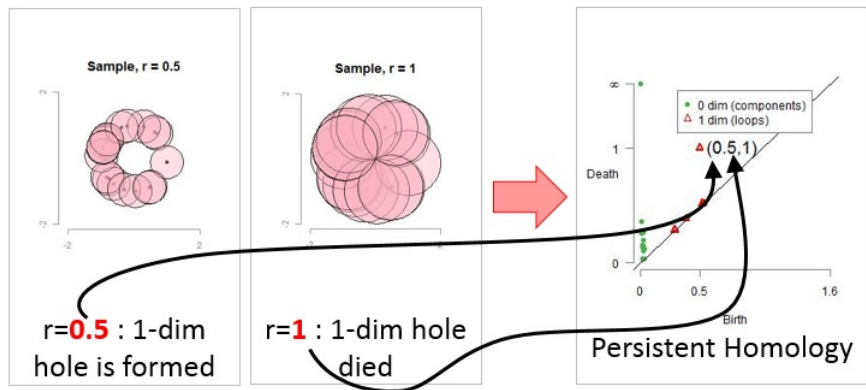
Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.



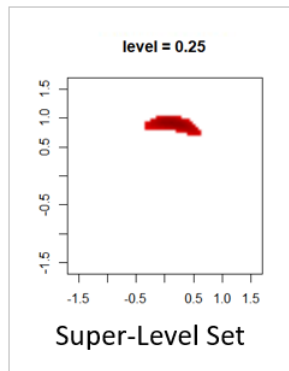
Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.



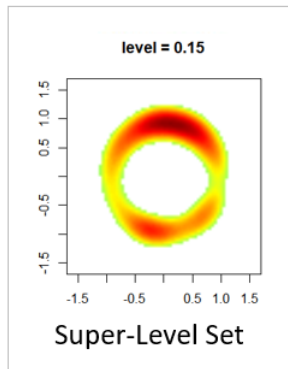
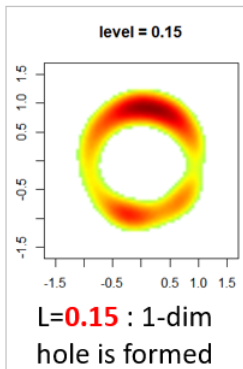
Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.



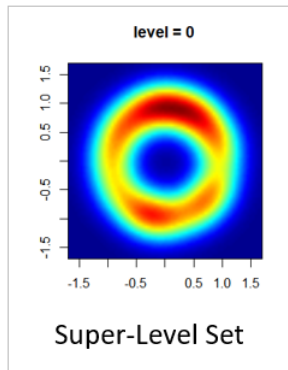
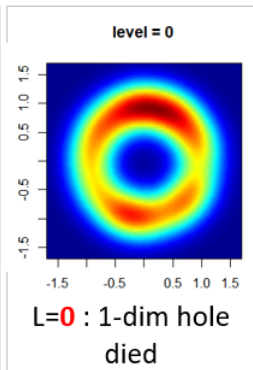
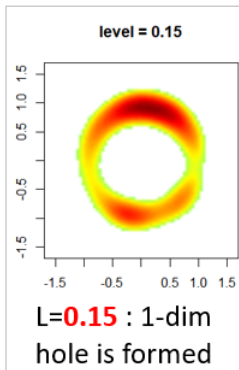
Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.



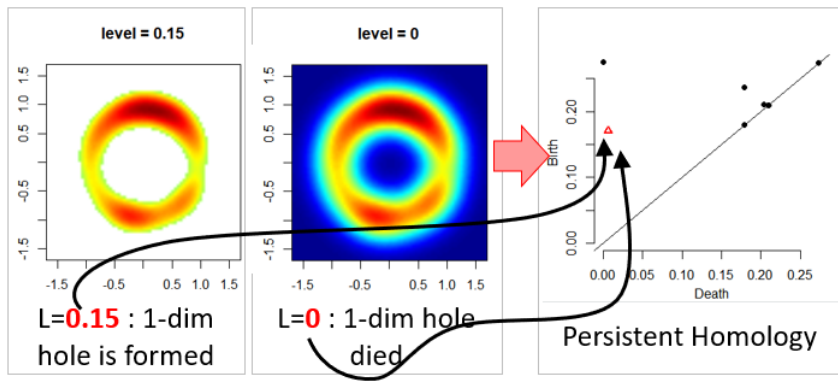
Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.



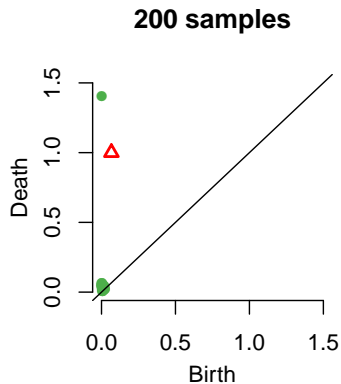
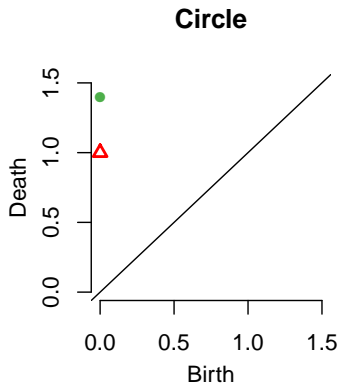
Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.



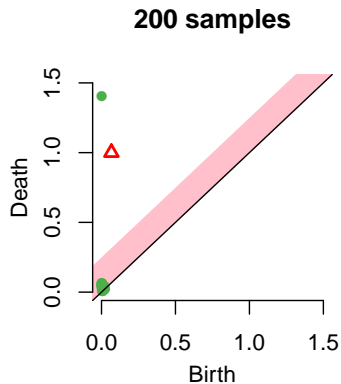
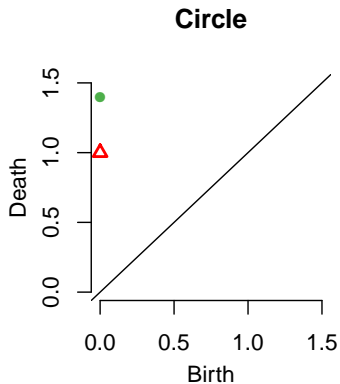
Persistent homology computes homologies on collection of sets, and tracks when topological features are born and when they die.



Persistent homology of the underlying manifold can be inferred from persistent homology of finite samples.

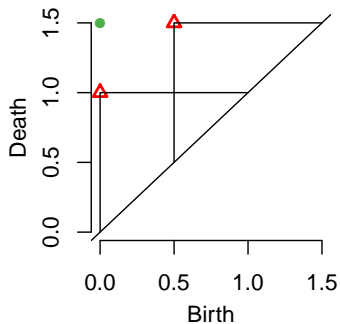


Confidence band for persistent homology separates homological signal from homological noise.

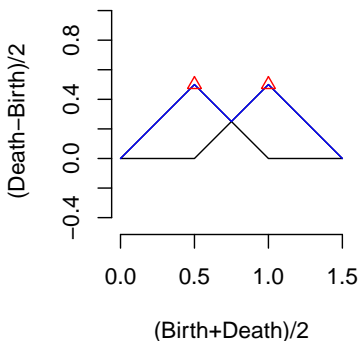


Landscape is a functional summary of the persistent homology.

Persistent Homology

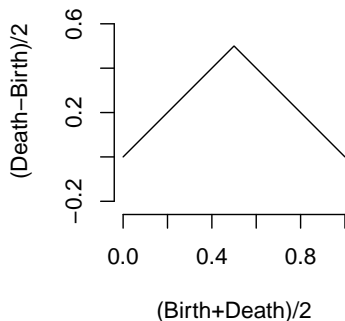


Landscape

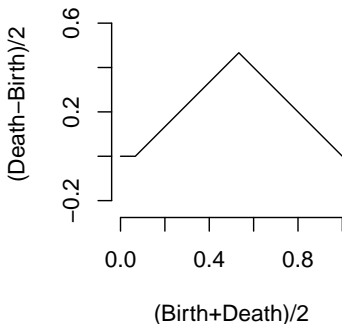


Landscape of the underlying manifold can be inferred from landscape of finite samples.

Circle



200 samples



Installation

R Package TDA: Statistical Tools for Topological Data Analysis

Homology and Persistent Homology

Sample on manifolds, Distance Functions, and Density Estimators

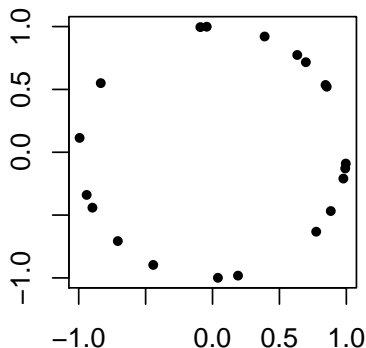
Persistent Homology and Landscape

Statistical Inference on Persistence Homology and Landscape

R Package TDA provides a function to sample on a circle.

The function `circleUnif()` generates n sample from the uniform distribution on the circle in \mathbb{R}^2 with radius r .

```
circleSample <- circleUnif(n = 20, r = 1)
plot(circleSample, xlab = "", ylab = "", pch = 20)
```



R Package TDA provides distance functions and density functions over a grid.

Suppose $n = 400$ points are generated from the unit circle, and grid of points are generated.

```
X <- circleUnif(n = 400, r = 1)

lim <- c(-1.7, 1.7)
by <- 0.05
margin <- seq(from = lim[1], to = lim[2], by = by)
Grid <- expand.grid(margin, margin)
```

R Package TDA provides DTM function over a grid.

The distance to measure (DTM) $d_{m0} : \mathbb{R}^d \rightarrow [0, \infty)$ is defined as

$$d_{m0}(y) = \left(\frac{1}{k} \sum_{x_i \in N_k(y)} \|x_i - y\|^r \right)^{1/r},$$

where $k = \lceil m0 \times n \rceil$ and $m0 \in (0, 1)$, $r \in [1, \infty)$ are tuning parameters. The function `dtm()` computes the DTM function d_{m0} on a grid of points.

```
m0 <- 0.1
DTM <- dtm(X = X, Grid = Grid, m0 = m0)

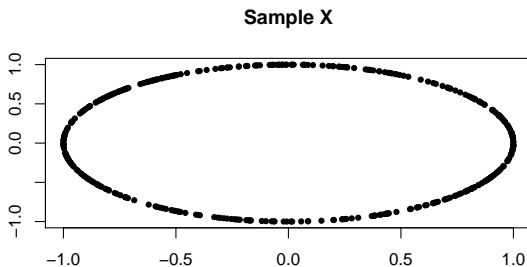
plot(X, xlab = "", ylab = "", main = "Sample X", pch = 20)
persp(x = margin, y = margin,
      z = matrix(DTM, nrow = length(margin), ncol = length(margin)),
      xlab = "", ylab = "", zlab = "", theta = -20, phi = 35, scale = FALSE,
      expand = 3, col = "red", border = NA, ltheta = 50, shade = 0.5,
      main = "DTM")
```

R Package TDA provides DTM function over a grid.

The distance to measure (DTM) $d_{m0} : \mathbb{R}^d \rightarrow [0, \infty)$ is defined as

$$d_{m0}(y) = \left(\frac{1}{k} \sum_{x_i \in N_k(y)} \|x_i - y\|^r \right)^{1/r},$$

where $k = \lceil m0 \times n \rceil$ and $m0 \in (0, 1)$, $r \in [1, \infty)$ are tuning parameters. The function `dtm()` computes the DTM function d_{m0} on a grid of points.



DTM

R Package TDA provides KDE function over a grid.

The Gaussian Kernel Density Estimator (KDE) $\hat{p}_h : \mathbb{R}^d \rightarrow [0, \infty)$ is defined as

$$\hat{p}_h(y) = \frac{1}{n(\sqrt{2\pi}h)^d} \sum_{i=1}^n \exp\left(\frac{-\|y - x_i\|_2^2}{2h^2}\right),$$

where h is a smoothing parameter.

The function `kde()` computes the KDE function \hat{p}_h on a grid of points.

```
h <- 0.3
KDE <- kde(X = X, Grid = Grid, h = h)

par(mfrow = c(1,2))
plot(X, xlab = "", ylab = "", main = "Sample X", pch = 20)
persp(x = margin, y = margin,
      z = matrix(KDE, nrow = length(margin), ncol = length(margin)),
      xlab = "", ylab = "", zlab = "", theta = -20, phi = 35, scale = FALSE,
      expand = 3, col = "red", border = NA, ltheta = 50, shade = 0.5,
      main = "KDE")
```

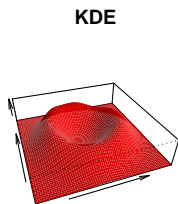
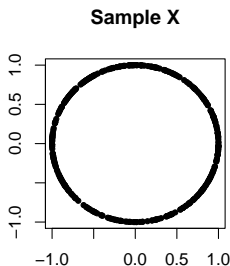

R Package TDA provides KDE function over a grid.

The Gaussian Kernel Density Estimator (KDE) $\hat{p}_h : \mathbb{R}^d \rightarrow [0, \infty)$ is defined as

$$\hat{p}_h(y) = \frac{1}{n(\sqrt{2\pi}h)^d} \sum_{i=1}^n \exp\left(\frac{-\|y - x_i\|_2^2}{2h^2}\right),$$

where h is a smoothing parameter.

The function `kde()` computes the KDE function \hat{p}_h on a grid of points.



Installation

R Package TDA: Statistical Tools for Topological Data Analysis

Homology and Persistent Homology

Sample on manifolds, Distance Functions, and Density Estimators

Persistent Homology and Landscape

Statistical Inference on Persistence Homology and Landscape

R Package TDA computes Persistent Homology over a grid.

- ▶ The function `gridDiag()` computes the persistence diagram of sublevel (and superlevel) sets of the input function.
 - ▶ `gridDiag()` evaluates the real valued input function over a grid.
 - ▶ `gridDiag()` constructs a filtration of simplices using the values of the input function.
 - ▶ `gridDiag()` computes the persistent homology of the filtration.
- ▶ The user can choose to compute persistent homology using either C++ library GUDHI, Dionysus, or PHAT.

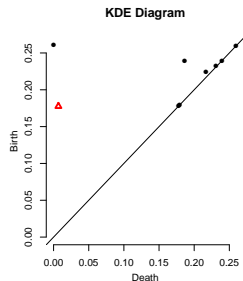
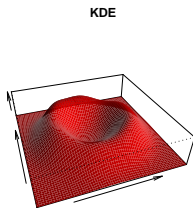
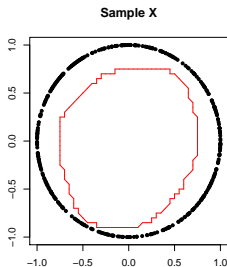
R Package TDA computes Persistent Homology over a grid.

```
DiagGrid <- gridDiag(X = X, FUN = kde, lim = c(lim, lim), by = by,
  sublevel = FALSE, library = "Dionysus", location = TRUE,
  printProgress = FALSE, h = h)

par(mfrow = c(1,3))
plot(X, xlab = "", ylab = "", main = "Sample X", pch = 20)
one <- which(DiagGrid[["diagram"]][, 1] == 1)
for (i in seq(along = one)) {
  for (j in seq_len(dim(DiagGrid[["cycleLocation"]][[one[i]]])[1])) {
    lines(DiagGrid[["cycleLocation"]][[one[i]]][j, , ], pch = 19, cex = 1,
      col = i + 1)
  }
}
persp(x = margin, y = margin,
  z = matrix(KDE, nrow = length(margin), ncol = length(margin)),
  xlab = "", ylab = "", zlab = "", theta = -20, phi = 35, scale = FALSE,
  expand = 3, col = "red", border = NA, ltheta = 50, shade = 0.9,
  main = "KDE")
plot(x = DiagGrid[["diagram"]], main = "KDE Diagram")
```

R Package TDA computes Persistent Homology over a grid.

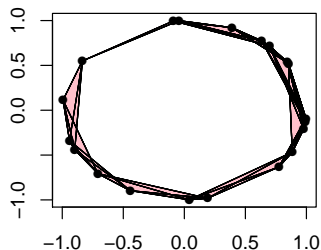
- ▶ The function `gridDiag()` computes the persistent homology of sublevel (and superlevel) sets of the input function.
 - ▶ `gridDiag()` evaluates the real valued input function over a grid.
 - ▶ `gridDiag()` constructs a filtration of simplices using the values of the input function.
 - ▶ `gridDiag()` computes the persistent homology of the filtration.
- ▶ The user can choose to compute persistent homology using either GUDHI, Dionysus, or PHAT.



R Package TDA computes Rips Persistent Homology.

- ▶ Rips complex consists of simplices whose pairwise distances of vertices are at most ϵ apart, i.e.

$$R(X, \epsilon) = \{[X_{n_1}, \dots, X_{n_r}] : d(X_{n_i}, X_{n_j}) \leq \epsilon\}.$$



- ▶ Rips filtration is formed by Rips complexes with gradually increasing ϵ .

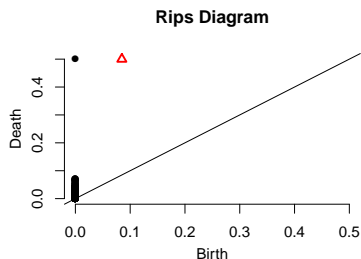
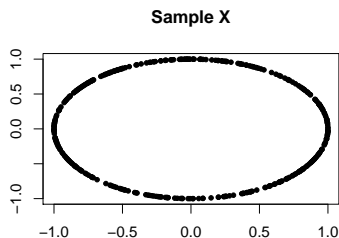
R Package TDA computes Rips Persistent Homology.

- ▶ The function `ripsDiag()` computes the persistence diagram of the Rips filtration built on top of a point cloud.
 - ▶ `ripsDiag()` constructs the Rips filtration using the data points.
 - ▶ `ripsDiag()` computes the persistent homology of the Rips filtration.
- ▶ The user can choose to compute persistent homology using either C++ library GUDHI, Dionysus, or PHAT.

```
DiagRips <- ripsDiag(X = X, maxdimension = 1, maxscale = 0.5,  
  library = c("GUDHI", "Dionysus"), location = TRUE)  
  
par(mfrow = c(1,2))  
plot(X, xlab = "", ylab = "", main = "Sample X", pch = 20)  
plot(x = DiagRips[["diagram"]], main = "Rips Diagram")
```

R Package TDA computes Rips Persistent Homology.

- ▶ The function `ripsDiag()` computes the persistence diagram of the Rips filtration built on top of a point cloud.
 - ▶ `ripsDiag()` constructs the Rips filtration using the data points.
 - ▶ `ripsDiag()` computes the persistent homology of the Rips filtration.
- ▶ The user can choose to compute persistent homology using either C++ library GUDHI, Dionysus, or PHAT.



R Package TDA computes Landscape.

- ▶ Let Λ_p be created by tenting each point $p = (x, y) = (\frac{b+d}{2}, \frac{d-b}{2})$ representing a birth-death pair (b, d) in the persistence diagram D .
- ▶ The persistence landscape of D is the collection of functions

$$\lambda_k(t) = k \max_p \Lambda_p(t), \quad t \in [0, T], k \in \mathbb{N},$$

where k max is the k th largest value in the set.

- ▶ The function `landscape()` evaluates the landscape function $\lambda_k(t)$.

```
tseq <- seq(0, 0.2, length = 1000)
Land <- landscape(DiagGrid[["diagram"]], dimension = 1, KK = 1, tseq = tseq)

par(mfrow = c(1,2))
plot(x = DiagGrid[["diagram"]], main = "KDE Diagram")
plot(tseq, Land, type = "l", xlab = "(Birth+Death)/2",
      ylab = "(Death-Birth)/2", asp = 1, axes = FALSE, main = "Landscape")
axis(1); axis(2)
```

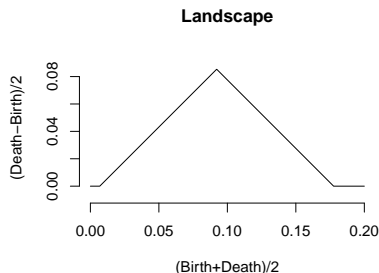
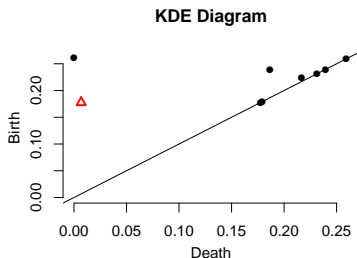
R Package TDA computes Landscape.

- ▶ Let Λ_p be created by tenting each point $p = (x, y) = (\frac{b+d}{2}, \frac{d-b}{2})$ representing a birth-death pair (b, d) in the persistence diagram D .
- ▶ The persistence landscape of D is the collection of functions

$$\lambda_k(t) = k \max_p \Lambda_p(t), \quad t \in [0, T], k \in \mathbb{N},$$

where $k \max$ is the k th largest value in the set.

- ▶ The function `landscape()` evaluates the landscape function $\lambda_k(t)$.



Installation

R Package TDA: Statistical Tools for Topological Data Analysis

Homology and Persistent Homology

Sample on manifolds, Distance Functions, and Density Estimators

Persistent Homology and Landscape

Statistical Inference on Persistence Homology and Landscape

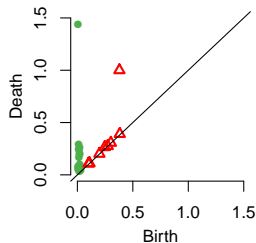
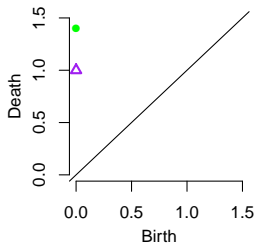
Bottleneck distance gives a metric on the space of persistent homology.

Definition

Let D_1, D_2 be multiset of points. Bottleneck distance is defined as

$$W_\infty(D_1, D_2) = \inf_{\gamma} \sup_{x \in D_1} \|x - \gamma(x)\|_\infty,$$

where γ ranges over all bijections from D_1 to D_2 .



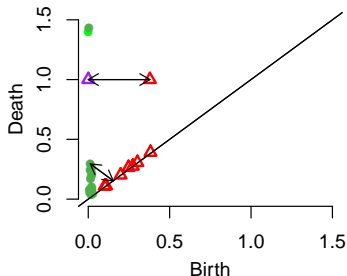
Bottleneck distance gives a metric on the space of persistent homology.

Definition

Let D_1, D_2 be multiset of points. Bottleneck distance is defined as

$$W_\infty(D_1, D_2) = \inf_{\gamma} \sup_{x \in D_1} \|x - \gamma(x)\|_\infty,$$

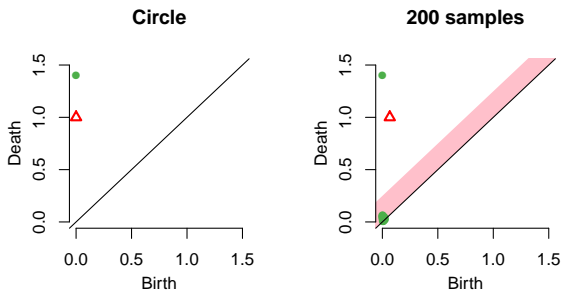
where γ ranges over all bijections from D_1 to D_2 .



Confidence band for the persistent homology is a random quantity containing the persistent homology with high probability.

Let M be a compact manifold, and $X = \{X_1, \dots, X_n\}$ be n samples. Let f_M and f_X be corresponding functions whose persistent homology is of interest. Given the significance level $\alpha \in (0, 1)$, $(1 - \alpha)$ confidence band $c_n = c_n(X)$ is a random variable satisfying

$$\mathbb{P}(W_\infty(Dgm(f_M), Dgm(f_X)) \leq c_n) \geq 1 - \alpha.$$



Confidence band for the persistent homology can be computed using the bootstrap algorithm.

1. Given a sample $X = \{x_1, \dots, x_n\}$, compute the kernel density estimator $\hat{\rho}_h$.
2. Draw $X^* = \{x_1^*, \dots, x_n^*\}$ from $X = \{x_1, \dots, x_n\}$ (with replacement), and compute $\theta^* = \sqrt{n} \|\hat{\rho}_h^*(x) - \hat{\rho}_h(x)\|_\infty$, where $\hat{\rho}_h^*$ is the density estimator computed using X^* .
3. Repeat the previous step B times to obtain $\theta_1^*, \dots, \theta_B^*$
4. Compute $q_\alpha = \inf \left\{ q : \frac{1}{B} \sum_{j=1}^B I(\theta_j^* \geq q) \leq \alpha \right\}$
5. The $(1 - \alpha)$ confidence band for $\mathbb{E}[\hat{\rho}_h]$ is $\left[\hat{\rho}_h - \frac{q_\alpha}{\sqrt{n}}, \hat{\rho}_h + \frac{q_\alpha}{\sqrt{n}} \right]$.

R Package TDA computes the bootstrap confidence band for a function.

The function `bootstrapBand()` computes $(1 - \alpha)$ bootstrap confidence band for $\mathbb{E}[\hat{\rho}_h]$.

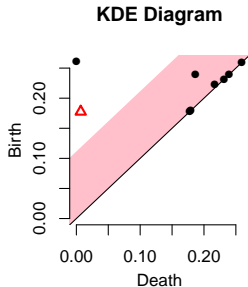
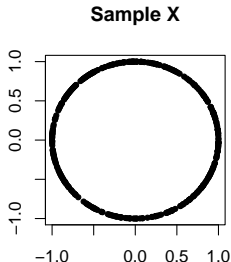
```
bandKDE <- bootstrapBand(X = X, FUN = kde, Grid = Grid, B = 20,  
  parallel = FALSE, alpha = 0.1, h = h)  
print(bandKDE[["width"]])
```

```
##          90%  
## 0.05576625
```


The bootstrap confidence band for a function is used as the confidence band for the persistent homology.

The $(1 - \alpha)$ bootstrap confidence band for $\mathbb{E}[\hat{\rho}_h]$ is used as the confidence band for the persistent homology.

```
par(mfrow = c(1,2))
plot(X, xlab = "", ylab = "", main = "Sample X", pch = 20)
plot(x = DiagGrid[["diagram"]], band = 2 * bandKDE[["width"]],
     main = "KDE Diagram")
```

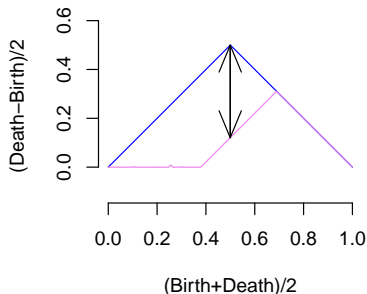


∞ -landscape distance gives a metric on the space of landscapes.

Definition

Let D_1, D_2 be multiset of points, and λ_1, λ_2 be corresponding landscapes. ∞ -landscape distance is defined as

$$\Lambda_\infty(D_1, D_2) = \|\lambda_1 - \lambda_2\|_\infty.$$



∞ -landscape distance can be controlled by the corresponding distance on functions: Stability Theorem.

Theorem

Let $f, g : \mathbb{X} \rightarrow \mathbb{R}$ be two functions, and let $Dgm(f)$ and $Dgm(g)$ be corresponding persistent homologies. Then

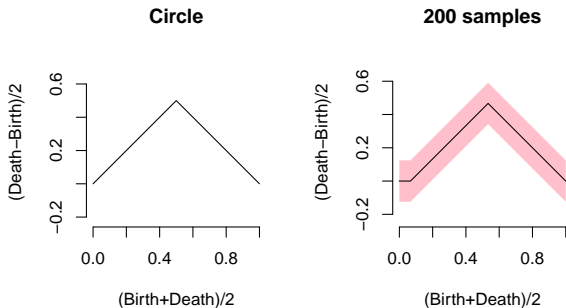
$$\Lambda_{\infty}(Dgm(f), Dgm(g)) \leq \|f - g\|_{\infty}.$$

Confidence band for the landscape can be computed using the bootstrap algorithm.

- ▶ Let λ_M and λ_X be landscapes of the manifold M and samples X . From Stability Theorem, $\mathbb{P}(\|f_M - f_X\| \leq c_n) \geq 1 - \alpha$ implies

$$\mathbb{P}(\lambda_X(t) - c_n \leq \lambda_M(t) \leq \lambda_X(t) + c_n \forall t) \geq \mathbb{P}(\|f_M - f_X\| \leq c_n) \geq 1 - \alpha,$$

so the confidence band of corresponding functions f_M can be used for confidence band of the landscape λ_M .



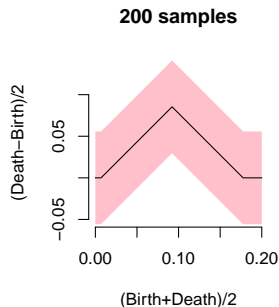
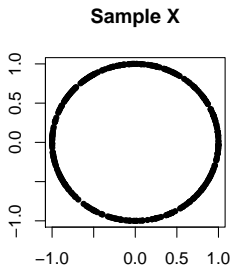
The bootstrap confidence band for a function is used as the confidence band for the landscape.

The $(1 - \alpha)$ bootstrap confidence band for $\mathbb{E}[\hat{\rho}_h]$ is used as the confidence band for the landscape.

```
par(mfrow = c(1,2))
plot(X, xlab = "", ylab = "", main = "Sample X", pch = 20)
plot(tseq, Land, type = "l", xlab = "(Birth+Death)/2",
      ylab = "(Death-Birth)/2", asp = 1, axes = FALSE, main = "200 samples")
axis(1); axis(2)
polygon(c(tseq, rev(tseq)), c(Land - bandKDE[["width"]],
                              rev(Land + bandKDE[["width"]])), col = "pink", lwd = 1.5,
        border = NA)
lines(tseq, Land)
```

The bootstrap confidence band for a function is used as the confidence band for the landscape.

The $(1 - \alpha)$ bootstrap confidence band for $\mathbb{E}[\hat{\rho}_h]$ is used as the confidence band for the landscape.



Reference

Thank you!