1 Exercises on Topological Data Analysis

1.1 Delivery 5

Exercise 1. *Consider the functions $f, g : [-1, 1] \to \mathbb{R}$, defined by $f(x) = x^5 - x$, and $g(x) = \frac{1}{5}(x^9 + 7x^5 - 10x)$.*

(a) Find $V(f), V(g)$, and their spectrums A_f, A_g.

For each $t \in \mathbb{R}$ we define the sublevel set as

$$L_t(f) = \{x \in [-1, 1] \mid f(x) \leq t\}$$

then we want to find $V(f)$ defined by $V_t(f) = H_0(L_t(f))$, and $\pi_{s,t} : V_s(f) \to V_t(f)$ induced by the inclusion.
Let us first find the critical points:

- $f'(x) = 5x^4 - 1 = 0$ has the real roots $x_1 = -\sqrt[4]{\frac{1}{5}} \approx -0.669, x_2 = \sqrt[4]{\frac{1}{5}} \approx 0.669$
- $g'(x) = \frac{1}{5}(9x^8 + 35x^4 - 10) = 0$ has the real roots $x_3 \approx -0.719$, $x_4 \approx 0.719$ in the interval $[-1, 1]$.

An illustrative plot of the two functions is shown in Figure 1.

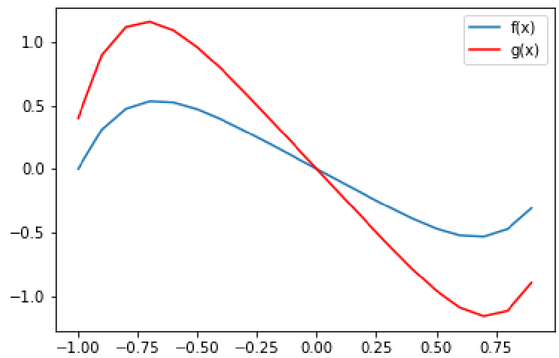

Figure 1: Illustrative plot of functions f, g.

Consider now the corresponding sublevel sets for each function and then the computation of the 0-homologies are direct since we obtain real intervals, and by homotopy invariance of singular homology we thus have $H_0(I) \cong \mathbb{F}$, and $H_0(I \cup J) \cong \mathbb{F} \oplus \mathbb{F}$.

- We have $f(x_1) \approx 0.535 = sup(f)$, $f(x_2) = -0.535 = inf(f)$, and $f(-1) = f(1) = 0$. Then the sublevel sets are defined as follows:

  $t < inf(f) = f(x_2) \Rightarrow L_t(f) = \emptyset \Rightarrow H_0(\emptyset) = 0$
  $f(x_2) \leq t < 0 = f(-1) \Rightarrow L_t(f) = [a, b] \subset [0, 1] \Rightarrow H_0([a, b]) = \mathbb{F}$
  $0 \leq t < f(x_1) = sup(f) \Rightarrow L_t(f) = [a, b] \cup [c, d] \subset [-1, 1]$, where $[a, b] \subset [-1, 0], [c, d] \subset [0, 1] \Rightarrow H_0([a, b] \cup [c$
  $t \geq f(x_1) \Rightarrow L_t(f) = [-1, 1] \Rightarrow H_0([-1, 1]) = \mathbb{F}$

- Following the same procedure for function g, knowing that $g(x_3) \approx 1.159 = sup(g)$, $g(x_4) \approx$

$-1.159 = inf(g)$ we have:

$$t < inf(g) = g(x_4) \Rightarrow L_t(g) = \emptyset \Rightarrow H_0(\emptyset) = 0$$
$$g(x_4) \leq t < g(-1) = 0.4 \Rightarrow L_t(g) = [a,b] \subset [y,1], \text{ where } g(y) = g(0) = 0.4 \Rightarrow H_0([a,b]) = \mathbb{F}$$
$$g(-1) \leq t < g(x_3) = sup(g) \Rightarrow L_t(g) = [a,b] \cup [c,d] \subset [-1,1], \text{ where } [a,b] \subset [-1,y], [c,d] \subset [y,1] \Rightarrow H_0([a,b$$
$$t \geq g(x_3) \Rightarrow L_t(g) = [-1,1] \Rightarrow H_0([-1,1]) = \mathbb{F}$$

Additionally, in this case since the functions are real, we can take $\mathbb{F} = \mathbb{Z}$ as the field of coefficients.

Note therefore that the corresponding spectrums are given by $A_f = \{-1, x_1, x_2\}, A_g = \{-1, x_3, x_4\}$.

(b) Compute the interleaving distance.

Note that for $t = 0$, $t \in [f(x_2), 0] \cap [g(x_4, g(-1)]$ we have $V_t(f) = V_t(g) = \mathbb{Z}$ and therefore the interleaving distance is 0. In any other case, this is if we are considering t in the intervals $I_1 = [f(x_1), g(x_2)]$, $I_2 = [g(x_4), f(x_3)]$, or $I_3 = [f(-1), g(-1)]$, then we would have to shift the persistence module a distance equal to the absolute difference between the value of the functions, since in those intervals we have different definitions of the corresponding $V_t(f), V_t(g)$. For example, for the interval $t \in [f(-1), g(-1)]$, we have $V_t(f) \cong \mathbb{F} \oplus \mathbb{F}$, while $V_t(g) \cong \mathbb{F}$, then, by shifting $V_t(f)$ a distance $d_{int} = \delta = |f(-1) - g(-1)| = 0.4$ we have again $V_t(f)[\delta] \cong \mathbb{F} \cong V_t(g)$ as wanted. Same argument applies to the other two intervals in which we have, for instance for $t \in I_1$ then $V_t(f)[\delta] \cong \mathbb{F} \oplus \mathbb{F} \cong V_t(g)$ if δ brings t to the interval $[g(-1), f(x_1)]$, $\delta = t - f(x_1)$, which is maximum when $t = g(x_3)$, in which case $\delta \approx 0.624$. Similar argument applies to the remaining interval, obtaining the same results since the functions are symmetric in the interval $[-1, 1]$.

(c) Check $d_{int}(V(f), V(g)) < ||f - g||_\infty$ on $[-1, 1]$

We know $||f - g||_\infty = sup\{|f(x) - g(x), -1, \leq x \leq 1\}$, thus consider the function $h(x) = f(x) - g(x) = \frac{1}{5}(x^9 + 2x^5 - 5x)$, which by computing the corresponding critical points we know that we have a local maximum and a local minimum at points $(-0.782, 0.643), (0.782, -0.643)$ respectively, in the interval $[-1, 1]$. Thus the corresponding maximum of the absolute value of $h(x)$ is given by $||f - g||_\infty \approx 0.643$. Note therefore that clearly the values obtained on the previous exercise are lower than this boundary found, verifying the stability theorem.

Exercise 2: *Consider the point clouds:*

$$X = \{(0.81, 2.87), (2.15, 1.18), (3.19, 3.62), (4.17, 2.01), (5.32, 4.88), (6.21, 3.13)\}$$
$$Y = \{(0.75, 2.80), (2.33, 1.25), (3.28, 3.66), (4.15, 2.15), (5.24, 4.78), (6.34, 3.12)\}$$

(a) Compute the Hausdorff distance and the Gromov–Hausdorff distance.

By definition the Hausdorff distance is given by

$$d_H(X, Y) = max\{sup_{x \in X} inf_{y \in Y} d(x, y), sup_{y \in Y} inf_{x \in X} d(x, y)\}$$

. By implementing the following code in python the corresponding Hausdorff distance is computed:

```
from scipy.spatial.distance import directed_hausdorff
X1 = np.array([(0.81, 2.87), (2.15, 1.18), (3.19, 3.62),
(4.17, 2.01), (5.32, 4.88), (6.21, 3.13)])
Y1 = np.array([(0.75,2.80),(2.33,1.25),(3.28,3.66),
(4.15,2.15),(5.24,4.78),(6.34,3.12)])
```

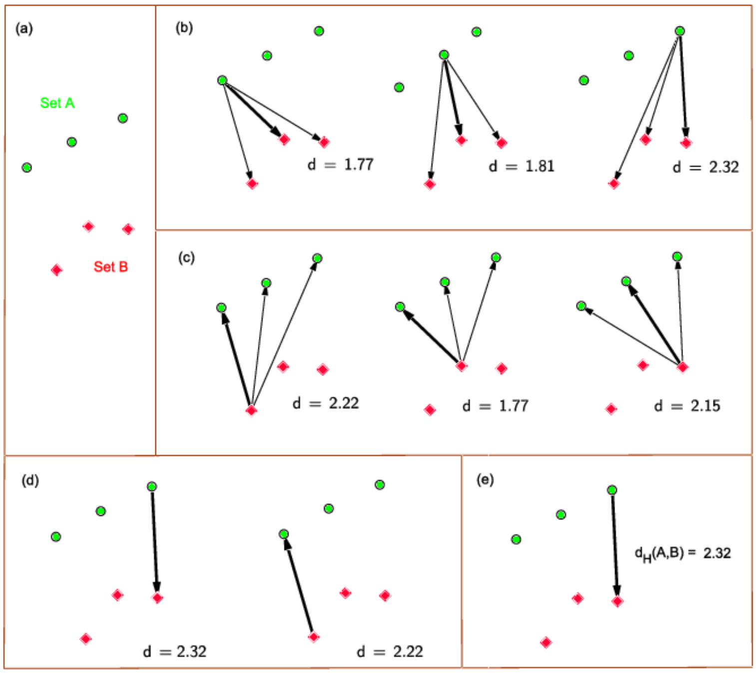

Figure 2: Hausdorff distance for two point sets A and B. (a) The point sets. (b) Computation of the Euclidean distance from each element of the set A to each point of B (values in white refer to the thicker arrow corresponding to the minimum distance). (c) Computation of the Euclidean distance from each element of the set B to each point of A (values in white refer to the thicker arrow corresponding to the minimum distance). (d) Maximum distances between sets. (e) Hausdorff distance between set A and B: dH (A, B) = 2.32.

```
max(directed_hausdorff(X1, Y1)[0], directed_hausdorff(Y1, X1)[0])
```

and the value obtained accounts for the maximum distance between the supreme of the minimum distance per point to each of the other points. An illustration of the algorithm is shown on Figure 2[1]. The value obtained corresponds to $d_H \approx 0.1931$, and is given by the distance between the second point from each point cloud, which can be distinguished in Figure 3. Note in fact that this could have been implemented using R studio by using the following code:

```
X1 <- matrix(c(0.81, 2.15, 3.19, 4.17, 5.32, 6.21,
       2.87, 1.18, 3.62, 2.01, 4.88, 3.13), ncol = 2, byrow = FALSE)
Y1 <- matrix(c(0.75, 2.33, 3.28, 4.15, 5.24, 6.34,
       2.80, 1.25, 3.66, 2.15, 4.78, 3.12), ncol = 2, byrow= FALSE)
max(distFct(X1, Y1))
```

which gives the exact same result.

Similarly, the Gromov–Hausdorff distance measures how far two compact metric spaces are from being isometric. If X and Y are two compact metric spaces, then $d_{GH}(X, Y)$ is defined to be the infimum of all numbers $d_H(f(X), g(Y))$, for all metric spaces M and all isometric embeddings $f : X \to M$ and $g : Y \to M$.

[1] Diagram from `https://www.researchgate.net/figure/Hausdorff-distance-for-two-point-sets-A-and-B-a-The-point-sets` `fig1_273649880`

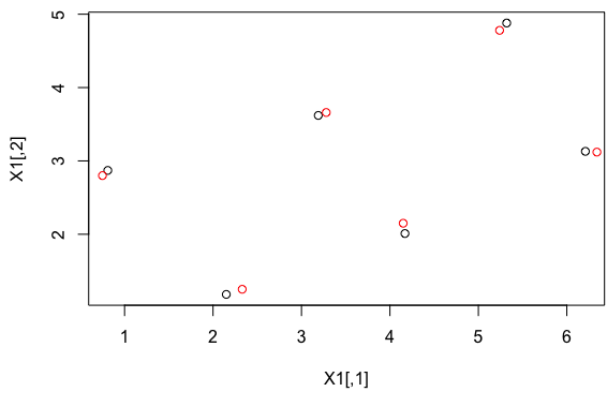

Figure 3: Plot of point clouds X, in black, and Y in red.

(b) Compute the bottleneck distance $W_\infty(D(X), D(Y))$ between the Vietoris– Rips persistence diagrams of X and Y

Bottleneck distance between two persistence diagrams is defined as

$$W_\infty(D, D') = min\{||\phi|| \ s.t. \ \phi : D \to D' \text{ matching } \}$$

this is the smallest $\epsilon \geq 0$ for which there exists a matching $\phi : D \to D'$ for which $d_\infty((x, y), \phi(x, y)) \leq \epsilon$, for all $(x, y) \in D$.
By implementing the following code in R the corresponding Bottleneck distance is computed:

```
X1 <- matrix(c(0.81, 2.15, 3.19, 4.17, 5.32, 6.21,
       2.87, 1.18, 3.62, 2.01, 4.88, 3.13), ncol = 2, byrow = FALSE)
Y1 <- matrix(c(0.75, 2.33, 3.28, 4.15, 5.24, 6.34,
       2.80, 1.25, 3.66, 2.15, 4.78, 3.12), ncol = 2, byrow= FALSE)

Diag <- ripsDiag(X1, maxdimension = 2, maxscale = 5, library = 'GUDHI')
Diag_y <- ripsDiag(Y1, maxdimension = 2, maxscale = 5, library = 'GUDHI')
bot = bottleneck(Diag[['diagram']], Diag_y[['diagram']])
```

obtaining a distance of $W_\infty = 0.2125653$, since `ripsDiag` already computes the Vietoris-Rips complex of the corresponding point cloud.