

## More Harmonics: Sliding Windows, and 1-D Persistence

In this lab, we will study the analysis of the same set of harmonics as in the previous lab. This time, we will use your newly acquired knowledge of 1-dimensional persistence.

To start, a thought exercise:

## Question

1. Draw  $Dgm_1(S^1)$ , where  $S^1$  is the unit sphere in  $\mathbb{R}^2$ .
2. Suppose we sample some points from  $S^1$ , but with a small amount of bounded radial noise. Draw the new 1-dimensional persistence diagram.

**Question** Now that we know what to expect, let's use computational tools to compute the same diagrams. See below for the syntax of the commands used.

1. First, no noise:
  - (a) Generate a set of 500 points sampled from the sphere in  $\mathbb{R}^2$ , with no noise:

```
S=spheresampler(2,1,500,0);
```

- (b) Plot the points to check that they look like they're sampled from a circle.
- (c) Use the command

```
[I1 I0]= rca1pc(S,5);
```

to calculate the 0- and 1-dimensional persistence intervals for this sample. `I1` will then contain the 1-D persistence intervals.

- (d) Use the command `plotpersistencediagram(I1);` to plot the diagram. Compare your answer to the one above.

2. Now, let's make some noise:

- (a) Generate a set of 500 points sampled from the sphere in  $\mathbb{R}^2$ , with some noise:

```
S=spheresampler(2,1,500,0.2);
```

- (b) Plot the points to check that they look like they're sampled from a circle, with noise.
- (c) Use the command

```
[I1 I0]= rca1pc(S,5);
```

to calculate the 0- and 1-dimensional persistence intervals for this sample. `I1` will then contain the 1-D persistence intervals.

- (d) Use the command `plotpersistenceinterval(I1)` to plot the diagram. Compare your answer to the one above.

To sample points from an arbitrary dimensional sphere with added radial Gaussian noise, use the following command:

```
S = spheresampler(dimension, radius, numpoints, noisemultiplier)
```

To compute the 0- and 1-dimensional persistence diagrams of a point cloud, use the following command

```
[I1 I0] = rca1pc(data, distancebound)
```

# 1 Sliding Windows

## 1.1 Basics

As you have seen in lecture, signals are often analyzed using a sliding window. In this section, we will computationally apply these notions to harmonics (single and mixed) to see what we can find. But first, let's get a handle on how this all works with a simple example.

**Question** By hand first: Suppose you divide an interval into five evenly spaced ticks, and you are given the function values  $[1, 0, -2, 0, 1]$  on those ticks.

1. By hand, compute and plot the sliding window point cloud in  $\mathbb{R}^2$  for this data.

2. Sketch the 0- and 1-dimensional persistence diagrams for your point cloud, with the given distance bound:

(a) Distance bound = 1.

(b) Distance bound = 2.

(c) Distance bound = 3.

(d) Distance bound = 4.

To generate sliding window point clouds in Matlab (from function values), use the following command

```
SW = im2col(data,[1,dimension])'
```

Note especially the ' operator at the end of that last command. It transposes a matrix. Very useful.

**Question** Using the same function values and set of distance bounds given in the previous question, use Matlab to compute and plot the 0- and 1-dimensional diagrams for the sliding window in  $\mathbb{R}^2$ . Compare them to the diagrams you plotted. They should be identical. If not, go back to the previous question and figure out what went wrong.

## 1.2 Harmonics

1. To start, consider a simple harmonic of period 1,  $f(x) = \sin(2\pi x)$ .

(a) By hand!

- i. Using your knowledge from class, predict what the 2- and 3-dimensional sliding window point clouds should look like. Write a couple of sentences describing them here.
  
- ii. Sketch  $Dgm_1$  of these two point clouds here.

(b) Computer time!

- i. In a variable  $s$ , store 100 evenly spaced samples from this harmonic. You may use the `generate_harmonic` function, but you only really need the  $y$ -values from it, transposed.
- ii. Create a sliding-window point cloud of dimension 2 from your points using the `im2col` command.
- iii. Plot your point-cloud. Does it look like what you expected?
- iv. Compute the  $Dgm_1$  of your point cloud using `rca1pc`, and plot it using `plotpersistence`. Does it look like what you expected?
- v. Repeat the above questions for the 3-dimensional sliding window point cloud.

- (c) Analysis and higher dimensions!
- i. Compute the persistence of the 1-cycle you found in each of the two cases. Is it larger in the 2-D or 3-D case?
  - ii. Repeat the above computation for the 4-D sliding window point cloud (without plotting the point cloud....why?). You will find more than one 1-cycle. Plot the persistence diagram and determine which of the 1-cycles you should find the persistence of. Do so.
  - iii. Repeat for 5-D, etc.
2. Repeat the above for a mixed harmonic of periods 1 and 2. Investigate what changing the amplitude of either of the two harmonic components does. Record your observations here, along with relevant sketches.

## 2 Adding Noise

Investigate the effects of adding noise of various amplitudes to single harmonics and multiple harmonics.

### 3 Non-Periodic Functions

**Question** Take a few minutes away from the computer to think again about what features in the sliding window point clouds correspond to periodicity.

**Question** Suppose that instead of a periodic function, we consider a polynomial of degree  $n$  with all  $n$  real roots, all in the interval under consideration.

1. Generate data from such a function, on the interval  $[-2, 2]$ . Record here the commands you use.
2. Generate sliding window point clouds in a few dimensions, and plot their  $Dgm_1$ 's. What do you observe? Did you correctly predict what would happen? If not, go back to the previous question and consider what you may have missed.

## 4 Pure Noise

The command `rand()` generates a uniformly distributed random number between 0 and 1.

**Question** Write a command that will generate a uniformly distributed random number between  $-1$  and  $1$ .

As is often the case in Matlab, we can tell it to generate a matrix of random values with  $m$  rows and  $n$  columns by using the syntax `rand(m, n)`.

**Question** Generate 200 uniformly distributed random numbers from the interval  $[-2, 2]$ . Compute (not by hand) its 2-D sliding window point cloud and the corresponding  $Dgm_1$ . Repeat this for higher dimensions and another couple of pure noise data. Record your observations and relevant sketches here.



## 5.1 Elements of Classification

## Questions

- Page 9 of 10

## 5.2 Classification!

Once again, load the cell array from *signalmasterset.mat*. Using what you learned in this lab and the previous one, attempt to classify the data sets. The answers are in column 2, but don't look at that until you're done.