Laboratory 4: How to compare 1D arrays

The aims of today's lab are:

- Use the STL vector class to store an array of `float` values;
- Use built-in functions to compute the min, max, and sum of the array;
- Compare two arrays using the Sum of Absolute Errors (SAE) and the normalised cross-correlation (NCC).

Note, today you work using 1-D arrays. In Assignment 2, you have to do the same in 2-D with images.

Task 0: Using CMake

Same as usual, we will use CMake to make our lives easier.

Task 1: Min/Max/Sum/Average/Variance/Standard deviation

You have been given a ZIP file containing a small (incomplete class). For this task, you mostly have to modify `MyVector.cpp`. The methods you need to complete are:

1. `float getMinValue() const` (see previos labs)
2. `float getMaxValue() const` (see previos labs)
3. `float getSum() const` (see previos labs)
4. `float getAverage() const`
5. `float getVariance() const`
6. `float getStandardDeviation() const`

In the ZIP file, you also got 4 ASCII files:

- `y.mat`
- `y_quadriple.mat`
- `y_noise.mat`
- `y_negative.mat`

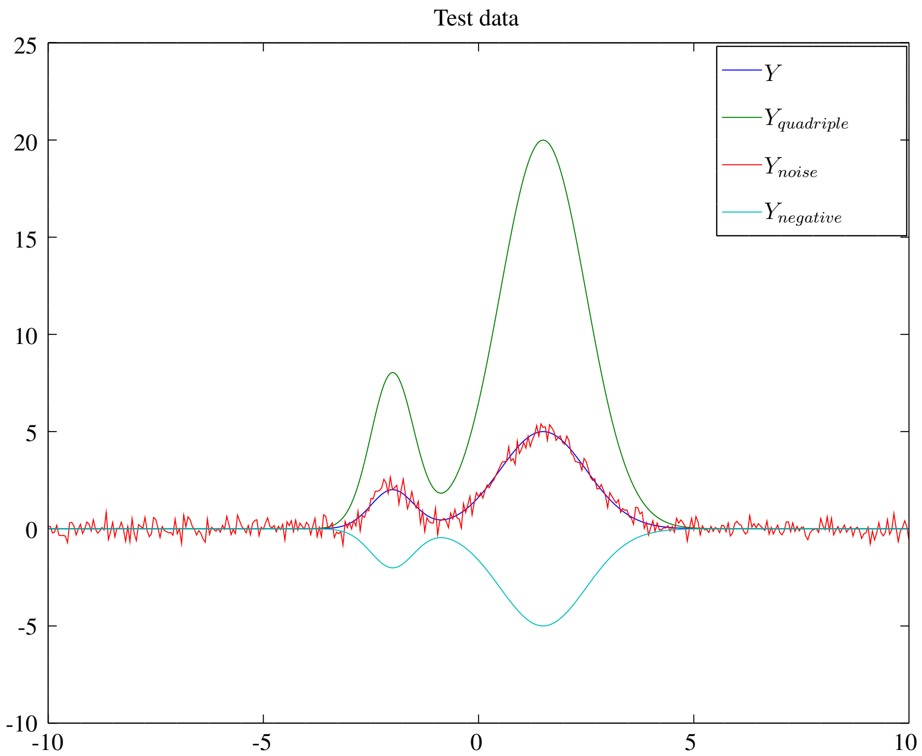

Figure 1: Test data from the ASCII files.

Table 1: Statistics about the test data from Figure 1

Test case	Min	Max	Sum	Average ($\overline{X}$)	Variance (σ^2)	Standard deviation (σ)
$\mathbf{Y}$	9.5774e-29	5.0000	300.80	0.75011	1.8371	1.3554
$\mathbf{Y_{quadriple}}$	3.8310e-28	20.000	1203.2	3.0005	29.393	5.4216
$\mathbf{Y_{noise}}$	-0.83714	5.3959	303.32	0.75641	1.9532	1.3976
$\mathbf{Y_{negative}}$	-5.0000	-9.5774e-29	-300.80	-0.75011	1.8371	1.3554

They contain test data that you can use to assess your code. Figure 1 shows the content of the files. You can load each file in independent instances of the class `MyVector`. Table 1 provides statistics about the test data from Figure 1. You can use them to compare the results of your computations.

Equations 1 to 3 show how to compute the average, variance and standard deviation of a vector $\mathbf{X}$ of N elements:

$$Average(\mathbf{X}) = \overline{X} = \frac{1}{N}\sum_{i=0}^{i<N}\mathbf{X}(i) \tag{1}$$

$$Variance(\mathbf{X}) = \sigma_X^2 = \frac{1}{N}\sum_{i=0}^{i<N}\left(\mathbf{X}(i) - \overline{X}\right)^2 \tag{2}$$

$$Standard\,deviation(\mathbf{X}) = \sigma_X = \sqrt{Variance(\mathbf{X})} \tag{3}$$

Task 2: How dissimilar two vectors are: the SAE

SAE stands for sum of absolute errors. It is also called sum of absolute distance (SAD), Manhattan distance, and L^1-norm. In statistics, it is used as a quantity to measure how far two vectors are from each other. The SAE between two vectors $\mathbf{Y_1}$ and $\mathbf{Y_2}$ of N element is:

$$SAE(\mathbf{Y_1}, \mathbf{Y_2}) = \sum_{i=0}^{N-1} |\mathbf{Y_1}(i) - \mathbf{Y_2}(i)| \tag{4}$$

Add the method as follows in your class:

```
float MyVector::SAE(const MyVector& aVector) const;
```

One of the main advantages of the SAE is that it is fast to compute. However, it has limitations. To test your computations, here are the results for:

- $SAE(\mathbf{Y}, \mathbf{Y_{quadriple}}) = 902.39$
- $SAE(\mathbf{Y}, \mathbf{Y_{negative}}) = 601.59$
- $SAE(\mathbf{Y}, \mathbf{Y_{noise}}) = 108.52$

$\mathbf{Y_{quadriple}}$ is equal to $4 \times \mathbf{Y}$. However, the SAE between $\mathbf{Y_{quadriple}}$ and $\mathbf{Y}$ is the largest. $\mathbf{Y_{negative}}$ is equal to $-\mathbf{Y}$. However, the SAE between $\mathbf{Y_{negative}}$ and $\mathbf{Y}$ is the second largest. We can conclude that even if SAE is commonly used in imaging it may not provide a good error metrics in some cases.

Task 3: How similar two vectors are: the NCC

NCC stands for normalised cross-correlation. The normalisation in NCC addresses the limitation highlighted in our tests. The formula is:

$$NCC(\mathbf{Y_1}, \mathbf{Y_2}) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{(\mathbf{Y_1}(i) - \overline{Y_1})(\mathbf{Y_2}(i) - \overline{Y_2})}{\sigma_{Y_1} \sigma_{Y_2}} \tag{5}$$

- $NCC(\mathbf{Y_1}, \mathbf{Y_2})$ = 1, if $\mathbf{Y_1}$ and $\mathbf{Y_2}$ are fully correlated (e.g. $\mathbf{Y_1} = \alpha\mathbf{Y_2}$);
- $NCC(\mathbf{Y_1}, \mathbf{Y_2})$ = -1, if $\mathbf{Y_1}$ and $\mathbf{Y_2}$ are fully anti-correlated (e.g. $\mathbf{Y_1} = -\alpha\mathbf{Y_2}$);
- $NCC(\mathbf{Y_1}, \mathbf{Y_2})$ = 0, if $\mathbf{Y_1}$ and $\mathbf{Y_2}$ are fully uncorrelated (they are unrelated).

Often the NCC is expressed as a percentage. With our examples, we get:

- $NCC(\mathbf{Y}, \mathbf{Y_{quadriple}}) = 1 = 100\%$
- $NCC(\mathbf{Y}, \mathbf{Y_{negative}}) = -1 = -100\%$
- $NCC(\mathbf{Y}, \mathbf{Y_{noise}}) = 0.97 = 97\%$

Summary

Today, you saw how to compare 1-D vectors in two different ways:

- SAE
- NCC

You will adapt them to your next assignment.