

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`

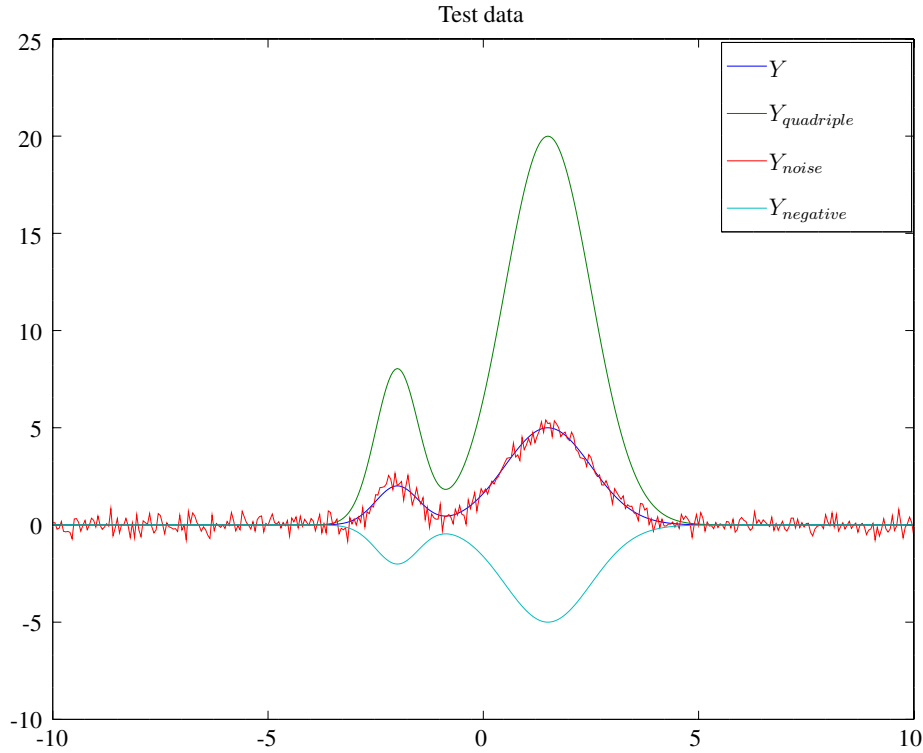


Figure 1: Test data from the ASCII files.

Table 1: Statistics about the test data from Figure 1

Test case	Min	Max	Sum	Average (\bar{X})	Variance (σ^2)	Standard deviation (σ)
Y	9.5774e-29	5	300.795	0.750113	1.8325	1.3537
Y_{quadriple}	3.83096e-28	20	1203.18	3.00045	29.3201	5.4148
Y_{noise}	-0.837137	5.39592	303.319	0.756407	1.94837	1.39584
Y_{negative}	-5	-9.5774e-29	-300.795	-0.750113	1.8325	1.3537

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:

$$\text{Average}(\mathbf{X}) = \bar{X} = \frac{1}{N} \sum_{i=0}^{i < N} \mathbf{X}(i) \quad (1)$$

$$\text{Variance}(\mathbf{X}) = \sigma_X^2 = \frac{1}{N} \sum_{i=0}^{i < N} (\mathbf{X}(i) - \bar{X})^2 \quad (2)$$

$$\text{Standard deviation}(\mathbf{X}) = \sigma_X = \sqrt{\text{Variance}(\mathbf{X})} \quad (3)$$

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:

$$SAE(\mathbf{Y}_1, \mathbf{Y}_2) = \sum_{i=0}^{N-1} |\mathbf{Y}_1(i) - \mathbf{Y}_2(i)| \quad (4)$$

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:

- $SAE(\mathbf{Y}, \mathbf{Y}) = 0$
- $SAE(\mathbf{Y}, \mathbf{Y}_{\text{quadruple}}) = 902.386$
- $SAE(\mathbf{Y}, \mathbf{Y}_{\text{negative}}) = 601.591$
- $SAE(\mathbf{Y}, \mathbf{Y}_{\text{noise}}) = 104.505$

$\mathbf{Y}_{\text{quadruple}}$ is equal to $4 \times \mathbf{Y}$. However, the SAE between $\mathbf{Y}_{\text{quadruple}}$ and \mathbf{Y} is the largest. $\mathbf{Y}_{\text{negative}}$ is equal to $-\mathbf{Y}$. However, the SAE between $\mathbf{Y}_{\text{negative}}$ 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:

$$NCC(\mathbf{Y}_1, \mathbf{Y}_2) = \frac{1}{N} \sum_{i=0}^{N-1} \frac{(\mathbf{Y}_1(i) - \bar{Y}_1)(\mathbf{Y}_2(i) - \bar{Y}_2)}{\sigma_{Y_1} \sigma_{Y_2}} \quad (5)$$

- $NCC(\mathbf{Y}_1, \mathbf{Y}_2) = 1$, 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:

- $NCC(\mathbf{Y}, \mathbf{Y}) = 0.999996 = 99.9996\%$
- $NCC(\mathbf{Y}, \mathbf{Y}_{\text{quadruple}}) = 0.999996 = 99.9996\%$
- $NCC(\mathbf{Y}, \mathbf{Y}_{\text{negative}}) = 0.999996 = -99.9996\%$
- $NCC(\mathbf{Y}, \mathbf{Y}_{\text{noise}}) = 0.97188 = 97.188\%$

Summary

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

- SAE
- NCC

You will adapt them to your next assignment.