Essential Mathematics for Data Scientists

Assessment 2C: Image comparison code workbook

(In preparation for assessment commencing Week 4)

This exercise has you run MATLAB code that implements the theory presented in section 4 of Week 3 (Application of SVDs – image comparison). The goal is to have you recognise the various elements of the code and the outputs it produces. This is in preparation for assessment in Week 4, which will draw upon these ideas.

The implementation of the image comparison is given in the MATLAB file `compareImages.m` that you need to download from the assessment page in LearnJCU together with the test images folder.

This implementation requires the images to be of the same size. We have ensured this is the case by resizing all of our test images beforehand.

Comparison between like images

In order to test the `compareImages` function, we can apply it to two cases where we expect to have a high degree of similarity. This has been done in the first section of code in the MATLAB file `similarity.m.` We compare an image with:

a) itself.

b) An image showing the same type of object.

Here we choose the apple image `apple-9.png`, and compared it with `apple-13.png`. We expect a) to return 1, due to the normalisation and b) to return a large value (close to 1). The MATLAB output is:

```
a) The similarity value for the same image is:

   ans =

         1.0000

b) The similarity value for the two apple images is

   ans =

         0.7125
```

Comparison between all images

We can now use MATLAB to read in each image file consecutively to apply the comparison technique between the first apple image and each of the other images. Before you run this code it would be beneficial for you to predict which pairs of images have high similarity ratings. Can you predict an order of pairs from highest to least similarity?

The required code appears in the second half of the file `similarity.m`. It generates an array with each row and column corresponding to one of the images – thus along the diagonal of the array each image is compared with itself. What patterns can you see in the array? Do they match your expectations from the predictions you made before you ran the code?

Write your answers to these questions, next to a copy of the code and its output, in your code workbook.