CM20220: Fundamentals of Machine Learning

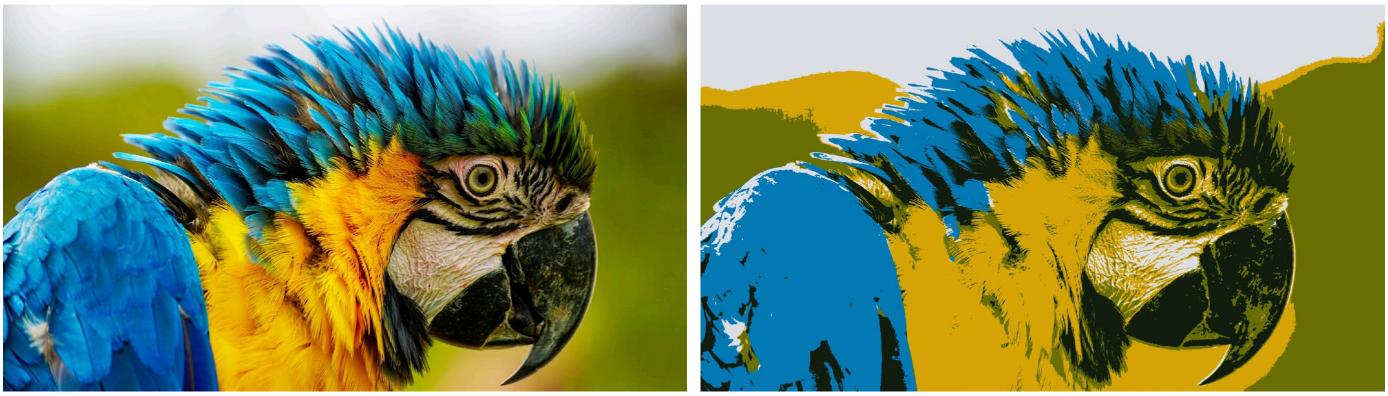
Lab Sheet 2: Unsupervised Learning

The tasks in this lab sheet focus on problems where the training data is not labelled with the class to which it belongs. There are two distinct problems. The first has a single task. The second is split into two tasks. All three tasks involve replacing the use of an existing module implementation of a machine learning task with your own version. There are two Jupyter Notebooks available on Moodle that you are expected to download and modify. The modified versions are what you will demonstrate and then upload to Moodle.

Task 1: Image Segmentation using kMeans (4 Marks)

You will need to download the SegmentationKMeans Jupyter Notebook from Moodle to undertake this task.

The supplied code loads an image and turns each pixel into a feature vector. Each component colour, red, green and blue becomes a feature. It then uses the sklearn implementation of kMeans to cluster the pixels. Once clustered, it computes the median value of each component within the clusters. A new image is created that replaces all pixels in a cluster with the median values. The overall effect of this segmentation is known as *posterisation* and results in transformation of images like that shown in Figure 1.



(a)

(b)

Figure 1: Example of Posterisation

Your task is to replace the module implementation of kMeans with one you create from scratch. You are only required to replace the cell indicated in the notebook. Your new code does not need to be kept in a single cell, you may add further cells if you wish.

kMeans works in the following way:

1. Specify a number of clusters.
2. Create a random centroid for each cluster.
3. For each data point identify the closest centroid and assign it to the corresponding cluster.
4. Compute a new centroid for each cluster based on the current cluster members.
5. Loop back to step 3 until the assignment of clusters is stable.

Task 2: Linear Regression (3 Marks)

You will need to download the RegressionRANSAC Jupyter Notebook from Moodle to undertake this task. This notebook generates some (constrained) random data to which it applies linear regression. The data consists of a mix of inlier and outlier data. Your task is to replace the cell that uses the module implementation of simple linear regression with one that you create from scratch yourself. You are only required to replace the cell indicated in the notebook. Your new code does not need to be kept in a single cell, you may add further cells if you wish.

Linear Regression works in the following way:

If we have data of the form,

$$D = \{(x_1,y_1), (x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)\}$$

This looks more like the supervised data we've seen previously except that $y_i \in \mathbb{R}$ rather than being a class. The goal is to come up with a model of the data that allows us to compute a value of *y* for any input *x*. If we decide that *y* is related to *x* via the straight line $y_i = mx_i + c$, then the task is to find the values of *m* and *c*. To measure how well a given (m,c) fits the data we need an objective function:

$$E(m,c) = \sum_{i=1}^{N} (mx_i + c - y_i)^2$$

We then look to find the values that minimise this function. Fortunately for us, there is a closed form solution for this particular case. See the lectures for details.

Task 3: Linear Regression using RANSAC (3 Mark)

You should extend the RegressionRANSAC Jupyter Notebook from Moodle that you downloaded for Task 2. In addition to simple linear regression this notebook also performs linear regression using RANSAC. Your task is to replace the cell that implements the RANSAC based linear regression using the module with one that you create yourself from scratch. You are only required to replace the cell indicated in the notebook. Your new code does not need to be kept in a single cell, you may add further cells if you wish. In addition to displaying the predicated values you should also indicate if an input point is considered to be an inlier or outlier.

RANSAC works as follows:

1. Pick two data points.
2. Compute parameters, m and c.
3. Classify remaining data points as either outliers or inliers.
4. Repeat from Step 1, N times.
5. Select the parameter pair with the fewest outliers.

Marking Guidance

During any lab session, when you've completed a task demonstrate it successfully to a tutor and they will record the completion of the task. Tutors will not record completion of tasks for this lab sheet after its deadline unless a student has an extension agreed by their Director of Studies.

Tutors will give priority to students who are assigned to a lab session and highest priority to those for whom it is the final allocated lab session before the deadline and who wish to demonstrate their code.

If you are unable to successfully demonstrate any of the tasks, you may still receive partial credit, not exceeding half marks for the task. This will be assessed on the basis of the uploaded code, after the deadline and not in the lab.

You *must* upload your code in the form of the Jupyter Notebooks for all tasks attempted to Moodle by the deadline for this assignment or by any agreed extension deadline. You will not be allocated any marks from the demonstrations if you have not done so. There is no specific name you need give it, and it's helpful if you are seeking partial credit to upload the version than contains the outputs.

The deadline for all three tasks of this lab sheet is **Friday 27th March 8pm**.

Parrot Image by Susanne Jutzeler released under Pexels License.