CM20220: Fundamentals of Machine Learning

Lab Sheet 2: Unsupervised Learning

Deadline: 16th April 2021, 8pm.

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 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 do not need to create a class or module. Functions

are sufficient. 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. You do not need to create a class or module. Functions are sufficient. In addition to displaying the predicted values you should also indicate if an input point is 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, the smallest error.

Lab Support

Tutors will be available during the LOIL sessions to help you to complete the labs. This will be done via Microsoft Teams. You can also ask for 1 to 1 help in order to share your screen without other students seeing your work.

Marking Guidance

The deadline for all three tasks of this lab sheet is **Friday 16st April 2020, 8pm**.

You must upload your Jupyter Notebooks containing all the tasks attempted to Moodle for this unit by the deadline for this assignment or by any agreed extension deadline. They should contain the output embedded in them. Failing to do so will mean you do not receive the marks for the work. Marks will be given for each task successfully demonstrated. The provided notebooks give you an indication of the result you should be expecting. Tasks that are incomplete or produce the wrong answer will receive no marks. An allowance will be made for rounding errors in calculations. You must upload a version of your notebook that includes the output of running the code. We only expect to run your notebooks in exception case. No specific name is need for the files uploaded. You do not need to upload the parrot image.

Parrot Image by Susanne Jutzeler released under Pexels License.