COMP9517

Lab 4

T3, 2020

The lab files should be submitted online.

Instructions for submission will be posted closer to the deadline.

Deadline for submission is Week 8, Tuesday, Nov 3rd, 2020, 11:59:59, AEST. Keeping with current school policy, note that the deadline is now close to NOON and NOT at midnight.

Objectives: The goal of this lab is to become familiar with the training/testing methods for pattern recognition / machine learning algorithms.

Materials: You are required to use OpenCV 3+ with Python 3+. Jupyter notebook files are preferred for submitting your code.

Submission: The Assessment question is assessable **after the lab** and is **worth 2.5% of the total course marks**. Submit your code and results as a Jupyter notebook in a zip file via WebCMS3 by the deadline.

1 Pattern Recognition

Pattern Recognition is the classification of data based on already existing knowledge or on statistical information extracted from patterns and/or their representations. It finds application in several areas like Speech recognition, data compression, image analysis and many others. In this lab we will explore K-Nearest Neighbour (kNN) classifier, a Support vector machines (SVM) classifier and a Random forest classifier to recognise patterns in images and classify them into appropriate classes based on these patterns. The sub-sections below will provide some basic information about each of these classifiers.

1.1 K-Nearest Neighbours (KNN)

The KNN algorithm is very simple and very effective. The model representation for KNN is the entire training data set. Predictions are made for a new data point by searching through the entire training set for the K most similar instances (the neighbours) and summarizing the output variable for those K instances. For regression problems, this might be the mean output variable, for classification problems this might be the mode (or most common) class value. The trick is in how to determine the similarity between the data instances.

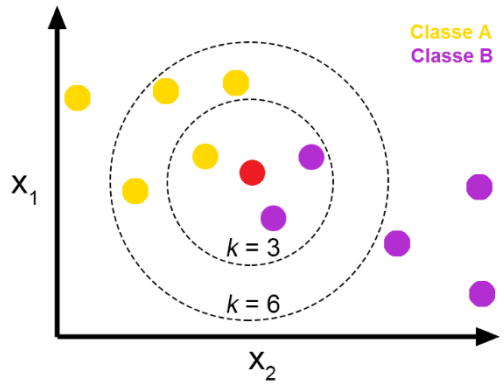

Figure 1: A 2-class KNN example with 3 and 6 neighbours (Image from [1]).

To make predictions we need to calculate the similarity between any two data instances. This way we can locate the K most similar data instances in the training data set for a given member of the test data set and in turn make a prediction. For a numeric data set, we can directly use the Euclidean distance measure. This is defined as the square root of the sum of the squared differences between the two arrays of numbers.

Implementation: Refer to the scikit-learn documentation for more information. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

1.2 Support vector machines (SVM) Classifier

SVM is a supervised machine learning algorithm that can be used for both classification and regression. Each data item is plotted on an n-dimensional (see Figure 2) space (n is the number of features per data point), and classification is done by finding the best hyper-plane that differentiates two classes. There can be many possible hyperplanes that separate two classes of data points, but the objective here is to find that plane that maximises the margin (distance between data points of the two classes)

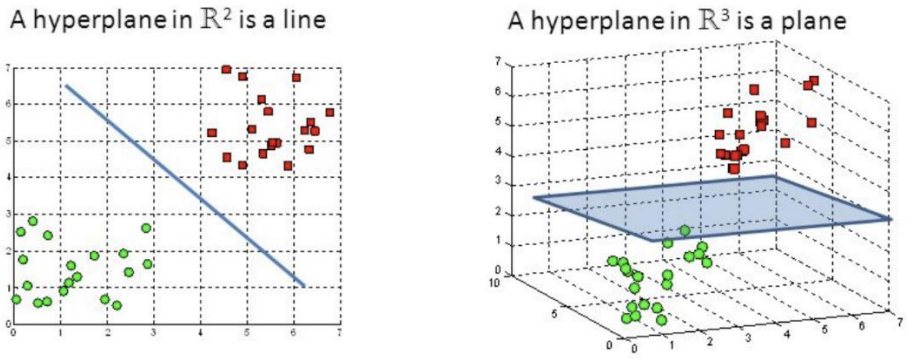

Figure 2: Hyperplanes in 2D and 3D feature space (Image from [2])

Implementation: Refer to the scikit-learn documentation for more information. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html and https://scikit-learn.org/stable/modules/svm.html#classification

1.3 Random Forest Classifier (RF)

As the name implies this classifier consist of several individual decision trees [3] that operate as an ensemble. Each individual decision tree makes a class prediction and the class with the maximum votes becomes the model prediction. Many uncorrelated models operating together is making better predictions than individual models.

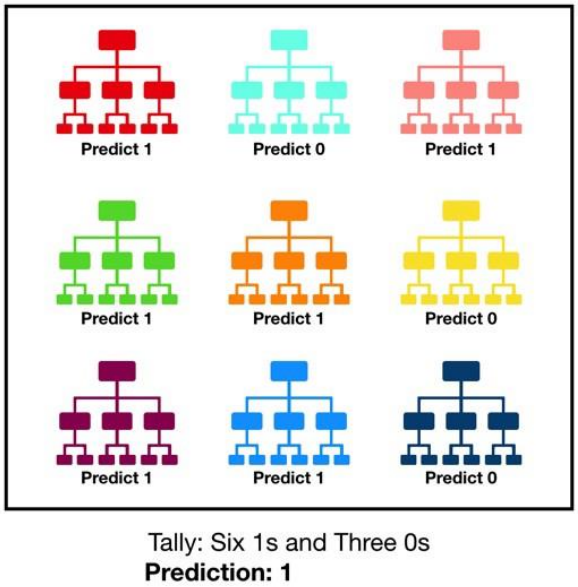

Figure 3: Random Forest making predictions (Image from [4])

Implementation: Refer to the scikit-learn documentation for more information. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

2 ASSESSMENT QUESTION (2.5 marks): IMAGE CLASSIFICATION

Develop a program to perform image classification on **scikit learn's digits** dataset. Classify the images from the data set using the three classifiers mentioned above and compare the classification results.

2.1 Scikit learn's Digits Data Set

It has a total of 1797 images and their corresponding labels. The dataset contains low resolution 8*8 images of digits ranging from 0 to 9 and was designed to test classification algorithms.

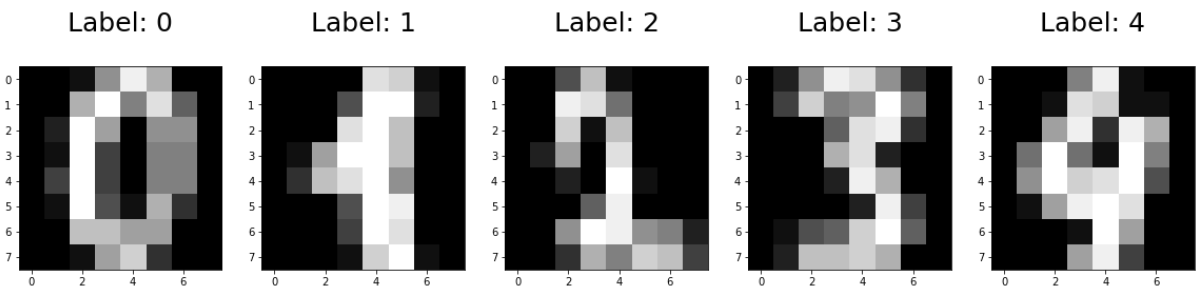

Figure 4: Samples from the dataset, with actual labels displayed on top.

2.2 Steps to Follow

1. Import relevant packages
2. Load the images
    a. Optional: Familiarize yourself with the dataset. For example, find out how many images and labels there are, the size of each image, and display some of the images and their labels.
3. Split the images using sklearn's train_test_split() with a test size anywhere from 20% to 30% (inclusive).
4. CLASSIFICATION: For each of the classifiers perform the following steps:
    a. Initialize the classifier model.
    b. Fit the model to the training data.
    c. Use the trained/fitted model to evaluate the test data.
5. EVALUATION: For each of the three classifiers, evaluate the digit classification performance by calculating the accuracy, average-recall and confusion matrix
    a. Experiment with the number of neighbours used in the KNN classifier to find the best number for this data set. You can adjust the number of neighbours with the n_neighbours parameter (the default value is 5).
6. Print the accuracy and average recall of all three classifiers and the confusion matrix of the best-performing classifier. (Use the default hyper-parameter values for SVM and RF, and for KNN, use the best number of neighbours that you found)

3 REFERENCES

[1]. https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d
[2]. https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47
[3]. https://en.wikipedia.org/wiki/Decision_tree_learning
[4]. https://towardsdatascience.com/understanding-random-forest-58381e0602d2