Menoufia University

Assignment 1

Image Labelling - Linear Classification

This assignment will give you a chance to familiarize yourself with Jupyter, scikit-learn and other ML tools. You will use the **Chars74K dataset** of characters of the English alphabet. In particular, you will be working with the images of characters and not with handwritten letters. The dataset can be download from http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/ at the Center of Vision, Speech and Signal Processing at the University of Surrey, Uk. You are looking for an archive called **EnglishImg.tgz**, gzipped tar archive which you can extract with, e.g., gzip or 7-zip.

This archive contains a Bmp folder in EnglishImg/English/Img/GoodImg where you can find 62 subdirectories for each lower and upper case character in the latin alphabet (without accents) and for the digits. The color images are of different size and contains background. The archive also contains binary foreground-background masks in the Folder Msk for all images.

In this assignment, you will train binary and multiclass linear classifiers available from scikit- learn on this dataset and evaluate the performance of these classifiers with various metrics.

1.1 Getting Started

You will need to use the scikit-learn image processing tools to read and process the images. The images are stored in a directory structure with one directory per character or digit.

For this assignment, we want to work with characters **'o','q','G' and digit '8'** which can be found in subdirectories Sample051, Sample053, Sample017 and Sample009, respectively. You will need to work with all images of 'o','q','G' and '8' from these four subdirectory and organize them suitably.

You will need to load the images into numpy matrices. You will need to generate a list of filenames for the files to read. You can achieves this with hardcoding a loop for the filenames according to the above filename rules, or by reading the directory and file structure using the os package in python.

The next step is reading and preprocessing the images. For this part, you will need to use the skimage package. Amongst others, the following commands will be helpful: skimage.io.imread, skimage.color.rgb2gray, and skimage.resize. **Make sure to resize the images to a common size**. For questions 1.2 and 1.3, please work with **grayscale images**.

1.2 Binary Classifiers

Train **two logistic regression classifers** to distinguish grayscale images of the **letters 'o' and 'q'**, as well as of **'G' and '8'**. Organize your data into training and test set. State clearly in your Jupyter notebook what split you are using and give the number of images for each of the three sets. Also consider careful how to compose these sets, e.g., selecting images at **random** vs. some form of **stratified** sampling. Find and print the **confusion matrix**. **Without using any scikit-learn function calculate** the **accuracy**, the **recall** and the **precision** of your two classifiers based on the confusion matrix of the training data, as well as on the testing data. Print the **ROC curves** for each. Annotate your Jupyter notebook with an explanation of the results. Without an explanation, you will not receive full marks.

1.3 Multiclass Classifier [4]

Train a logistic regression classifier but this time to classify all grayscale images into multiple (4) classes: characters 'o','q','G' and digit '8'. Re-organize your data into training and test set. State clearly in your Jupyter notebook what split you are using and give the number of images for each of the three sets. Also consider careful how to compose these sets, e.g., selecting images at random vs. some form of stratified sampling. Find and print the confusion matrix. **Without using any scikit-learn function calculate** the accuracy, the average recall, the average precision and the F1 score of your two classifiers based on the confusion matrix of the training data, as well as on the testing data. Print the ROC curves for each and the accuracy. In your Jupyter notebook, discuss briefly how the classifier perform on this task, e.g., are there easier or more difficult classes, do certain classes tend to be confused etc. Use the metrics in your argument.

2 Submission

You will need to submit your solution in a Jupyter file. Make sure you have run all the cells. All your comments must be embedded in the Jupyter file.