Statistical Machine Learning
Class Project

In this class project, we will systematically implement and examine the three major categories of machine learning techniques of this course, including supervised learning, un-supervised learning and deep learning.

Part 1: Density Estimation and Classification

Project Overview:

In this part, you need to first perform parameter estimation for a given dataset (which is a subset from the MNIST dataset). The MNIST dataset contains 70,000 images of handwritten digits, divided into 60,000 training images and 10,000 testing images. We use only images for digit “0” and digit “1” in this question.

Therefore, we have the following statistics for the given dataset:

Number of samples in the training set: "0": 5923 ;"1": 6742.
Number of samples in the testing set: "0": 980; "1": 1135

You are required to extract the following two features for each image:

1) The average brightness of the image
2) The average of the variances of each rows of the image.

We assume that these two features are independent, and that each image (represented by a 2-D features vector) is drawn from a 2-D normal distribution.

We also further assume that the prior probabilities are the same (P(Y=0) = P(Y=1) =0.5), although you may have noticed that these two digits have different numbers of samples in both the training and the testing sets.

You may go to the original MNIST dataset (available here http://yann.lecun.com/exdb/mnist/) to extract the images for digit 0 and digit 1, to form the dataset for this project. To ease your effort, we have also extracted the necessary images, and store them in “.mat” files. You may use the following piece of code to read the dataset:

```
import scipy.io
Numpyfile= scipy.io.loadmat(‘matlabfile.mat’)
```

The specific algorithmic tasks you need to perform for this part of the project include:

1) Extracting the features and then estimating the parameters for the 2-D normal distribution for each digit, using the training data. Note: You will have two distributions, one for each digit.
2) Use the estimated distributions for doing Naïve Bayes classification on the testing data. Report the classification accuracy for both “0” and “1” in the testing set.

Algorithms:

MLE Density Estimation, Naïve Bayes classification

Resources:

A subset of MNIST dataset, download either from http://yann.lecun.com/exdb/mnist/ (requiring you to extract data corresponding to digit 0 and digit 1 only), or from the .mat files provided.

Workspace:

Any Python programming environment.

Software:

Python environment.

Language(s):

Python. (MATLAB is equally fine, if you have access to it.)

Required Tasks:

1. Write code to extract features for both training set and testing set.
2. Write code to estimate/compute the parameters of the relevant distributions.
3. Write code to implement the Naïve Bayes Classifier and use it produce a predicted label for each testing sample.
4. Write code to compute the classification accuracy.
5. Submit a short report summarizing the results, including the estimated parameters of the distributions and the final classification accuracy.

Optional Tasks:

1. Repeat the experiments for different pairs of digits.
2. Consider doing multi-class classification.

Optional tasks are to be explored on your own if you are interested and have extra time for them. No submission is required on the optional tasks. No grading will be done even if you

submit any work on the optional tasks. No credit will be assigned to them even if you submit them. (So, please do not submit any work on optional tasks.)

What to Submit and Due Dates:

1. Code:
    - Acceptable file types are .py/.m or .zip.
    - If you have only one file, name the file to be main.py or main.m for matlab users, and submit it.
    - If you have multiple code files, please name the main file as main.py and name other files properly based on its content; Similarly, for matlab users, you should have only one main.m and other relevant .m files. Next, zip all the files and submit Code.zip file.
    - Documentation comment is important and required.
        - Please add comment properly to explain what you do for each task.
        - You do not need to explain every line. But for each task, a brief introduction is required before the code segment. To be specifically, in the comment part, you should explain what you do for the task, what is the input, what is the output, what is the meaning of each variable you use and what is the meaning of the functions you use. If there is no comment and the code is unreadable, no score will be given.
2. Report:
    - Acceptable file types: .pdf or .doc/docx.
    - Length of the report: 2-5 A4 pages.
    - Content: (The following must be included)
        - The formula you used to estimate the parameters, and the estimated values for the parameters.
        - The expression for the estimated normal distributions.
        - Explanation on how the distributions are used in classifying a testing sample(i.e., explaining how you implement the Naïve Bayes Classifier).
        - The final classification accuracy for both “0” and “1” for the testing set.

The code and report are due at the end of Week 3.

Evaluation criteria:

Working code; Correct final results (for both estimated parameters and the classification results).

Code

- 3 points - Correctly extracts features for both training and testing set.
- 3 points - Correctly estimates and computes the parameters of the relevant distributions

- 3 points - Correctly implements the Naive Bayes Classifier and uses it to produce a predicted label for each testing sample
- 1 point - Correctly computes the classification accuracy
- 4 points - Each of these code parts is correctly documented in comments

Report:

- 2 points - Formula for estimating the parameters and their estimated values
- 1 points - Expression for the estimated normal distributions
- 2 points - Explains how the distributions are used in classifying a testing sample
- 1 points - The final classification accuracy for both “0” and “1” for the testing set