Assignment 1

Due on 19 March, 2021 (23:59:59)

Click here to accept your Assignment 1

Instructions. There are two parts in this assignment. The first part involves a series of theory questions and the second part involves coding. The goal of this problem set is to make you understand and familiarize with kernel regression algorithm.

Part I: Theory Questions

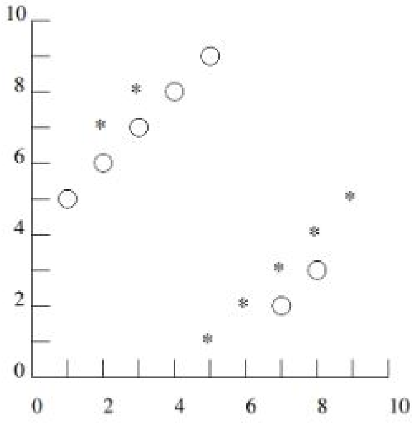
k-Nearest Neighbor Classification

1. Let k-NN(S) denote the k-Nearest Neighbor classifier on a sample set S, containing samples from 2 classes (positive, negative).

   (a) Show that if in both 1-NN(S_1) and 1-NN(S_2) the label of point x is positive, then in 1-NN($S_1 \cup S_2$) the label of x is positive.

   (b) Show an example such that in both 3-NN(S_1) and 3-NN(S_2) the label of x is positive, and in 3-NN($S_1 \cup S_2$) the label of x is negative.

2. One of the problems with k-nearest neighbor learning is how to select a value for k. Say you are given the following data set. This is a binary classification task in which the instances are described by two real-valued attributes (* and ∘ denote positive and negative classes, respectively).



   (a) What value of k minimizes the training set error for this data set, and what is the resulting training set error? Why is training set error not a reasonable estimate of test set error, especially given this value of k?

(b) What value of k minimizes the leave-one-out cross-validation error for this data set, and what is the resulting error? Why is cross-validation a better measure of test set performance?

(c) Why might using too large values k be bad in this dataset? Why might too small values of k also be bad?

(d) Sketch the 1-nearest neighbor decision boundary for this dataset.

Linear Regression

1. Suppose you are given m=23 training examples with n=5 features (excluding the additional all-ones feature for the bias term, which you should add).
   Recall that the closed form solution of linear regression is $\theta = (X^T X)^{-1} X^T y$. For the given values of m and n, what are the dimensions of X, y, θ in this equation?

2. Suppose you have m=50 training examples which are represented with n=200,000 dimensional feature vectors. You want to use multivariate linear regression to fit paremeters θ to our data. Should you prefer gradient descent or the closed form solution?

3. Which of the following are valid reasons for using feature scaling?

   (a) It speeds up solving for θ using the normal equation.

   (b) It prevents the matrix $X^T X$ (used in the normal equation) from being non-invertable (singular/degenerate).

   (c) It speeds up gradient descent by making it require fewer iterations to get to a good solution.

   (d) It is necessary to prevent gradient descent from getting stuck in local optima.

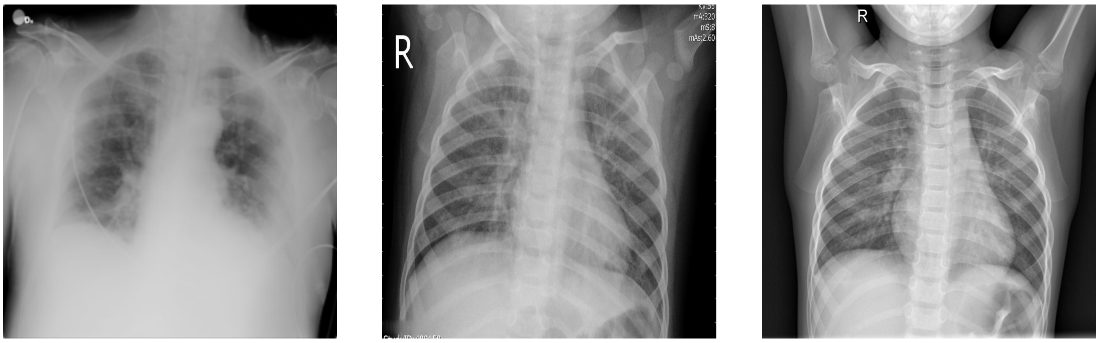
PART II: Classification of Images

In this part of the assignment, you will implement a nearest neighbor algorithm to detect Covid-19 disease from images. You will also extend your implementation as weighted KNN algorithm.

A dataset is provided for your training phase. Test images will be provided later and announced from Piazza group. Since test images will be provided later, you should use a subset of the training set to validate the performance of your model. In other words, you should split your training dataset into two set; training set which will be used to learn model, and validation set which will be used to measure the success of your model. You can use k-fold cross-validation method which is explained in the class.

Dataset

- You can download the dataset from given link.



(a) patients with Covid-19 (b) ptients with Pneumonia (c) Normal Subjects

Figure 1: Example of CXR images of (a) patients with Covid-19 (b) patients with Pneumonia (c) Normal subjects

- Dataset consists of three type CXR images for COVID-19, positive cases, Viral Pneumonia images and Normal CXR images.
- There are 1200 COVID-19 positive case, 1345 viral pneumonia and 1341 normal images in the dataset. Figure 1 shows sample images belonging to these three groups.

Features

- There is no limitation about features. You can use any feature that you think it's proper for your classification assignment. Some features are listed below:
  - Tiny images: You can resize images to a very small size.
  - Shape features: Shape is an important and powerful feature for image classifcation. You can use shape information extracted using histogram of edge detection. Edge information in the image is obtained by using the Canny edge detection.
  - Texture features: The texture feature is extracted usually using filter based method. The Gabor filter is a frequently used filter in texture extraction.
- You can use more than one feature by concatenating them.

Steps you need to follow:

1. Extract features for each image in training set (Canny, Gabor, etc.).
2. For each given test sample,
   (a) predict its class using k-NN.
   (b) predict its class using weighted k-NN.

3. Finally compute accuracy of your model to measure the success of your classification method for each setting you have used:

$$\textbf{Accuracy} = 100 * (\frac{\textbf{number of correctly classified examples}}{\textbf{number of examples}}) \quad (1)$$

   You will report mean accuracy by averaging your accuracy results for k folds (cross-validation).

4. **Error Analysis**

   - Find a few miclassified images and comment on why you think they were hard to classify.
   - Compare performance of different feature choices and investigate the effect of important system parameters (number of training images used, k in kNN, etc.). Wherever relevant, feel free to discuss computation time in addition to classification rate.

5. **(Bonus) Deep Image Features**
   Extract deep image features of VGG-19 net for classification of images.

Submit

You are required to submit all your code (*all your code should be written in Jupyter notebook* long with a report in ipynb format (should be prepared using Jupyter notebook). The codes you will submit should be well commented. Your report should be self-contained and should contain a brief overview of the problem and the details of your implemented solution. You should explain your choices and their effects to the results. You can include pseudocode or figures to highlight or clarify certain aspects of your solution. You can also include a table to report your results like:

Feature	Accuracy
Gabor	0.5
Canny	0.08
Attribute	0.3

Finally, prepare a ZIP file named name_surname_pset1.zip containing,

- report.ipynb (Jupyter notebook file containing your report)
- code/ (directory containing all your codes as Python file .py)

The ZIP file will be submitted via Github Classroom. Click here to accept your Assignment 1

NOTE: To enter the competition, you have to register kaggle in Class with your department email account. The webpage of the competition will be announced later. Top 5 assignment will earn extra points.

Grading

- Code (70): k-NN: 20, Weighted k-NN: 40, Part5 (Bonus): 10
- Report(40): Theory part: 12 points, Analysis of the results for prediction: 28 points.

Late Policy

You may use up to four extension days (in total) over the course of the semester for the three problem sets you will take. Any additional unapproved late submission will be weighted by 0.5. You have to submit your solution in (rest of your late submission days + 4 days), otherwise it will not be evaluated.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.