CAP 5771 / CIS 4930: Introduction to Data Mining: Fall 2022

Assignment 4

Submission Deadline: Wednesday, November 23, 2022, 11:59pm (through Canvas)

Instructions:

- All the questions are compulsory for the graduate students. For the undergraduate students, Problem 3 is optional. You are welcome to try it out, but you will not receive any extra credit / bonus points for this question.
- **You have to use Matlab or Python for this assignment**

Problem 1 (10 points)

In a **multi-class classification** problem, there are multiple classes in the dataset, but each data sample can belong to only one class. **Multi-label classification** is a generalization of multi-class classification, where each data sample can belong to multiple classes simlutaneously. For instance, consider the problem of classifying an outdoor image of a scene. Suppose the possible classes are beach, mountain, field and sunset. It is possible for a particular image to contain both beach and mountain or beach, mountain and sunset all together. The objective of multi-label learning is to predict all the classes present in a data sample.

The *Scene* dataset consist of 2407 images of an outdoor scene, where each image is represented by a feature vector of dimesnion 294. Also, there are 6 classes in the problem and an image can belong to one or more of the 6 classes. The dataset has been divided into a training set (with 1500 samples) and a test set (with 907 samples). Each row of X_train and X_test denotes a sample and each column denotes a feature. Each row of y_train (y_test) denotes the labels of the corresponding training (testing) sample, where 1 means the class is present and 0 means the class is absent. For instance, in the training set, sample 462 belongs to classes 4 and 5.

One strategy to solve a multi-label learning problem is to train an SVM separately for each class. To predict a test sample, each SVM is applied separately on it. A positive output indicates that the corresponding class is present and a negative output indicates that it is absent. The accuracy in the multi-label setting is computed as follows:

i) For each test sample, compute the quantity

$$A = \frac{|T \cap P|}{|T \cup P|}$$

where T is the true class label vector of a test sample and P is the predicted class label vector. The numerator denotes the number of entries in which both T and P have a value 1 (positive prediction) and the denominator denotes the number of entries in which at least one of T and P have a value 1

ii) Average this value over all the test samples to compute the final test accuracy

Train an SVM classification model on the training set and test on the test set. Report the percentage accuracy on the test set using the following classification models: (i) SVM with polynomial kernel with parameter 2 and (ii) SVM with Gaussian kernel with parameter 2.

Problem 2 (10 points)

The *seeds* dataset contains measurements of geometrical properties of kernels belonging to different varieties of wheat. It has 210 data samples and each sample is described by 7 attributes (features).

Implement the k-means clustering algorithm on this dataset. Use the simple Euclidean distance to compute the distance between any two samples. Start with a random initialization of the centroids and iterate until convergence. The algorithm is assumed to have converged if the number of iterations exceeds 100 **OR** the change in the sum of squared errors (SSE) between two successive iterations is less than 0.001. Run the algorithm for k = 3, 5 and 7. For each value of k, run the algorithm with 10 random initializations of the centroids. Report the average SSE value (averaged over the 10 initializations) for each value of k.

Note: You are **NOT** allowed to use any built-in functions for k-means in your implementation. You are allowed to use built-in functions to compute the Euclidean distance.

Problem 3 (10 points) – Optional for undergraduate students

The *credit card* dataset contains samples from two classes (fraudulent and legitimate). There is a significant imbalance between the two classes. The dataset contains more than 280,000 samples; you are free to work with a subset of the dataset. Use 70% of the subset for training and 30% for testing, at random.

Train a machine learning model on the training set and test on the test set. Use the F-score as the evaluation metric on the test set. Describe in details on how you addressed the class imbalance in the data (you may want to read up relevant research papers to solve this problem). You are also welcome to try out multiple techniques to address class imbalance and perform a comparative analysis.

Note: This problem is open-ended. The goal is not to achieve the highest possible F-score. Rather, the idea is to get you into the habit of reading relevant research papers to address a real-world problem.

Please submit the following through Canvas:

- The Matlab / Python code files
- A ReadMe file with clear instructions on how to run your code for each problem
- A brief report (about 2-3 pages) summarizing your findings. Include the accuracy values for Problems 1, SSE values for Problem 2 and the F-score for Problem 3. Also, describe in details what algorithms you used to address class imbalance. Mention your conclusions.
- Submit all the documents as a single zip file through Canvas