C S 487/519 Applied Machine Learning I

Ensemble approaches

1 Objective

In this *individual* homework, you are required to understand and utilize ensemble approaches.

2 Requirements

1. (20 points) AdaBoost algorithm. Given a dataset shown in the first three columns of the table below. Assume that it is at the boosting round i, the weights used in this round are shown in column 4 and the predicted class labels are shown in column 5. Please show the detailed steps to calculate the updated weight that will be used in the boosting round (i+1). You can calculate these values manually using calculators, or by writing a simple program. In the answer of this question, you need to show the steps 2(c), 2(d), 2(e), and 2(f) clearly (see the algorithm on slide 23 of lec10_ch07_ensemble). Put the detailed calculation and final results to **report.pdf** file.

Index	x	y	weights	$\hat{y}$	Updated weights
1	1.0	1	0.072	-1	
2	2.0	1	0.072	1	
3	3.0	1	0.072	1	
4	4.0	-1	0.072	-1	
5	5.0	-1	0.072	-1	
6	6.0	-1	0.072	-1	
7	7.0	1	0.167	1	
8	8.0	1	0.167	-1	
9	9.0	1	0.167	1	
10	10.0	-1	0.072	-1	

2. (35 points) Write classification code by utilizing several ensemble learning approaches: (1) Random forest, (2) Bagging, and (3) AdaBoost. For Bagging and AdaBoost approaches, you may use decision trees or support vector machines as a base classifier.
3. (20 points) Each classifier needs to be tested using two datasets: (1) the `digits` dataset offered by scikit-learn library, and (ii) the Chronic Kidney Disease Data Set (https://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Disease) in the UCI repository. If there are missing values in the dataset, you may want to properly process them.
4. (20 points) Properly analyze the performance of your ensemble approach. You may want to compare the ensemble approaches with the base classifiers (by reusing results from your previous project). You may also want to test which parameter(s) affect the performance more. Put the analysis in a **report.pdf** file.
5. (5 points) Write a readme file **readme.txt** with the commands to run your code. Your code needs to run in command line, accepting as input parameters the classifier name, the dataset filename, and any required parameter. For example, "`python main.py bagging dataset.csv -n_estimators 10`"
6. Your Python code should be written for Python version 3.5.2 or higher.
7. Please properly organize your Python code. Each required task had better be implemented in a separate python file and imported into the main script. For example, to use the bagging approach, you can create the script **mybagging.py**, then use "`import mybagging`" in the main.py file to test your implementation.

3 Submission instructions

Put all your files (Python code, readme file, report, etc.) to a zip file named **hw.zip** and upload it to Canvas.

4 Grading criteria

(1) The score allocation has already been put beside the questions.

(2) Please make sure that you test your code **thoroughly** by considering all possible test cases. For this project, your code will NOT be tested using more datasets. Thus, it does not need to be flexible to accept other datasets as input. However, you may not hardcode the datasets in your Python code.

(3) At least 5 points will be deducted if submitted files (including files types, file names, etc.) do not follow the instructions.