In the Name of God
Machine Learning (Spring 2020)
Assignment #3
AdaBoost.NC Algorithm

Due date: 20th of Khordad

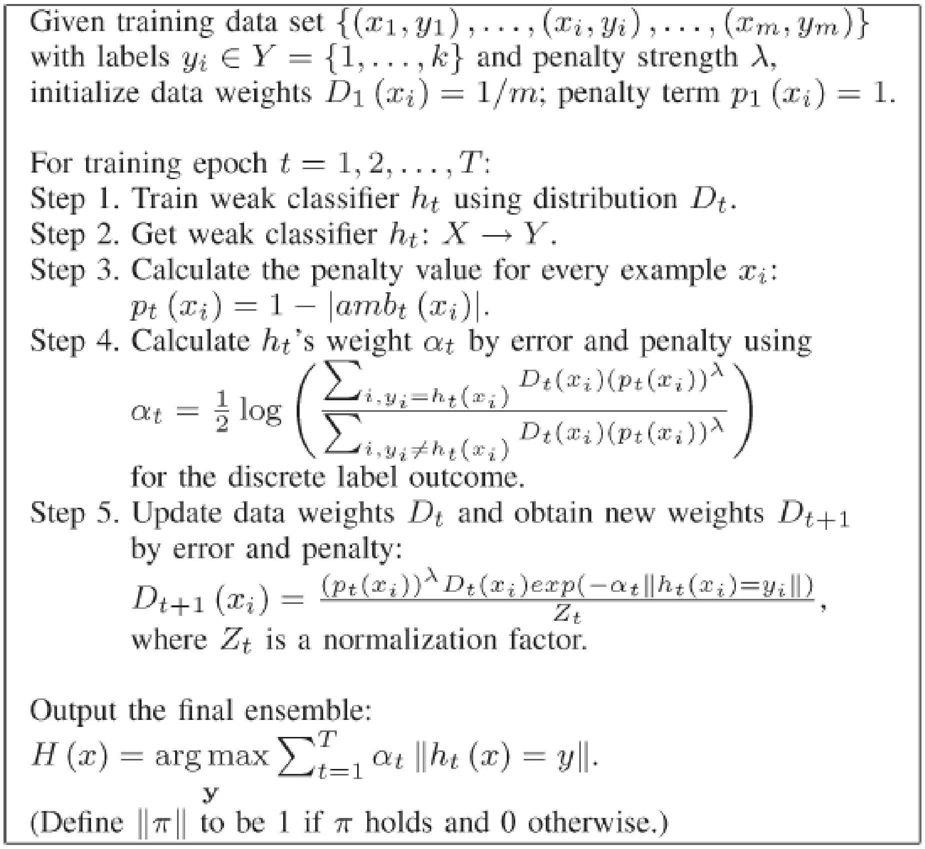
In this assignment, you are about to implement the AdaBoost.NC algorithm which is an ensemble learning algorithm suitable for imbalanced data classification. AdaBoost.NC penalizes classification errors and encourages ensemble diversity sequentially with the AdaBoost training framework. In step 3 of the algorithm, a penalty term p_t is calculated for each training example, in which amb_t assesses the disagreement degree of the classification within the ensemble at the current iteration t. It is defined as:

$$amb_t = \frac{1}{t}\sum_{i=1}^{t}(\|H_t = y\| - \|h_i = y\|)$$

where H_t is the class label given by the ensemble composed of the existing t classifiers. The magnitude of amb_t indicates a “pure” disagreement. p_t is introduced into the weight-updating step (step 5). By doing so, training examples with small $|amb_t|$ will gain more attention. The expression of α_t in step 4 bounds the overall training error. The predefined parameter λ controls the strength of applying p_t. The optimal λ depends on problem domains and base learners. In general, $(0, 4]$ is deemed to be a conservative range of setting λ. As λ becomes larger, there could be either a further performance improvement or a performance degradation.

The pseudo code for AdaBoost.NC algorithm is provided in the figure below.

AdaBoost.NC ALGORITHM [1]



```
Given training data set $\{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_m, y_m)\}$
with labels $y_i \in Y = \{1, \ldots, k\}$ and penalty strength $\lambda$,
initialize data weights $D_1(x_i) = 1/m$; penalty term $p_1(x_i) = 1$.

For training epoch $t = 1, 2, \ldots, T$:
Step 1. Train weak classifier $h_t$ using distribution $D_t$.
Step 2. Get weak classifier $h_t$: $X \rightarrow Y$.
Step 3. Calculate the penalty value for every example $x_i$:
        $p_t(x_i) = 1 - |amb_t(x_i)|$.
Step 4. Calculate $h_t$'s weight $\alpha_t$ by error and penalty using
        $\alpha_t = \frac{1}{2} \log \left( \frac{\sum_{i, y_i = h_t(x_i)} D_t(x_i)(p_t(x_i))^\lambda}{\sum_{i, y_i \neq h_t(x_i)} D_t(x_i)(p_t(x_i))^\lambda} \right)$
        for the discrete label outcome.
Step 5. Update data weights $D_t$ and obtain new weights $D_{t+1}$
        by error and penalty:
        $D_{t+1}(x_i) = \frac{(p_t(x_i))^\lambda D_t(x_i) exp(-\alpha_t \|h_t(x_i) = y_i\|)}{Z_t}$,
        where $Z_t$ is a normalization factor.

Output the final ensemble:
$H(x) = \arg\max_{\mathbf{y}} \sum_{t=1}^{T} \alpha_t \|h_t(x) = y\|$.
(Define $\|\pi\|$ to be 1 if $\pi$ holds and 0 otherwise.)
```

The penalty strength λ in AdaBoost.NC should be tuned for the given data set empirically to achieve the best results. For example, $\lambda = 2$ is a relatively conservative setting to show if AdaBoost.NC can make a performance improvement, and $\lambda = 9$ encourages ensemble diversity aggressively. Use C4.5 decision tree with default parameters as the base learner. The iteration number T should be selected from the set $T \in \{11, 21, 31, 41, 51\}$. Report the results for each value of T.

Split the data set into train and test parts. Use 70% of the data for the training phase and the remaining 30% for the testing phase. Run your codes for 10 individual runs and report the mean and standard deviation of 10 runs for each performance metric.

Since the data set is imbalanced, you should use Precision, Recall, F-measure, AUC, and G-mean measures to evaluate the performance of your implemented algorithm. It is worth mentioning that Precision, Recall, F-measure measures are one-class measures; in other words, you should compute these metrics just for the minority class. For more information regarding the AdaBoost.NC algorithm, please see the attached paper.

Repeat all your experiments for AdaBoost.NC, and for AdaBoost, and Bagging of HW#2 with the noisy version of the data set and report the results. The explanation regarding adding noise to the discrete data set is provided in the attached file. Read it carefully and implement the code to add noise to the data set.

Questions:

- Why is AdaBoost.NC algorithm suitable for imbalanced data classification?
- Diversity is important in ensemble learning. How does AdaBoost.NC incorporate diversity in its model?
- What are the differences between Precision, Recall, F-measure, AUC, and G-mean measures? Please read more about the definitions of these measures.
- How does noise influence the classification results? Are AdaBoost.NC, AdaBoost, and Bagging robust to the added noise?

Important Notes:

- Pay extra attention to the due date. It is fixed and **will not be extended**.
- Be advised that no submissions after the deadline would **be graded**.
- Be sure to comment your code. Also, include instructions on how to run your code (if necessary).
- Prepare a complete report for your assignment and answer all the questions.
- The name of the uploading file should be your **Lastname_Firstname.**

Good Luck