
Problem assignment 10

Due: Thursday, April 15, 2021

Problem 1. Bagging and Boosting

In this problem you will have an opportunity to experiment with the bagging and boosting approaches that let you combine multiple classification models into an ensemble with the hope that the ensemble will improve the classification performance.

You have received a bagging and boosting code that implements both schemes and lets you combine them with multiple classification models. The bagging and boosting function code is implemented in files: $Bag\_classifier.m$ and $Boost\_classifier.m$. The inputs and outputs of these functions are the same. For example, the definition for $Bag\_classifier.m$ is:

```
function [test_y, E] = Bag_classifier(tr_x, tr_y, test_x, params)
% Inputs:
%  tr_x -  Train patterns: x
% tr_y -  Train targets: y
%       test_x  -  Test  patterns: x
%       params:
%           classifier
%     NumberOfBaseModels (iterations),
%           Learner's parameters]
%
% Outputs:
% test_y - Predicted target: y
%               - Errors through the iterations
```

The $Example\_SVM.m$ file shows how to use the bagging procedure with the SVM model implemented in the $SVM\_base.m$ file. The number of SVM models built in this example is 5.

Part a. Use the dataset in $hw10\_train.txt$ $hw10\_test.txt$ files (last columns are class labels) and the code provided to test and compare the performance of the base SVM model, the bagged SVM model and the boosted SVM model. In this experiment, please vary the

number of models bagging and boosting builds from $T = 2, \ldots 10$. For each T, repeat the learning 20 times and average the train and test errors (for each T). Plot the results of the base SVM (T=1) with the bagged SVM and the boosted SVM for ($T = 2, \ldots 10$) in a graph. Analyze and discuss the results.

Part b. Following the $SVM\_base.m$ code definition, write $DT\_base\_full.m$ function that implements the decision tree algorithm without pruning. Submit the code. Run and compare the $DT\_base\_full.m$ alone, and in combination with the bagging and boosting procedures for $T = 2 \ldots, 10$, similarly to the experiment in Part a. Plot the graphs showing the train and test errors performances for different T. Analyze and discuss the results.

Part c. Following the $SVM\_base.m$ code definition, write (and submit) $DT\_base\_simple.m$ function that uses a decision tree with just one decision node (with one splitting test) that lets you divide examples into two subpopulations. Run the same set of experiments as in Parts a and b comparing the base level model to its bagged and boosted version. Analyze and discuss the results.