CS 714 Winter 2012

Homework 3 Boosting

Adaboost is particularly easy to implement. We have, however, provided you with almost all the necessary MATLAB code.

1. Please complete our implementation of Adaboost by substituting in how the votes are set for each new decision stump. You only need to modify boost_ada.m.

2. cancer.mat contains training and test data for a simple leukemia classification problem. The two dimensional inputs correspond to normalized gene expression levels for two different genes measured across different tissues. Use call boosting to train the boosting classifier and plot the number of test errors as a function of the number of boosting iterations up to 5000. Modify the routine to also plot the minimum and average voting margins, minimum or average taken over the training examples, as a function of the boosting iterations. Note that eval_boost.m can also return the total number of votes. What can you say about the plots?

3. Since the data set is two dimensional, it is easy to visualise the decision regions of the resulting boosting classifier. Use plot decision.m to plot the decision regions of the trained AdaBoost classifier after 50, 500, and 5000 decision stumps.

4. Suppose we have collected a finite number of component classifiers from successive boosting iterations, and use the binary ±1 outputs of these component classifiers as features. Thus, with m component classifiers, we have a new m-dimensional feature space. We define a linear classifier without a bias term as follows:

$$f(x; w) = w_1 \times h(x; \theta_1) + \cdots + w_m \times h(x; \theta_m)$$

where $\theta_1, \ldots, \theta_m$ are fixed by boosting. We train this linear classifier by also minimizing the average exponential loss on the training examples. Is the resulting linear classifier equivalent to the one returned by AdaBoost? Justify your answer.