Recitation 9. November 19, 2004

6.034 - Artificial Intelligence

The Boosting Algorithm AdaBoost

AdaBoost Algorithm.

Given training examples $(x_1, y_1), \ldots, (x_m, y_m)$ such that $x_i \in X, y_i \in Y = \{-1, +1\}$.
Initialize $D_1(i) = 1/m$. ($D_t(i)$ represents how much weight is given to example i on iteration t.)
For $t = 1, \ldots, T$:

1. Train weak learner using distribution D_t
2. Get weak classifier $h_t : X \rightarrow Y$ (h_t can be an ID tree, a NN-based classifier, ...)
3. Compute the error of the classifier h_t:

$$\epsilon_t = \Pr_{i \sim D_t} [h_t(x_i) \neq y_i]$$

   and use the error to compute $\alpha_t \in R$:

$$\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)$$

   (α_t represents the weight on each classifier.)

4. Update

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

   where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution)

Output the final classifier to be a weighted majority vote of the T base classifiers:

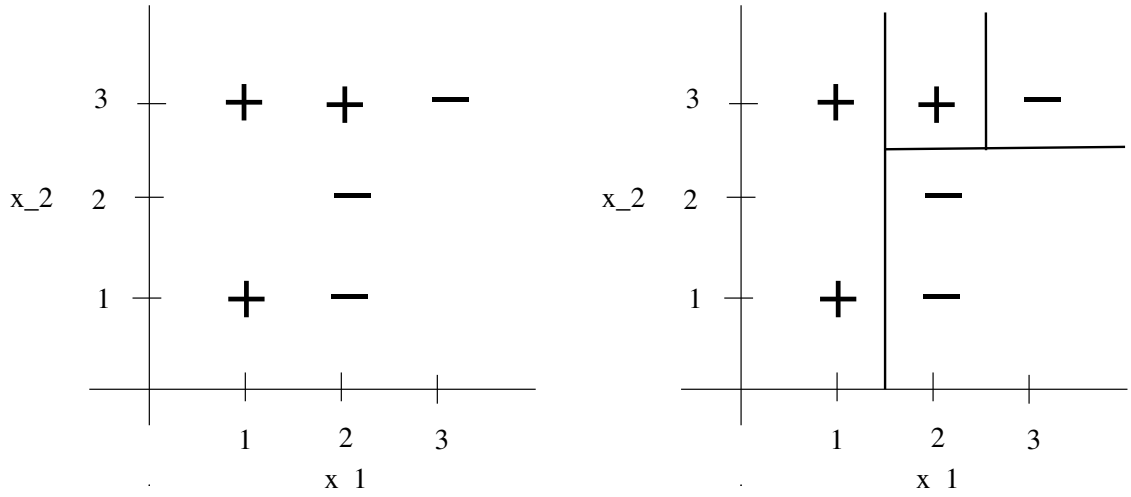
$$H(x) = \text{sign} \left(\sum_{t=1}^{T} \alpha_t h_t(x) \right)$$

Slogan: "T heads are better than 1."
Important properties of Adaboost:

- Integrates disparate classifiers together
- Theoretical bounds – adding a new classifier can't hurt
- Easy to program, doesn't get stuck in local minima on its own
- Sensitive to outliers, prone to overfitting

Problem 1.

Suppose we had the following dataset on the left-hand side below:



The ID tree algorithm classifier line is shown on the right-hand side above. Let's run 3 steps on AdaBoost on the same dataset, using axis-parallel weak classifiers, or in other words, single-test ID trees as the weak classifiers (also known as decision stumps).

There are only eight possible base classifiers based on the four possible decision boundaries. The four possible tests are:

1. $x_1 \geq 1.5$
2. $x_1 \geq 2.5$
3. $x_2 \geq 1.5$
4. $x_2 \geq 2.5$

Here is the table to build:

$i : (x_1, x_2); y_i$	$D_1(i)$	$D_2(i)$	$D_3(i)$
$1 : (1, 1); +1$			
$2 : (1, 3); +1$			
$3 : (2, 3); +1$			
$4 : (2, 1); -1$			
$5 : (2, 2); -1$			
$6 : (3, 3); -1$			
Weak classifier h_t	$h_1 =$	$h_2 =$	$h_3 =$
Error ϵ_t	$\epsilon_1 =$	$\epsilon_2 =$	$\epsilon_3 =$
Weights α_t	$\alpha_1 =$	$\alpha_2 =$	$\alpha_3 =$

What is the resulting classifier of AdaBoost? In others words, how does it treat new data? Draw AdaBoost's decision boundary, and compare it to the decision boundary of the ID tree.

Problem 2.

Now that you have some experience running through the algorithn, let's explore some questions:

1. How does the weight α_t given to classifier h_t relate to the performance of h_t as a function of the error ϵ_t?
2. How does the error of the classifier ϵ_t affect the new $D_t(i)$ on the samples?
3. How can we adapt the distortion-based ID tree learning algorithm to handle weighted samples?
4. How does AdaBoost end up treating outliers?
5. Why is it not the case that the new classifiers "clash" with the old classifiers on the training data?
6. Draw a picture of the training error, theoretical bound on the true error, and the typical test error curve:
7. Do we expect the error of new weak classifiers to increase or decrease with the number of rounds? Why?