Chapter 4. Classification methods
Part 5

April 4, 2007

1 Boosting Methods

Boosting is one of the most powerful leaning idea introduced in the last ten years; see Freund and Schapire (1997)[1]. It was originally designed for classification problems. But is can be extended to regression as well. The motivation for boosting was a procedure that combines the outputs of many "week" classification methods to produce a powerful "committee". Some other idea has been proposed based on the idea.

For any methods we discussed above, it can be simply denoted by $G(x)$, we class a new sample according to the value of $G(X)$. We call $G(x)$ a classifier.

Justification of Boosting

- In practice, it is easy to find quite correct rules of thumb, however hard to find single highly accurate prediction rule.

[1] Y. Freund and R.E. Schapire (1997) "A decision-theoretic generalization of on-line learning and an application to boosting," Journal of Computer and System Sciences, 55(1):119-139

- If the training examples are few and the hypothesis space is large then there are several equally accurate classifiers.
- Hypothesis space does not contain the true function, but it has several good approximations.
- Exhaustive global search in the hypothesis space is expensive so we can combine the predictions of several locally accurate classifiers.

What is different with the previous methods

- Boosting: Successive classifiers depends upon its predecessors. Previous methods : Individual classifiers were independent.
- Training Examples may have unequal weights.
- Look at errors from previous classifier step to decide how to focus on next iteration over data
- Set weights to focus more on "hard" examples. (the ones on which we committed mistakes in the previous iterations)

Boosting Algorithm Consider two classes $Y \in \{-1, 1\}$ Given a vector of predictor variables X, a classifier $G(X)$ produce a prediction taking values $\{-1, 1\}$. The error rate is defined as

$$err = n^{-1} \sum_{i=1}^{n} I(y_i \neq G(X_i))$$

Suppose we are going to combine M classifiers

$$G(X) = \sum_{m=1}^{M} G_m(X)$$

The following is the most popular "AdaBoost" algorithm

1. Initially assign uniform weights $w_i = 1/n$

2. For $m = 1$ to M

   (a) Fit a classifier $G_m(x)$ to the training data using weights w_i

   (b) Compute err_m the error rate as

   $err_m = [\sum_{i=1}^{n} w_i * I(y_i \neq G_m(x_i))] / \sum_{i=1}^{n} w_i]$

   (c) weight $\alpha_m = \log((1 - err_m)/err_m)$

   (d) Set $w_i \leftarrow w_i * \exp(\alpha_m * I(y_i \neq G_m(x_i))), i = 1, ..., n$

3. output

$$G_{final}(x) = sign[\sum_{m=1}^{M} \alpha_m G_m(x)]$$

Example 1.1 *Classification in genetics For the leukemia gene expression data (***(training points)***. There are 38 cells with 250 genes (selected from about 7000 genes). they are from two types of cells.*

We arbitrarily choose cell 21-33 for test and the others for learning

*Boosting method: All are correctly clasified. (***(code)***)*

The procedure ca

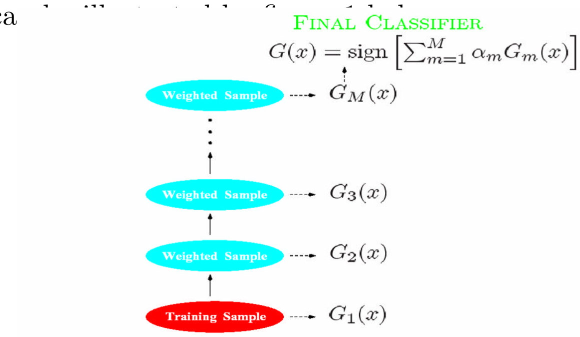


Figure 1: An illustration of the algorithm

As an example, we consider a classification problem as shown in figures 2-5.

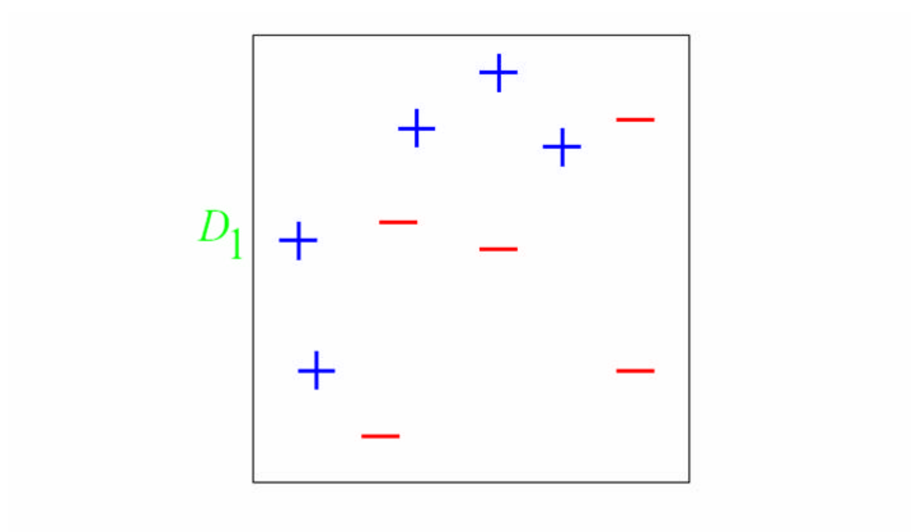


Figure 2: An example for classification

Round 1

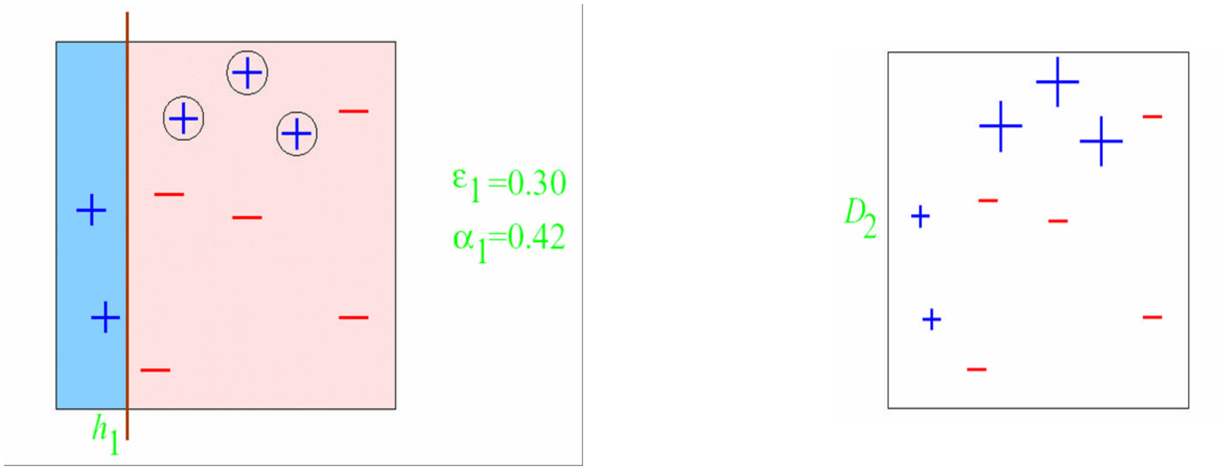


Figure 3: Example: The misclassified cases (circled) are given bigger weight

Round 2

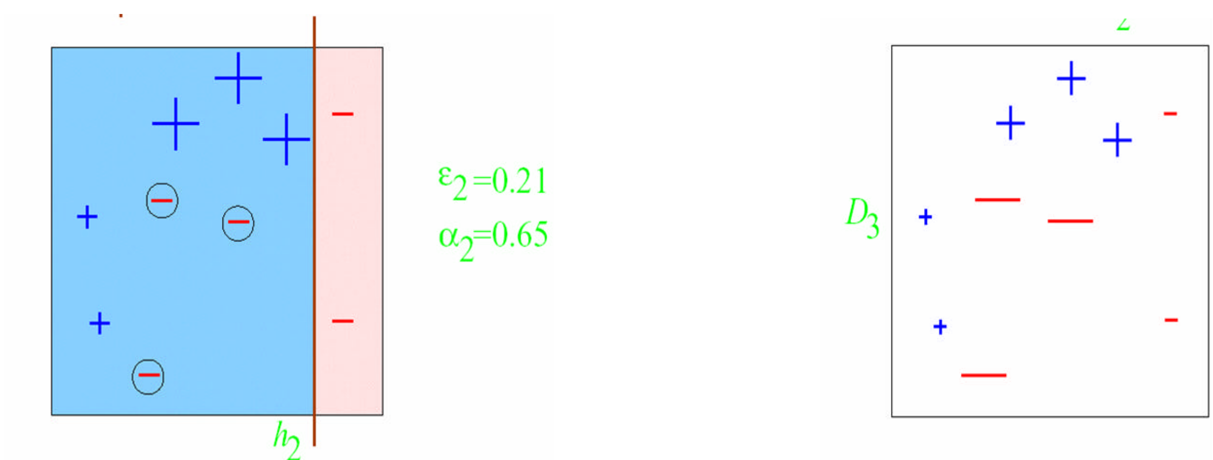
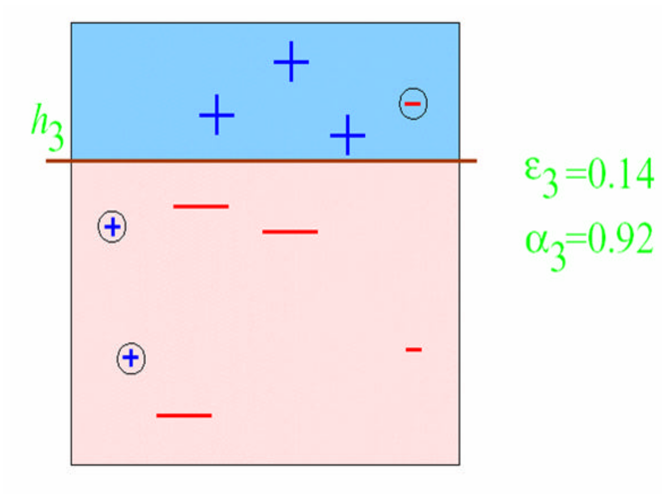


Figure 4: Example: The misclassified cases (circled) are given bigger weight

⇉

Round 3



The final classifier is

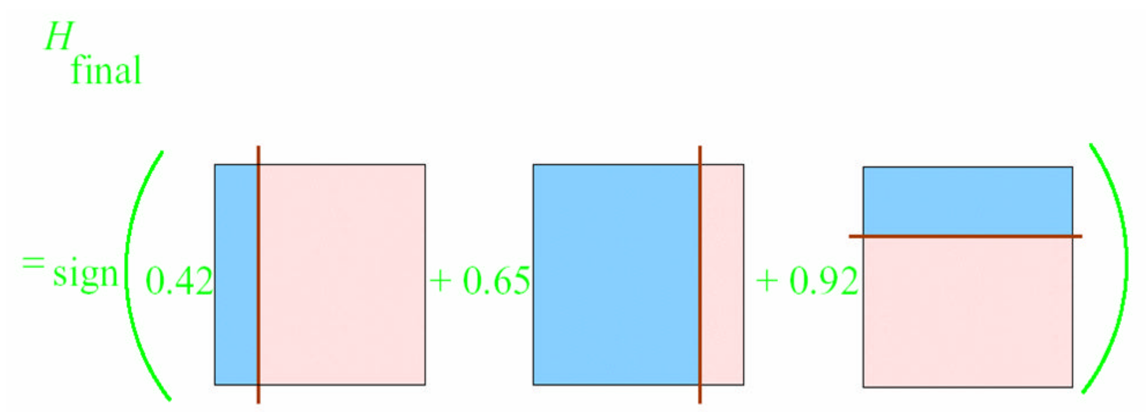


Figure 5: Example: (cont')