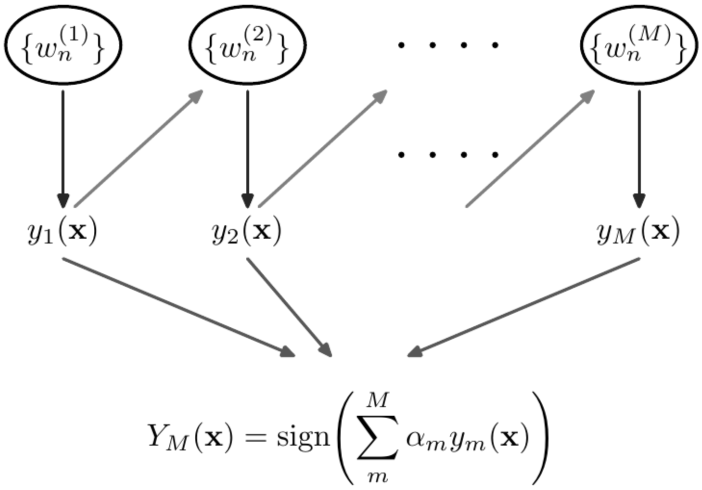
Pattern recognition and machine learning Bishop

AdaBoost

AdaBoost, short for 'adaptive boosting', develoAdaBoostped by Freund and

Schapire (1996).



Schematic illustration of the boosting framework. Each base classifier $y_m(x)$ is trained on a weighted form of the training set (blue arrows) in which the weights $w_n(m)$ depend on the performance of the previous base classifier $y_{m-1}(x)$(green arrows). Once all base classifiers have been trained, they are combined to give the final classifier $Y_M(x)$ (red arrows).

AdaBoost

1. Initialize the data weighting coefficients $\{w_n\}$ by setting $w_n^{(1)} = \frac{1}{N}, n = 1, \cdots, N$.

2. For $m = 1, \cdots, M$:

(a) Fit a classifier $y_m(x)$ to the training data by minimizing the weighted error function

$$J_m = \sum_{n=1}^N w_n^{(m)} I(y_m(x_n) \neq t_n) \tag{1}$$

where $I(y_m(\text{xn}) \neq t_n)$ is the indicator function and equals 1 when $y_m(x_n) = t_n$ and 0 otherwise.

(b) Evaluate the quantities

$$\varepsilon_m = \frac{w_n^{(m)} I(y_m(x_n) \neq t_n)}{\sum_{n=1}^N w_n^{(m)}} \tag{2}$$

and then use these to evaluate

$$\alpha_m = \ln\left\{\frac{1 - \varepsilon_m}{\varepsilon_m}\right\} \tag{3}$$

(c) Update the data weighting coefficients

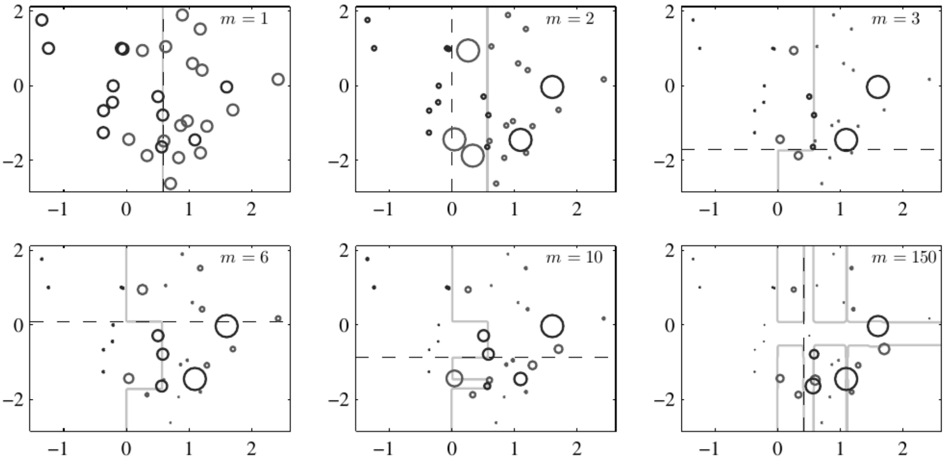
$$w_n^{(m+1)} = w_n^{(m)} \exp\{\alpha_m I(y_m(x_n) \neq t_n)\} \tag{4}$$

3. Make predictions using the final model, which is given by

$$Y_M(x) = \operatorname{sign}\left(\sum_{m=1}^{M} \alpha_m y_m(x)\right) \tag{5}$$

Illustration of boosting in which the base learners consist of simple thresholds applied to one or

other of the axes. Each figure shows the number m of base learners trained so far, along with the decision boundary of the most recent base learner (dashed black line) and the combined decision boundary of the ensemble (solid green line). Each data point is depicted by a circle whose radius indicates the weight assigned to that data point when training the most recently added base learner. Thus, for instance, we see that points that are misclassified by the m = 1 base learner are given greater weight when training the m = 2 base learner.



minimizing exponential error

Friedman et al. (2000) gave a different and very simple interpretation of boosting in terms of the sequential minimization of an exponential error function.

Consider the exponential error function defined by

$$E = \sum_{n=1}^{N} \exp\{-t_n f_m(x_n)\} \tag{6}$$

where $f_m(x)$ is a classifier defined in terms of a linear combination of base classifiers $y_l(x)$ of the form

$$f_m = \frac{1}{2}\sum_{l=1}^{m} \alpha_l y_l(x) \tag{7}$$

and $t_n \in \{-1, 1\}$ are the training set target values. Our goal is to minimize E with respect to both the weighting coefficients α_l and the parameters of the base classifiers $y_l(x)$.

Instead of doing a global error function minimization, however, we shall suppose that the base classifiers y_1 (x), $\cdots$, y_{m-1} (x) are fixed, as are their coefficients α_1, $\cdots$, α_{m-1} , and so we are minimizing only with respect to αm and $y_m(x)$. Separating off the contribution from base classifier $y_m(x)$, we can then write the error function in the form

$$\begin{aligned} E &= \sum_{n=1}^{N} \exp\left\{-t_n f_{m-1}(x_n) - \frac{1}{2} t_n \alpha_m y_m(x_n)\right\} \\ &= \sum_{n=1}^{N} w_n^{(m)} \exp\left\{-\frac{1}{2} t_n \alpha_m y_m(x_n)\right\} \end{aligned} \tag{8}$$

where the coefficients $w_n = \exp\{-t_n f_{m-1}(x_n)\}$ can be viewed as constants because we are optimizing only α_m and $y_m(x)$. If we denote by T_m the set of data points that are correctly classified by $y_m(x)$, and if we denote the remaining misclassified points by M_m , then we can in turn rewrite the error function in the form

$$\begin{aligned} E &= e^{-\alpha_m/2} \sum_{n \in T_m} w_n^{(m)} + e^{\alpha/2} \sum_{n \in M_m} w_n^{(m)} \\ &= (e^{\alpha_m/2} - e^{-\alpha_m/2}) \sum_{n=1}^{N} w_n^{(m)} I(y_m(x_n) \neq t_n) + e^{-\alpha_m/2} \sum_{n=1}^{N} w_n^{(m)} \end{aligned}$$

When we minimize this with respect to ym (x), we see that the second term is constant, and so this is equivalent to minimizing (1) because the overall multiplicative factor in front of the summation does not affect the location of the minimum.

Similarly, minimizing with respect to Îśm , we obtain (3) in which ε_m is defined by (2).

From (8) we see that, having found Îśm and $y_m(x)$, the weights on the data points are updated using

$$w_n^{(m+1)} = w_n^{(m)} \exp\left\{ -\frac{1}{2} t_n \alpha_m y_m(x_n) \right\}$$

Making use of the fact that

$$t_n y_m(x_n) = 1 - 2I(y_m(x_n) \neq t_n)$$

we see that the weights wn are updated at the next iteration using

$$w_n^{(m+1)} = w_n^{(m)} \exp(-\alpha_m/2) \exp\{\alpha_m I(y_m(x_n) \neq t_n)\}.$$

Because the term $\exp(-\alpha m/2)$ is independent of n, we see that it weights all data points by the same factor and so can be discarded. Thus we obtain (4).

Finally, once all the base classifiers are trained, new data points are classified by evaluating the sign of the combined function defined according to (7). Because the factor of 1/2 does not affect the sign it can be omitted, giving (5).