Gentle Boosting

The task of boosting is to take a given set of classifiers and assemble them in a way that the resulting classifier is much better than each individual base classifier (weak learner). Usually, it is much more easy to find a set of weak classifiers than finding a global optimal classifier. So it makes sense to find an algorithm assembling a set of weak learner to get better performance.
There are several variants of boosting algorithms in the literature, so there is no unique boosting allgorithm and their performance and requirements on the weak learner can both significantly differ. However, on a high-level all boosting methods follow basically the same type of pattern.

1. For eac training point (Xi, Yi) one has a weight γ_i .
2. A step of boosting method involves the following steps
   (a) One trains a classifier f_k using a base method (weak learning) with the weighted training data (X_i, Y_i, γ_i),
   (b) One re-computes the weights γ, where usually the weights of wrongly classified training points are increasing and the weights of correctly classified points are decreasing.
3. One aggregates the classifiers f_k to the final classifier *F(x)=*$sign(\sum_{k=1} \alpha_k f_k)$, where the coeffients α_k are either one or depend on the error of the error of the classifier f_k.

Proposition 1: Suppose the weak learner f_t is real-valued, $f_t: \chi \to \mathbb{R}$. The update step $F_{t+1} = F_t + c_t f_t$ of the GentleBoost algorithm is an approximate Newton step in order to minimize the empirical exponential loss.
Proof: we can expand the risk of F+f up to second order,

$$R(F+f) = \mathbb{E}[e^{-Y(F(X)+f(X))}] \approx \mathbb{E}[e^{-YF(X)}(1 - Yf(X) + \frac{1}{2}f(x)^2)]$$

We want to find a classifier $f(X)$ to minimize the risk. Note that the first term does not depend on *f* so we can modify it without changing the minimizer and get,

$$\mathbb{E}\left[e^{-YF(X)}\left(\frac{1}{2} - Yf(X) + \frac{1}{2}f(x)^2\right)\right] = \frac{1}{2}\mathbb{E}\left[e^{-YF(X)}(Y^2 - Yf(X) + f(x)^2)\right] = \frac{1}{2}\mathbb{E}\left[e^{-YF(X)}\big(Y - f(x)\big)^2\right],$$

which is up to the normalization of the weights $\gamma_i = e^{-Y_i F(X_i)}$ equal to the weighted squared loss minimizerd by the weak learner, Thus the weak learner minmizes in each step a second order approximation of the exponentianl loss which is equivalent to an approximate Newton step.

In principle, one can take every learning method which can fit the weighted zero-one loss as the weak learner. In practice, one uses for performance reasons (in particular at test time) just simple decision stump of the form: $f_t(x) = a\mathbb{1}_{<w,x>+b>0} + c$.
Then we need to optimize the parameters of each weak learning to minimize the weighted square loss.

$$L(f) = \sum_i^N \gamma_i (Y_i - f(X_i))^2$$

The base learner we use here is $f_t(x) = a\mathbb{I}_{<w,x>+b>0} + c$, the weight w is fixed. We need to compute parameters a,b,c as minimizer of the weighted least squared loss $L(f)$.

$$\frac{\partial L(f)}{\partial a} = \frac{\partial \sum_i^N \gamma_i (Y_i - a\mathbb{I}_{<w_i,x>+b>0} - c)^2}{\partial a}$$
$$= 2\sum_i^N \gamma_i \left(Y_i - a\mathbb{I}_{<w_i,x>+b>0} - c\right)\left(-\mathbb{I}_{<w_i,x>+b>0}\right) = 0$$

$$\frac{\partial L(f)}{\partial c} = \frac{\partial \sum_i^N \gamma_i (Y_i - a\mathbb{I}_{<w_i,x>+b>0} - c)^2}{\partial c} = -2\sum_i^N \gamma_i \left(Y_i - a\mathbb{I}_{<w_i,x>+b>0} - c\right) = 0$$

$$\sum_{i=1}^N \gamma_i Y_i \mathbb{I}_{<w_i,x>+b>0} = a\sum_{i=1}^N \gamma_i \mathbb{I}_{<w_i,x>+b>0} + c\sum_{i=1}^N \gamma_i \mathbb{I}_{<w_i,x>+b>0} \quad (1)$$

$$\sum_{i=1}^N \gamma_i Y_i = a\sum_{i=1}^N \gamma_i \mathbb{I}_{<w_i,x>+b>0} + c\sum_{i=1}^N \gamma_i \quad (2)$$

$$a = \frac{\sum_{i=1}^N \gamma_i Y_i \mathbb{I}_{<w_i,x>+b>0}}{\sum_{i=1}^N \gamma_i \mathbb{I}_{<w_i,x>+b>0}} - c \quad (3)$$

$$c = \frac{\sum_{i=1}^N \gamma_i Y_i (1-\mathbb{I}_{<w_i,x>+b>0})}{\sum_{i=1}^N \gamma_i (1-\mathbb{I}_{<w_i,x>+b>0})} \quad (4)$$

$$a = \frac{\sum_{i=1}^N \gamma_i Y_i \mathbb{I}_{<w_i,x>+b>0}}{\sum_{i=1}^N \gamma_i \mathbb{I}_{<w_i,x>+b>0}} - \frac{\sum_{i=1}^N \gamma_i Y_i (1-\mathbb{I}_{<w_i,x>+b>0})}{\sum_{i=1}^N \gamma_i (1-\mathbb{I}_{<w_i,x>+b>0})} \quad (5)$$

Note that we have now derived formulas for the optimal a and b, however they still depend on the choice of b. The key trick to compute the best decision stump is that for n training points there are only n+1 possible thresholds b. Thus we compute for each possible threshold the optimal parameters a; c for this threshold and compute the corresponding weighted least squares error. Then the parameters a; b; c are selected which yield the minimal weighted least squares error.