Machine Learning: Principles and Techniques

Boosting, Gradient Boosting
IE 506

April 5, 2024

1. Classification Algorithms
   - Boosting

Classification Algorithms: Boosting

Boosting: AdaBoost

Algorithm AdaBoost

Input: sequence of N labeled examples $\langle (x_1, y_1), \ldots, (x_N, y_N) \rangle$
distribution D over the N examples
weak learning algorithm **WeakLearn**
integer T specifying number of iterations

Initialize the weight vector: $w_i^1 = D(i)$ for $i = 1, \ldots, N$.

Do for $t = 1, 2, \ldots, T$

1. Set
$$\mathbf{p}^t = \frac{\mathbf{w}^t}{\sum_{i=1}^N w_i^t}$$
2. Call **WeakLearn**, providing it with the distribution $\mathbf{p}^t$; get back a hypothesis $h_t : X \to [0, 1]$.
3. Calculate the error of h_t: $\epsilon_t = \sum_{i=1}^N p_i^t |h_t(x_i) - y_i|$.
4. Set $\beta_t = \epsilon_t / (1 - \epsilon_t)$.
5. Set the new weights vector to be
$$w_i^{t+1} = w_i^t \beta_t^{1 - |h_t(x_i) - y_i|}$$

Output the hypothesis

$$h_f(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^T \left(\log \frac{1}{\beta_t}\right) h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \log \frac{1}{\beta_t} \\ 0 & \text{otherwise} \end{cases} .$$

Y. Freund, R. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting, Journal of Computer and System Sciences. Vol 55-1, pp. 119-139, 1997.

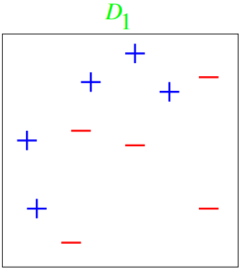
AdaBoost - a loss perspective†

- Input: N samples $\{(x^i, y^i)\}_{i=1}^N$, $x^i \in \mathbb{R}^d$, $y^i \in \{+1, -1\}, \forall i \in \{1, 2, \ldots, N\}$.
- Initialize weights $w_i^1 = 1/N, \forall i \in \{1, 2, \ldots, N\}$.
- For $t = 1, 2, \ldots, T$ do:
  - Train a weak classifier $h_t : \mathbb{R}^d \to \{+1, -1\}$ with examples weighed using current weights w_i^t by minimizing: $\epsilon_t = \sum_{i=1}^N w_i^t \mathbb{I}(h_t(x^i) \neq y^i)$.
  - Compute $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$
  - Update weights as: $w_i^{t+1} = w_i^t e^{-\alpha_t y^i h_t(x^i)}$
  - Normalize $w_i^{t+1} = w_i^{t+1} / \sum_{i=1}^N w_i^{t+1}$.
- Output: Final classifier $h(x) = \mathrm{sign}(\sum_{t=1}^T \alpha_t h_t(x))$.

†:J. Friedman, T. Hastie and R. Tibshirani. Additive logistic regression: A statistical view of Boosting, Annals of Statistics, 2000, Vol. 28, no. 2, pp. 337–407.

Bagging: AdaBoost

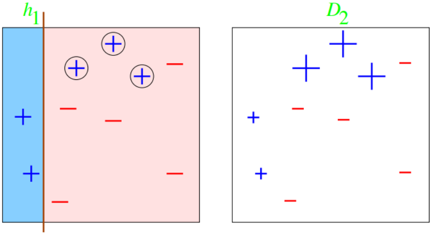
10 data points and 2 features



Example from Ameet Talwalkar's slides on AdaBoost

Bagging: AdaBoost

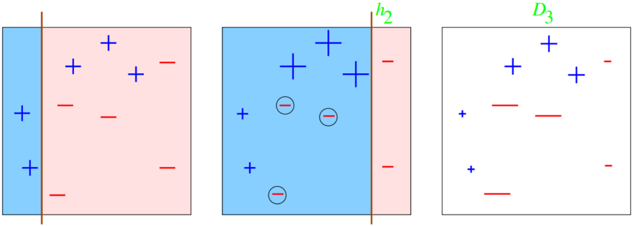
Round 1: $t = 1$



- 3 misclassified data points (denoted by circles): $\epsilon_1 = 0.3, \alpha_1 = 0.42$
- Weights are recomputed, and the 3 misclassified data points receive larger weights.

Bagging: AdaBoost

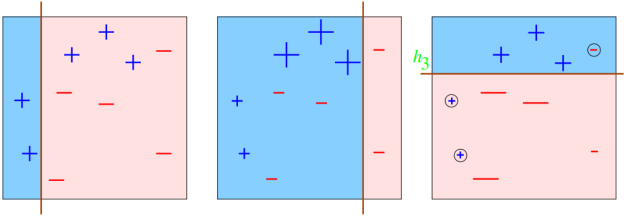
Round 2: $t = 2$



- **Note:** The new classifier h_2 strives to perform correctly for the data points misclassified in round 1.
- However in that process, there are 3 new misclassified data points in round 2 (denoted by circles): $\epsilon_2 = 0.21, \alpha_2 = 0.65$.
  Note that $\epsilon_2 \neq 0.3$ since the weights $w_2^i < 1/10$ for i-th misclassified example, which was correctly classified in the previous round.
- Weights are recomputed, and the weights of 3 misclassified data points increase.
- Data points which have been correctly predicted in both rounds have small weights.

Bagging: AdaBoost

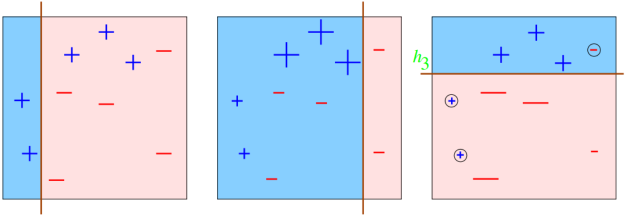
Round 3: $t = 3$



- **Note:** The new classifier h_3 strives to perform correctly for the data points misclassified in round 2.
- However in that process, there are 3 new misclassified data points in round 3 (denoted by circles): $\epsilon_3 = 0.14, \alpha_2 = 0.92$.

Bagging: AdaBoost

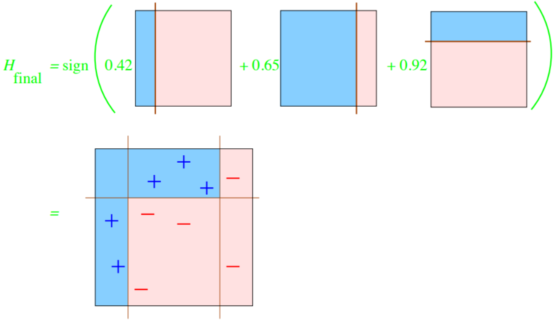
Round 3: $t = 3$



- Even though previously correctly classified points are misclassified in this round, we see that our error rate is low; what's the intuition?
  - Since they have been consistently correctly classified in the past, the current mispredictions will not have a huge impact on the overall prediction.
- Data points which have been correctly predicted in all previous rounds have very small weights.

Bagging: AdaBoost

Final classifier: combining 3 classifiers



- All data points are now classified correctly!

Gradient Boosting

Gradient Boosting

- Given: N samples $\{(x^i, y^i)\}_{i=1}^N$, $x^i \in \mathbb{R}^d$, $y^i \in \{+1, -1\}, \forall i \in \{1, 2, \ldots, N\}$.
- Suppose we have access to a model at round t: F_{t-1}
- Using F_{t-1}, we find the predictions $F_{t-1}(x^1), F_{t-1}(x^2), \ldots, F_{t-1}(x^N)$.
- If F_{t-1} does not make correct predictions on all samples, then we can improve the classifier in a stagewise manner similar to adaboost as: $F_t = F_{t-1} + \alpha_t h_t$.
- The idea is to find α_t, h_t.

Gradient Boosting

- Assume $\alpha_t = 1$ for simplicity.
- Then from $F_t = F_{t-1} + \alpha_t h_t$ and $\alpha_t = 1$, we have:

$$F_t(x^i) = F_{t-1}(x^i) + h_t(x^i), \forall i \in \{1, 2, \dots, N\}.$$

- Since we want $F_t(x^i) = y^i, \forall i$, we have:

$$y^i = F_{t-1}(x^i) + h_t(x^i), \forall i \in \{1, 2, \dots, N\}.$$

- Thus we can write: $h_t(x^i) = y^i - F_{t-1}(x^i), \forall i$.
- To find h_t, we can fit a regression tree on $\{(x^i, r^i_{t-1})\}$ where $r^i_{t-1} = y^i - F_{t-1}(x^i)$ is the residual for sample i, from the predictions made using F_{t-1}.

Gradient Boosting

- Recall: in adaboost, we solved a loss minimization of the form $\min_f \sum_{i=1}^N e^{-y^i f(x^i)}$.
- Suppose consider the loss minimization with squared loss:

$$\ell(F) = \frac{1}{2}\sum_{i=1}^{N}(y^i - F(x^i))^2.$$

- Now, the gradient of ℓ with respect to the predictions $F(x^i)$ can be given as:

$$\frac{\partial \ell}{\partial F(x^i)} = F(x^i) - y^i.$$

- Then from our previous discussion, we see that the residual r^i is simply the negative of the partial derivative $g^i = \frac{\partial \ell}{\partial F(x^i)}$.

Gradient Boosting

- Hence we can write: $\forall i \in \{1, 2, \ldots, N\}$:

$$\begin{aligned} F_t(x^i) &= F_{t-1}(x^i) + h_t(x^i) \\ &= F_{t-1}(x^i) + (y^i - F_{t-1}(x^i)) \\ &= F_{t-1}(x^i) + r^i_{t-1} \\ &= F_{t-1}(x^i) - \eta g^i_{t-1} \end{aligned}$$

  where $\eta = 1$.
- Thus the update to F_t can be written as: $F_t = F_{t-1} - \eta \nabla_F \ell$, where ℓ is the squared loss.
- This idea can be generalized to other loss function ℓ.

Gradient Boosting - a loss perspective†

- Input: N samples $\{(x^i, y^i)\}_{i=1}^N$, $x^i \in \mathbb{R}^d$, $y^i \in \{+1, -1\}, \forall i \in \{1, 2, \ldots, N\}$, loss function ℓ.
- Initialize $F_0 = \sum_{i=1}^N y^i / N$.
- For $t = 1, 2, \ldots, T$ do:
  - Find $g_{t-1}^i = \frac{\partial \ell}{\partial F_{t-1}(x^i)}, \forall i \in \{1, 2, \ldots, N\}$.
  - Fit a regression tree h_t on data $\{(x^i, -g_{t-1}^i)\}_{i=1}^N$.
  - $\alpha_t = 1$ (**Optional**: Find $\alpha_t = \arg\min_\alpha F_{t-1} + \alpha h_t$).
  - $F_t = F_{t-1} + \alpha_t h_t$.
- Output: Final classifier $h(x) = (\sum_{t=1}^T \alpha_t h_t(x))$.

†:Friedman, J. H. Greedy function approximation: a gradient boosting machine. Annals of Statistics, 2001, pages 1189–1232.

Gradient Boosting - a loss perspective

Other loss functions:

- Absolute loss: $\ell_{abs}(y, F(x)) = |y - F(x)|$.
- Huber loss:

$$\ell_{Huber}(y, F(x)) = \begin{cases} \frac{1}{2}(y - F(x))^2 & \text{if} |y - F(x)| \leq \delta \\ \delta(|y - F(x)| - \frac{\delta}{2}) & \text{if} |y - F(x)| > \delta \end{cases}.$$

These loss functions are robust to outliers.