Stochastic Boosting

Algorithm: Gradient Boosted Trees for Classification

Input: training set Z=$\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$,
M – number of iterations

1. $f_0(x) = log \frac{p_1}{1-p_1}$ p_1- part of objects of first class
2. For m=1…M:
3. $g_i = \frac{dL(y_i, f_m(\boldsymbol{x}_i))}{df_m(\boldsymbol{x}_i)}$
4. Fit a decision tree $h_m(\boldsymbol{x}_i)$ to the target g_i

   (auxiliary training set $\{(\boldsymbol{x}_1, g_i), \ldots, (\boldsymbol{x}_n, g_n)\}$)
5. $\rho_m = \underset{\rho}{\text{argmax}}\, Q[f_{m-1}(\boldsymbol{x}) + \rho h_m(\boldsymbol{x})]$
6. $f_m(x) = f_{m-1}(x) + \mathbf{v}\rho_m h_m(\boldsymbol{x}_i)$
7. Return: $f_M(\boldsymbol{x})$

v - regularization (learningRate), recommended ≤ 0.1

Algorithm: Gradient Boosted Trees for Classification

+Stochastic Boosting

Input: training set $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$, M – number of iterations

1. $f_0(x) = \frac{1}{n}\sum_{i=1}^{n} y_i$
2. For m=1…M:
3. $g_i = \frac{dL(y_i, f_m(x_i))}{df_m(x_i)}$
4. Fit a decision tree $h_m(\boldsymbol{x}_i)$ to the target g_i (auxiliary training set $\{(\boldsymbol{x}_1, g_i), \ldots, (\boldsymbol{x}_k, g_k)\}$), ***$k=0.5n$ created by random sampling with replacement***
5. $\rho_m = \underset{\rho}{\arg\max}\, Q[f_{m-1}(\boldsymbol{x}) + \rho h_m(\boldsymbol{x})]$
6. $f_m(x) = f_{m-1}(x) + v\rho_m h_m(x_i)$
7. Return $f_M(\boldsymbol{x})$

v - regularization (learningRate), recommended ≤ 0.1

Sampling with Replacement

n=8

1	2	3	4	5	6	7	8

k=4

7	3	1	3

Stochastic Boosting

- Improved performance
- Improved predictive power