Gradient Boosted Decision Trees

Classification

Classification

Given a training set: $\boldsymbol{Z=\{(x_1, y_1),\ldots,(x_n, y_n)\}}$
$\boldsymbol{x_i}$- features, $\boldsymbol{y_i}$- class labels (0, 1)

Goal is to find $\boldsymbol{f(x)}$ using training set, such as

$$\min \sum_{(x,y)\in T} [f(\boldsymbol{x}) \neq y]$$

at test set $\boldsymbol{T=\{(x_1, y_1),\ldots,(x_n, y_n)\}}$

How to build $\boldsymbol{f(x)}$?

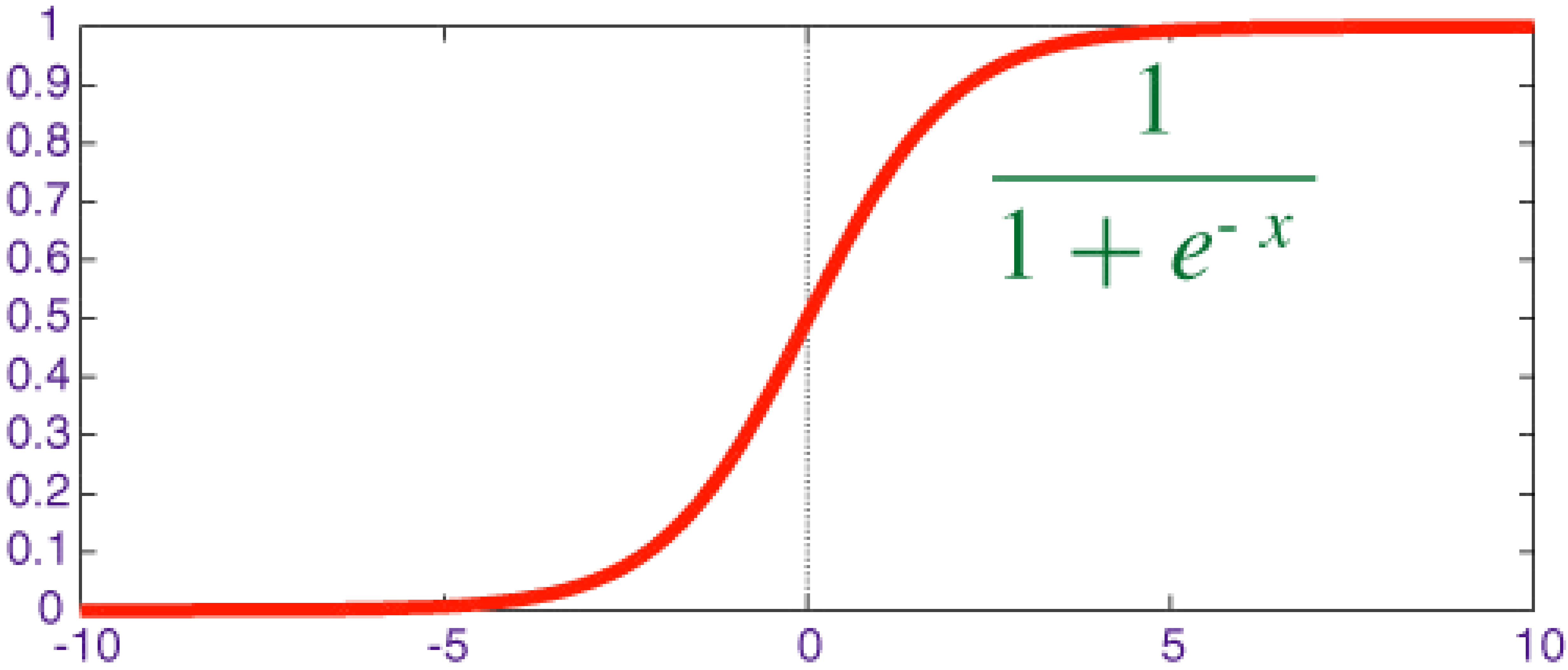
Gradient Boosted Trees for Classification

$$P(y = 1|\boldsymbol{x}) = \frac{1}{1 + \exp(-\sum_{m=1}^{M} h_m(\boldsymbol{x}))}$$

$h_m(\boldsymbol{x})$ - a decision tree

$$0 < P(y = 1|\boldsymbol{x}) < 1$$

Sigmoid function



$$f(\boldsymbol{x}) = \sum_{m=1}^{M} h_m(\boldsymbol{x})$$

$$P(y = 1|\boldsymbol{x}) = \frac{1}{1 + \exp(-f(\boldsymbol{x}))}$$

Likelihood:

$$\prod_{i=1}^{n} P(y_i|\boldsymbol{x}_i) = P(y_1|\boldsymbol{x}_1) \cdot \ldots \cdot P(y_n|\boldsymbol{x}_n)$$

“The principle of maximum likelihood”

Algorithm: find a function f(x) maximizing the **likelihood**

Equivalent: find a function f(x) maximizing the logarithm of the **likelihood**

(since logarithm is a monotone function)

$$Q[f] = \sum_{i=1}^{n} \log(P(y_i|\boldsymbol{x}_i))$$

$$\max Q[f]$$

$$L(y_i, f(\boldsymbol{x}_i)) = \log(P(y_i|\boldsymbol{x}_i))$$

$$Q[f] = \sum_{i=1}^{n} L(y_i, f(\boldsymbol{x}_i))$$

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. $g_i = \frac{dL(y_i, f_m(x_i))}{df_m(x_i)}$
4. Fit a decision tree $h_m(\boldsymbol{x}_i)$ to the target g_i

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

v - regularization (learningRate), recommended ≤ 0.1