Бустинг

X - множество признаков, Y - множество классов. Пусть есть выборка $\{(x_i, y_i)\}_{i=1}^n \in X \times Y$, а также семейство *базовых алгоритмов* H, в котором каждый элемент $h(x; a) \in H$ определяется набором параметров $a \in A$.

Основная идея - построить новый алгоритм классификации в следующей форме:

$$F_M(x) = \sum_{m=1}^{M} h(x; a) b_m, \quad \text{где } a_i \in A, b_i \in \mathbb{R}$$

Построение проводится *жадно*: пусть уже есть классификатор F_{k-1} (от $k-1$ параметра), тогда построение классификатора F_k сводится к подбору наиболее оптимальных параметров a_k, b_k:

$$F_k = F_{k-1} + h(\cdot, a_k) b_k$$

Подбор параметров

Оптимальность параметров определятся с помощью функции ошибки:

$$Q = \sum_{i=1}^{N} L(y_i, F(x_i)) \to min, \text{ где } L(y_i, F(x_i)) \text{ - \textit{функция потерь}}.$$

Минимизируем функционал с помощью метода *градиентного спуска*. Пусть есть алгоритм F_{k-1}. Найдем такой $b_k \in \mathbb{R}$, что значение Q для $F_k = F_{k-1} - b_k \nabla Q$ минимально.

$$\text{Тут } \nabla Q = \left[\frac{\partial Q}{\partial F_{k-1}}(x_i) \right]_{i=1}^{N} = \left[\frac{\partial L(y_i, F_{k-1})}{\partial F_{k-1}}(x_i) \right]_{i=1}^{N}$$

$$\text{То есть } b_k = \underset{b \in \mathbb{R}}{argmin} \left(\sum_{i=1}^{N} L\Big(F_{k-1}(x_i) - b \nabla Q_i \Big) \right)$$

Однако ∇Q не является алгоритмом из H, поэтому найдем наиболее "близкий"алгоритм $h(., a_k)$:

$$a_k = \underset{a \in A}{argmin} \left(\sum_{i=1}^{N} L\Big(\nabla Q_i, h(x_i; a)\Big) \right)$$

Это эквивалентно обучению базового алгоритма на выборке $\{(x_i, \nabla Q_i)\}_{i=1}^{N}$. Теперь осталось найти наиболее оптимальный b_k:

$$b_k = \underset{b \in \mathbb{R}}{argmin} \left(\sum_{i=1}^{N} L\Big(F_{k-1}(x_i) + h(x_i; a_k)b\Big) \right)$$

Таким образом, $F_k = F_{k-1} + b_k h(\cdot; a_k)$

Бустинг над деревьями Рассмотрим случай бинарной классификации, когда в качестве базовых алгоритмов выступают решающие деревья. Каждому листу дерева соответствует значение a_i - "степень принадлежности"одному из классов. Множество X разбивается на J непересекающихся областей $R_{i i=1}^{J}$, каждой из которых соответствует один лист дерева.

Тогда $h(x, a) = \sum_{j=1}^{J} a_j I\left[x \in R_j\right]$

Соответственно, $F_k = F_{k-1} + b_k \Big(\sum_{j=1}^{J} a_j I\left[x \in R_j\right] \Big) = F_{k-1} + \sum_{j=1}^{J} c_j I\left[x \in R_j\right]$, где $c_j = a_j b_j$. В таком случае, вместо того, чтобы сначала искать оптимальные a_i, а затем b_i, можно сразу искать оптимальные c_i:

$$c_{j_k} = \underset{c \in \mathbb{R}}{argmin} \sum_{x_i \in R_j} L(F_{k-1}(x_i) + c)$$