Композиции алгоритмов. Бустинг.

Кантонистова Е.О.

ВШЭ, 2019

БУСТИНГ

Идея: строим набор алгоритмов, каждый из которых исправляет ошибку предыдущих.

БУСТИНГ В ЗАДАЧЕ РЕГРЕССИИ

Решаем задачу регрессии с минимизацией квадратичной ошибки:

$$\frac{1}{2}\sum_{i=1}^{l}(a(x_i)-y_i)^2 \to \min_a$$

Ищем алгоритм $a(x)$ в виде суммы N базовых алгоритмов:

$$a(x)=\sum_{n=1}^{N} b_n(x)\,,$$

где базовые алгоритмы $b_n(x)$ принадлежат некоторому семейству A.

БУСТИНГ В ЗАДАЧЕ РЕГРЕССИИ

Шаг 1: Ищем алгоритм $b_1(x)$, минимизирующий ошибку:

$$b_1(x) = \underset{b \in A}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{l} (b(x_i) - y_i)^2$$

- Ошибка на i-м объекте:

$$s_i^{(1)} = y_i - b_1(x_i)$$

БУСТИНГ В ЗАДАЧЕ РЕГРЕССИИ

Шаг 1: Ищем алгоритм $b_1(x)$, минимизирующий ошибку:

$$b_1(x) = \underset{b \in A}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{l} (b(x_i) - y_i)^2$$

- Ошибка на i-м объекте:

$$s_i^{(1)} = y_i - b_1(x_i)$$

- Тогда $b_1(x_i) + s_i^{(1)} = y_i$

$\Rightarrow$ следующий алгоритм должен настраиваться на эти ошибки

БУСТИНГ В ЗАДАЧЕ РЕГРЕССИИ

Шаг 1: Ищем алгоритм $b_1(x)$, минимизирующий ошибку:

$$b_1(x) = \underset{b \in A}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{l} (b(x_i) - y_i)^2$$

- Ошибка на i-м объекте:

$$s_i^{(1)} = y_i - b_1(x_i)$$

- Тогда $b_1(x_i) + s_i^{(1)} = y_i$

$\Rightarrow$ следующий алгоритм должен настраиваться на эти ошибки:

если найдется алгоритм b_2: $b_2(x_i) = s_i^{(1)}$, то алгоритм $a(x) = b_1(x) + b_2(x)$ будет идеально предсказывать ответ.

БУСТИНГ В ЗАДАЧЕ РЕГРЕССИИ

Шаг 1: Ищем алгоритм $b_1(x)$, минимизирующий ошибку:

$$b_1(x) = \underset{b \in A}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{l} (b(x_i) - y_i)^2$$

Ошибка на i-м объекте:

$$s_i^{(1)} = y_i - b_1(x_i)$$

Шаг 2: Ищем алгоритм $b_2(x)$, настраивающийся на ошибки s_i первого алгоритма:

$$b_2(x) = \underset{b \in A}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{l} \left(b(x_i) - s_i^{(1)}\right)^2$$

БУСТИНГ В ЗАДАЧЕ РЕГРЕССИИ

Каждый следующий алгоритм настраиваем на ошибку предыдущих.

<u>Шаг N:</u> Ошибка: $s_i^{(N)} = y_i - \sum_{n=1}^{N-1} b_n(x_i) = y_i - a_{N-1}(x_i)$

Ищем алгоритм $b_N(x)$:

$$b_N(x) = \underset{b \in A}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{l} \left(b(x_i) - s_i^{(N)} \right)^2$$

БУСТИНГ В ЗАДАЧЕ РЕГРЕССИИ

Каждый следующий алгоритм настраиваем на ошибку предыдущих.

Шаг N: Ошибка: $s_i^{(N)} = y_i - \sum_{n=1}^{N-1} b_n(x_i) = y_i - a_{N-1}(x_i)$

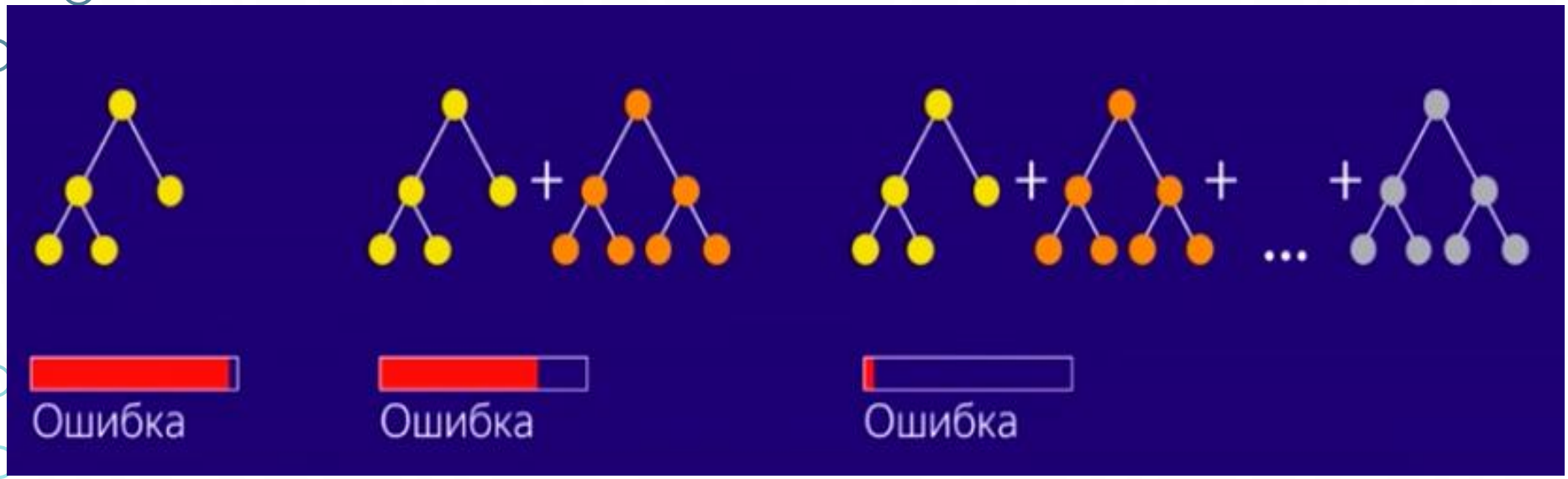
Ищем алгоритм $b_N(x)$:

$$b_N(x) = \underset{b \in A}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{l} \left(b(x_i) - s_i^{(N)} \right)^2$$

Утверждение. Ошибка на N-м шаге – это антиградиент функции потерь по ответу модели, вычисленный в точке ответа уже построенной композиции:

$$s_i^{(N)} = y_i - a_{N-1}(x_i) = -\frac{\partial}{\partial z} \frac{1}{2} (z - y_i)^2 \Big|_{z = a_{N-1}(x_i)}$$

БУСТИНГ



ГРАДИЕНТНЫЙ БУСТИНГ

Пусть $L(y, z)$ – произвольная дифференцируемая функция потерь.

Строим алгоритм $a_N(x)$ вида

$$a_N(x) = \sum_{n=1}^{N} \gamma_n b_n(x),$$

где на N-м шаге

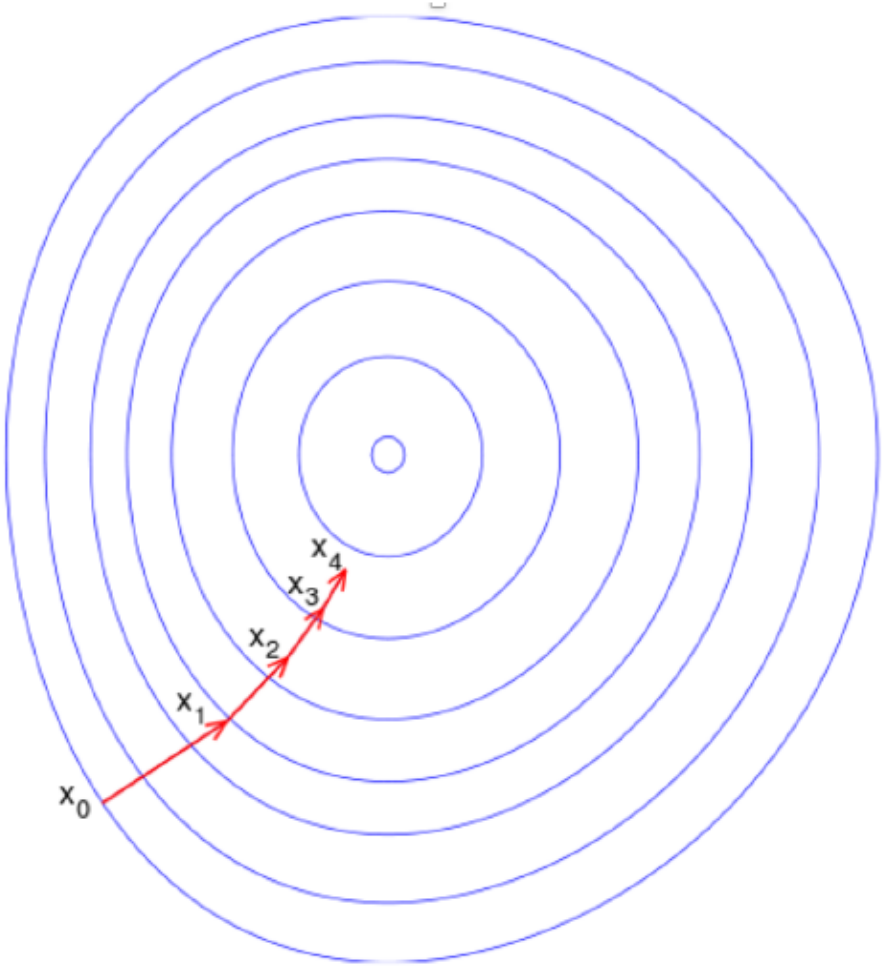
$$b_N(x) = \underset{b \in A}{\operatorname{argmin}} \sum_{i=1}^{l} \left(b(x_i) - s_i^{(N)} \right)^2,$$

$$s_i^{(N)} = -\frac{\partial L}{\partial z}$$

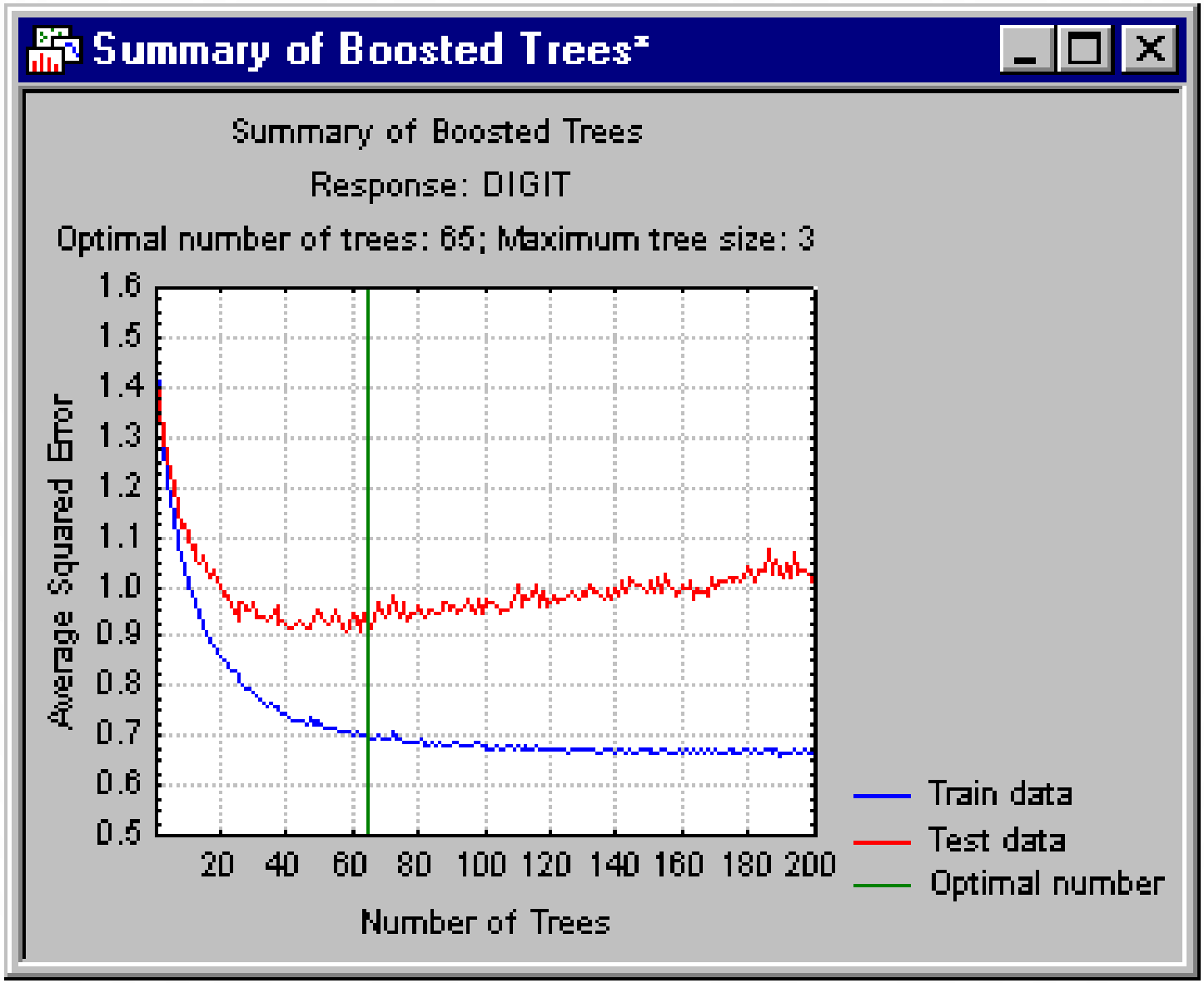
Коэффициент γ_N должен минимизировать ошибку:

$$\gamma_N = \min_{\gamma \in \mathbb{R}} \sum_{i=1}^{l} L(y_i, a_{N-1}(x_i) + \gamma_N b_N(x_i))$$

ГРАДИЕНТНЫЙ СПУСК В ПРОСТРАНСТВЕ ФУНКЦИЙ



КОЛИЧЕСТВО ИТЕРАЦИЙ БУСТИНГА



СТОХАСТИЧЕСКИЙ ГРАДИЕНТНЫЙ БУСТИНГ

- Будем обучать базовый алгоритм b_N не по всей выборке X, а по случайной подвыборке $X^k \subset X$.

+: снижается уровень шума в данных

+: вычисления становятся быстрее

Обычно берут $\left|X^k\right| = \frac{1}{2}|X|$.

СМЕЩЕНИЕ И РАЗБРОС

- Бустинг целенаправленно уменьшает ошибку, т.е. смещение у него маленькое.
- Алгоритм получается сложным, поэтому разброс большой.

Значит, чтобы не переобучиться, в качестве базовых алгоритмов надо брать неглубокие деревья (глубины 3-6).