

Введение в анализ данных

Лекция 10

Градиентный бустинг

Евгений Соколов

esokolov@hse.ru

НИУ ВШЭ, 2019

Случайный лес (Random forest)

1. Для $n = 1, \dots, N$:
2. Сгенерировать выборку \tilde{X} с помощью бутстрапа
3. Построить решающее дерево $b_n(x)$ по выборке \tilde{X}
4. Дерево строится, пока в каждом листе не окажется не более n_{min} объектов
5. Оптимальное разбиение ищется среди q случайных признаков

Чем плох случайный лес?

- Нужны глубокие деревья, могут очень долго обучаться
- Если одно дерево не справляется с задачей, то усреднение вряд ли поможет

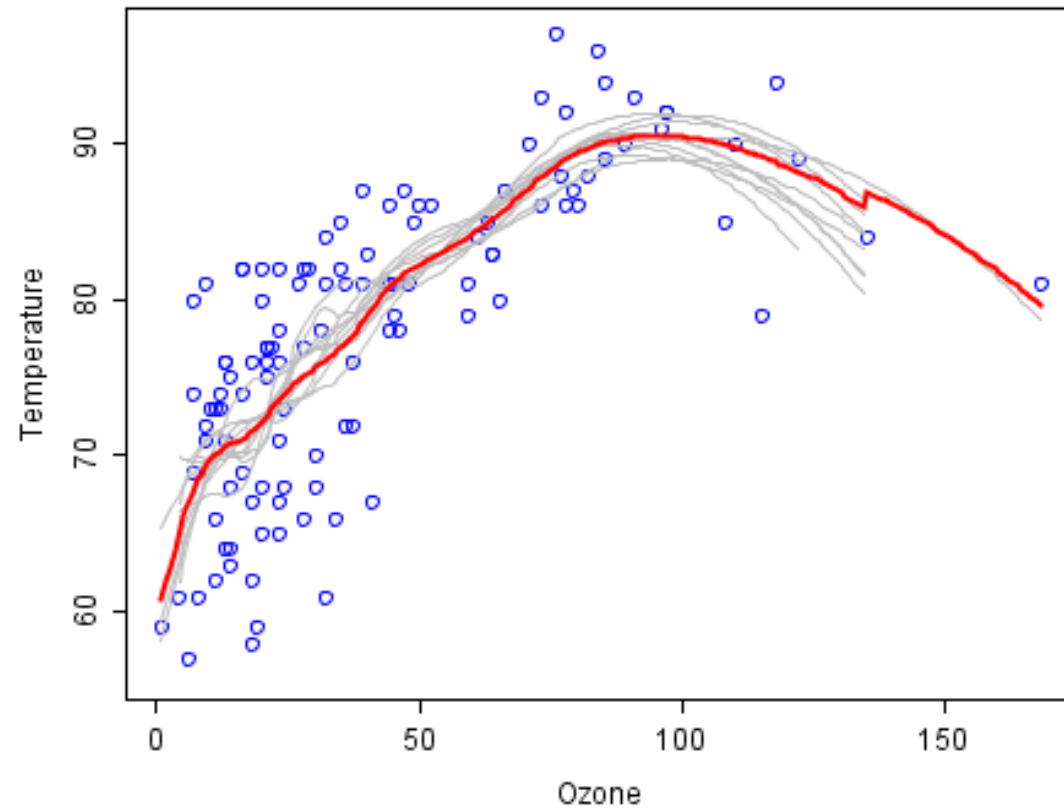
Bias-variance decomposition

$$\begin{aligned} L(\mu) = & \underbrace{\mathbb{E}_{x,y} \left[(y - \mathbb{E}[y | x])^2 \right]}_{\text{шум}} + \\ & + \underbrace{\mathbb{E}_x \left[(\mathbb{E}_X [\mu(X)] - \mathbb{E}[y | x])^2 \right]}_{\text{смещение}} + \underbrace{\mathbb{E}_x \left[\mathbb{E}_X \left[(\mu(X) - \mathbb{E}_X [\mu(X)])^2 \right] \right]}_{\text{разброс}} \end{aligned}$$

Bias-variance decomposition

- Можно показать, что ошибка метода обучения раскладывается на три слагаемых: шум, смещение, разброс
- Шум — как сильно ошибается лучшая модель
- Смещение — как сильно в среднем отклоняется наша модель от лучшей модели
- Разброс — как сильно может меняться модель, если немного поменять обучающую выборку

Bias-variance decomposition



Смещение и разброс в беггинге

Можно показать, что в беггинге:

- Смещение композиции такое же, как у одной модели
- Разброс уменьшается тем сильнее, чем меньше корреляция между базовыми моделями
 - Поэтому в случайном лесе мы придумывали способы повышения разнообразия моделей
- Вывод: если дерево имеет высокое смещение, то беггинг не даст хороший результат

Градиентный бустинг

Идея бустинга

- Будем обучать каждую следующую модель в композиции так, чтобы она исправляла ошибки предыдущих моделей

Бустинг для MSE

- Композиция:

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

- Обучим первый базовый алгоритм как обычно (например, стандартная процедура обучения дерева для регрессии):

$$\frac{1}{\ell} \sum_{i=1}^{\ell} (b_1(x_i) - y_i)^2 \rightarrow \min_{b_1}$$

Бустинг для MSE

- Вторая базовая модель должна корректировать ошибки первой:

$$b_2(x_i) \approx y_i - b_1(x_i)$$

- Если получится этого добиться, то

$$b_1(x_i) + b_2(x_i) \approx y_i$$

- Значит, вторую модель обучаем так:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left(b_2(x_i) - (y_i - b_1(x_i)) \right)^2 \rightarrow \min_{b_2}$$

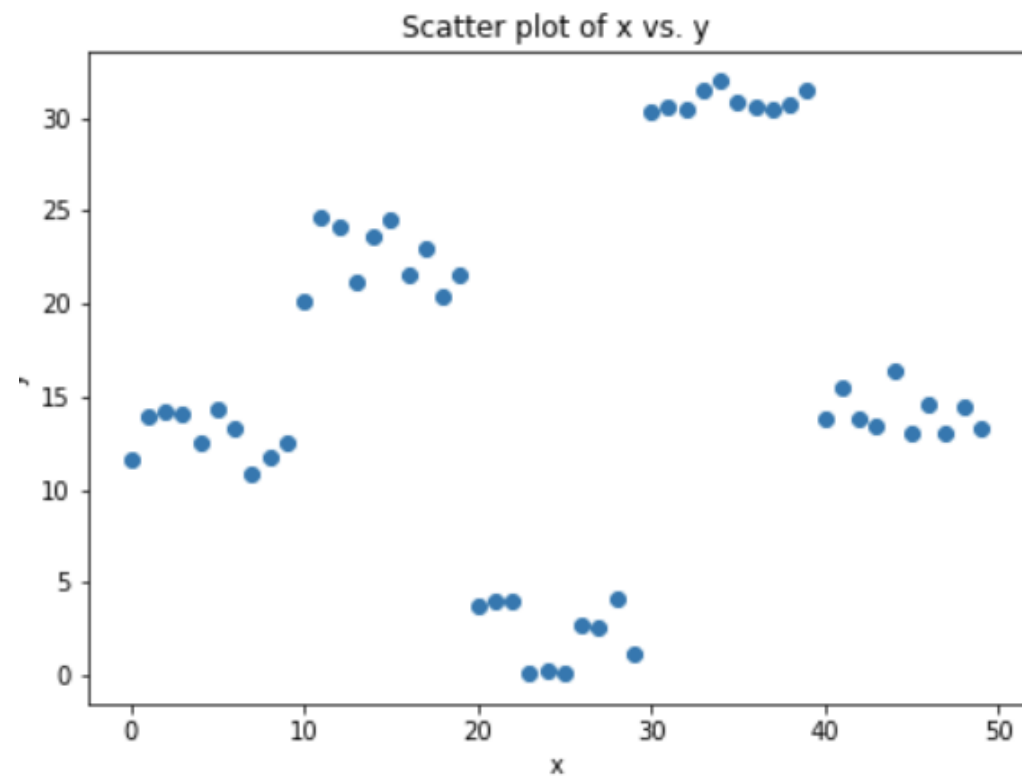
- b_1 тут уже фиксирован!

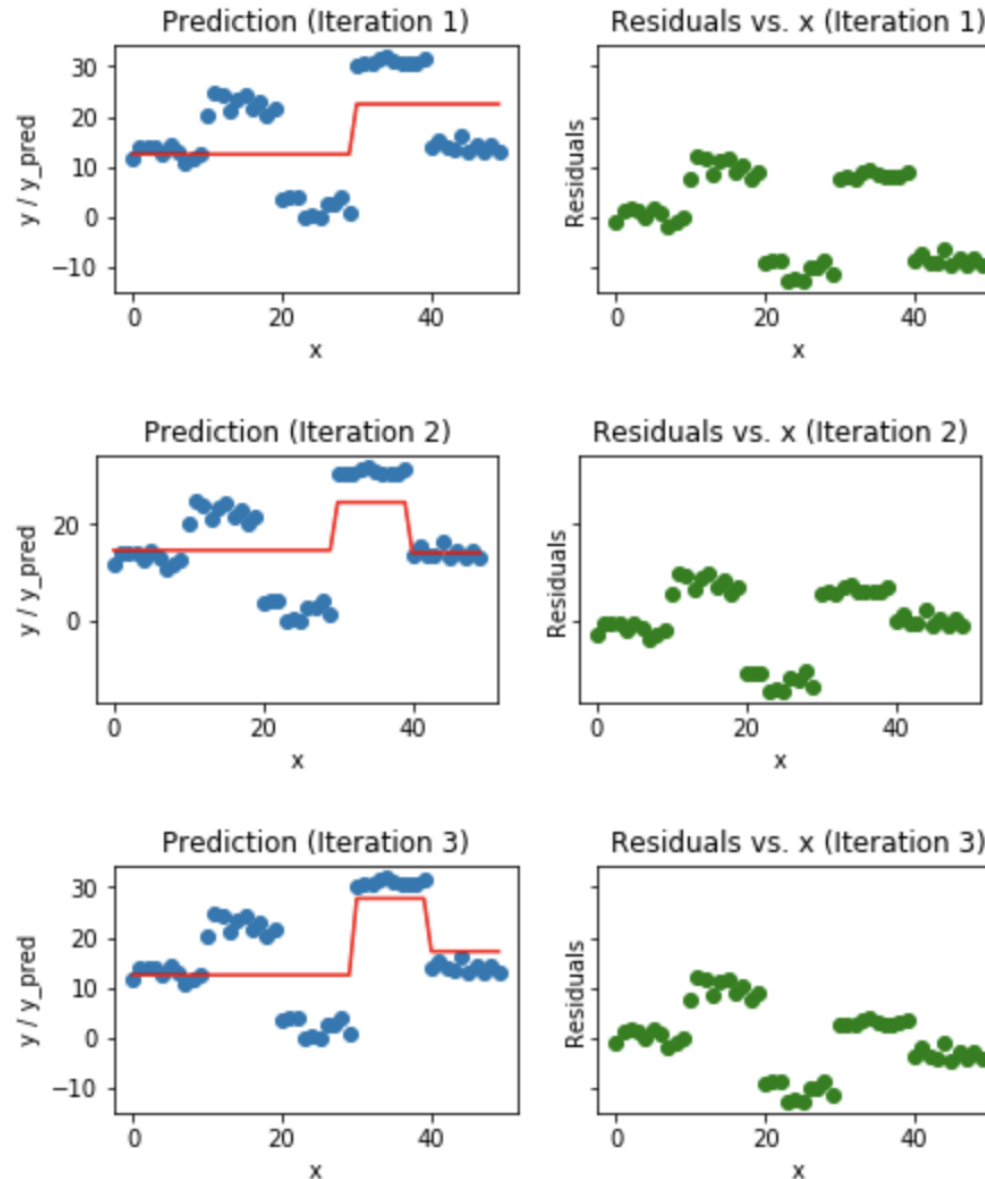
Бустинг для MSE

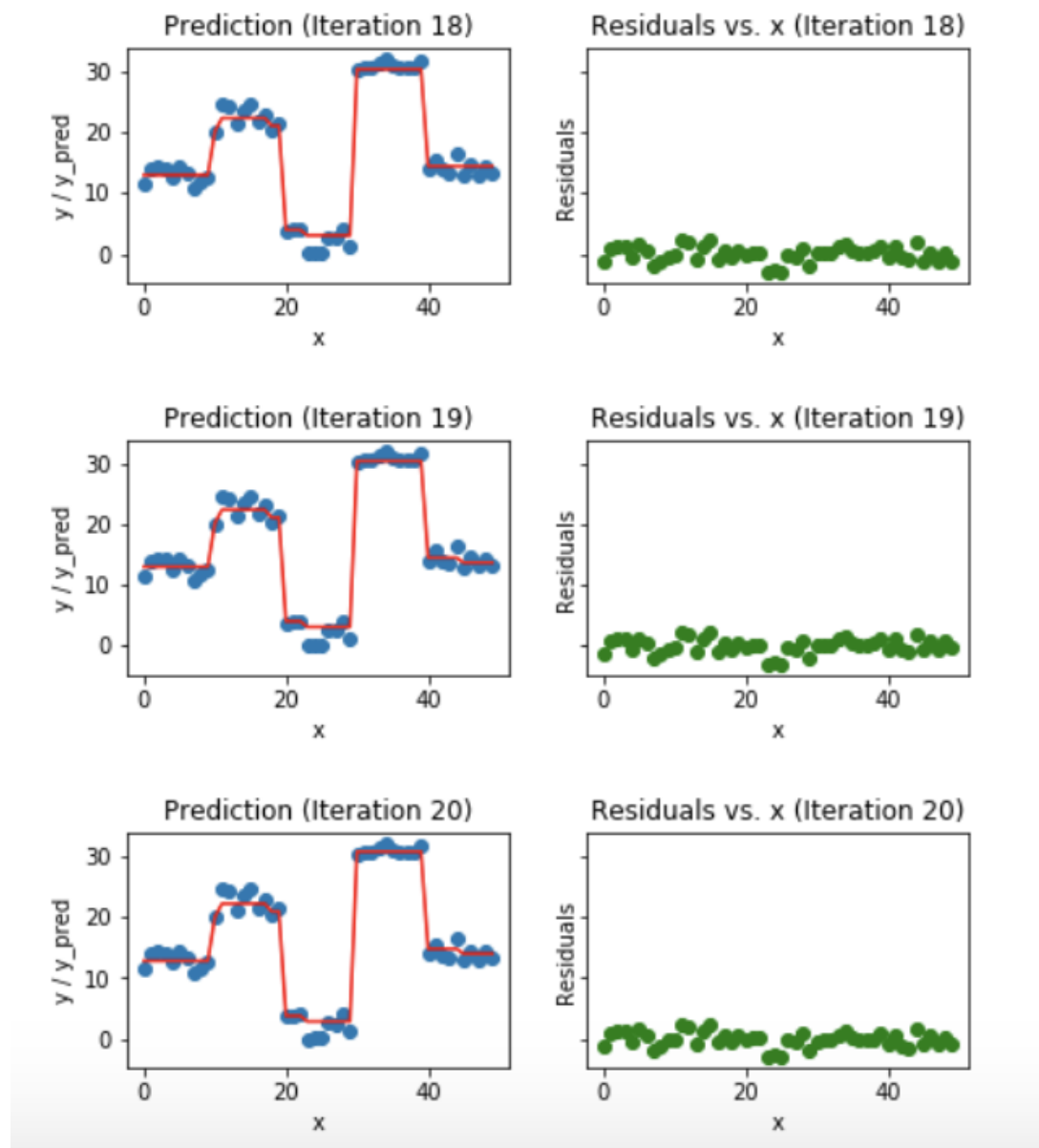
- И так далее:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left(b_3(x_i) - (y_i - b_1(x_i) + b_2(x_i)) \right)^2 \rightarrow \min_{b_3}$$

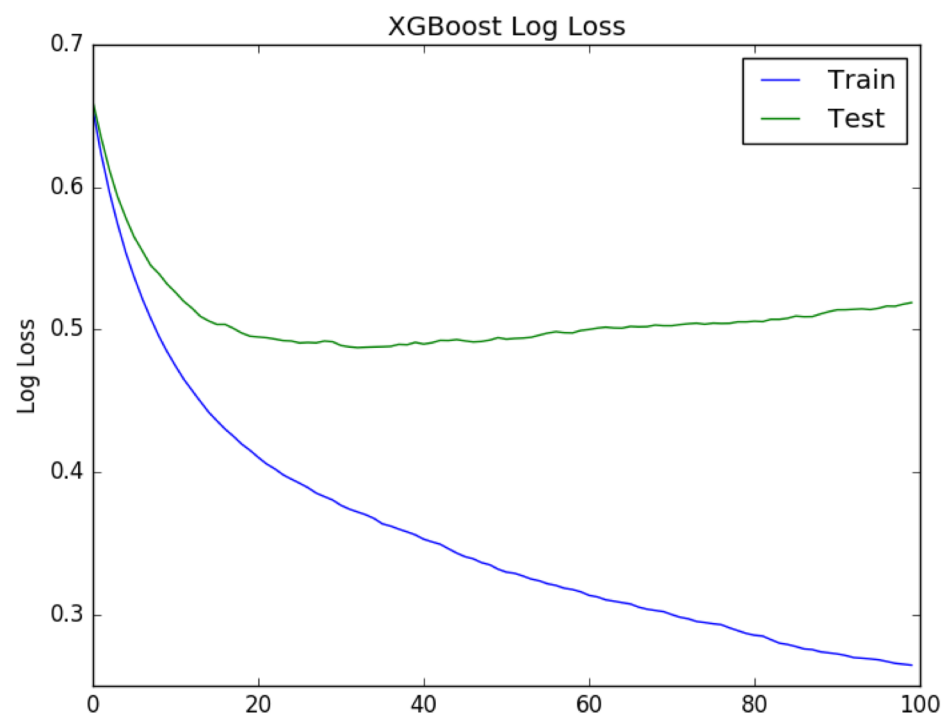
$$\frac{1}{\ell} \sum_{i=1}^{\ell} \left(b_4(x_i) - (y_i - b_1(x_i) + b_2(x_i) + b_3(x_i)) \right)^2 \rightarrow \min_{b_4}$$







Бустинг для MSE



Бустинг для MSE

- Переобучается по мере роста числа базовых моделей (в отличие от случайного леса)
 - Композиция деревьев с помощью бустинга **понижает** смещение и **повышает** разброс
 - Значит, базовые модели — неглубокие деревья (где-то от 1 до 6 уровней)
-
- Для сравнения: беггинг **не меняет** смещение и **понижает** разброс
 - Поэтому базовые модели — глубокие деревья

Градиентный бустинг в общем случае

- Задача обучения в общем виде:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, a(x_i)) \rightarrow \min_a$$

Градиентный бустинг в общем случае

- Допустим, мы уже обучили (N-1)-ую базовую модель:

$$a_{N-1}(x) = \sum_{n=1}^{N-1} b_n(x)$$

- Задача обучения N-й модели:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b_N(x_i)) \rightarrow \min_{b_N}$$

Градиентный бустинг в общем случае

- Для MSE есть важное свойство:

$$L(y, a + b) = ((a + b) - y)^2 = (b - (y - a))^2 = L(y - a, b)$$

- Поэтому задача обучения очередной базовой модели сводится к замене целевой переменной:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b_N(x_i)) \rightarrow \min_{b_N}$$

$$\frac{1}{\ell} \sum_{i=1}^{\ell} L(y_i - a_{N-1}(x_i), b_N(x_i)) \rightarrow \min_{b_N}$$

Градиентный бустинг в общем случае

- Но далеко не всегда это свойство выполнено!
- Например, для логистической функции потерь оно не работает

$$L(y, a) = \log(1 + \exp(-ya))$$

Градиентный бустинг в общем случае

Можно показать, что задача

$$\sum_{i=1}^{\ell} L(y_i, a_{N-1}(x_i) + b_N(x_i)) \rightarrow \min_{b_N}$$

примерно совпадает с задачей

$$\sum_{i=1}^{\ell} (b_N(x_i) - s_i)^2 \rightarrow \min_{b_N}$$

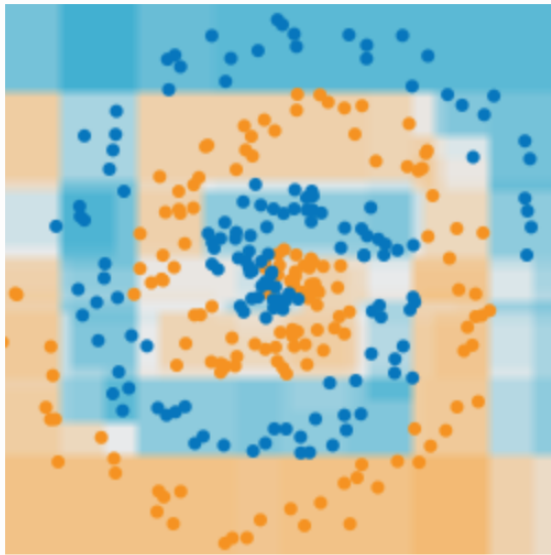
Где

$$s_i = - \left. \frac{\partial L}{\partial z} \right|_{z=a_{N-1}(x_i)}$$

Градиентный бустинг в общем случае

- Задачу построения следующей модели в композиции можно свести к задаче регрессии с новой целевой переменной
- Новая целевая переменная — производная функции потерь в точке текущего прогноза
- Мы как бы строим новую модель, чтобы она как можно сильнее снизила ошибку композиции

Prediction:

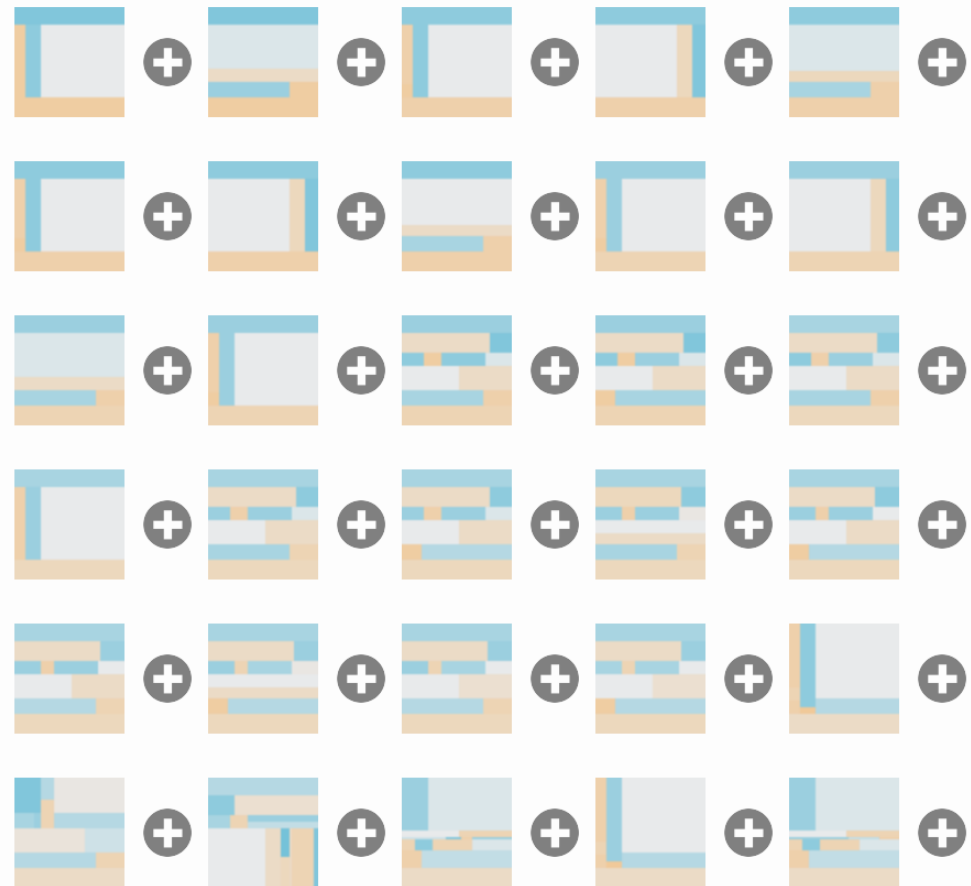


↑
predictions of GB (all 100 trees)

train loss: 0.341 test loss: 0.405



Decision functions of first 30 trees



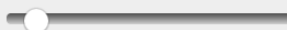
tree depth: 4



subsample: 100%



learning rate: 0.1



trees: 100



rotate dataset:



☐ rotate trees

☒ show gradients on hover

☐ use Newton-Raphson update

Обучение градиентного бустинга

- Основные гиперпараметры:
 - Число деревьев
 - Размер шага
 - Глубина дерева
- В имплементациях могут быть и другие важные настройки
 - Регуляризация
 - Семплирование объектов
 - и т.д.

Длина шага

- Базовые модели — неглубокие деревья с низким качеством
- Вряд ли им можно доверять
- Из-за принципа обучения градиентный бустинг может быстро вывести ошибку на обучении в ноль
- Нужно замедлять обучение!

$$a_N(x) = a_{N-1}(x) + \gamma b_N(x)$$

- $\gamma > 0$ — аналог длины шага в градиентном спуске

