Методы машинного обучения

Лекция 13

Градиентный бустинг

Эльвира Зиннурова

elvirazinnurova@gmail.com

НИУ ВШЭ, 2019

Случайный лес (Random forest)

1. Для $n = 1, \ldots, N$:
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

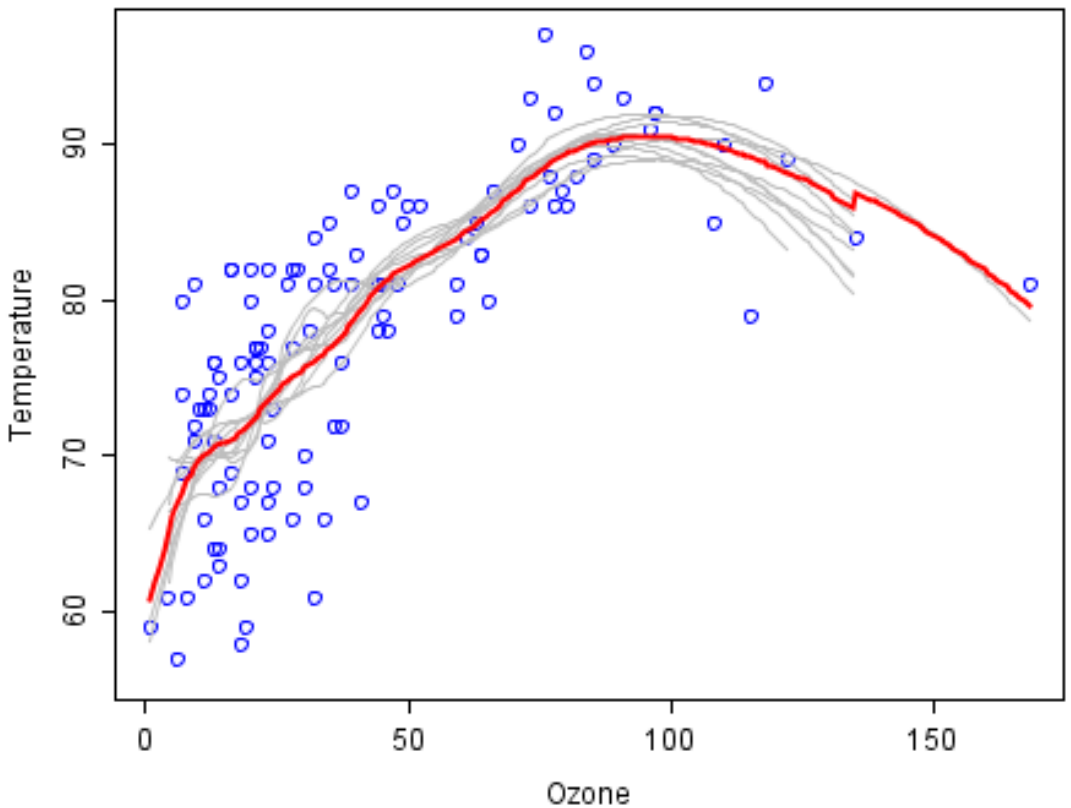
$$L(\mu) = \underbrace{\mathbb{E}_{x,y}\Big[\big(y - \mathbb{E}[y \mid x]\big)^2\Big]}_{\text{шум}} +$$

$$+ \underbrace{\mathbb{E}_x\Big[\big(\mathbb{E}_X\big[\mu(X)\big] - \mathbb{E}[y \mid x]\big)^2\Big]}_{\text{смещение}} + \underbrace{\mathbb{E}_x\Big[\mathbb{E}_X\Big[\big(\mu(X) - \mathbb{E}_X\big[\mu(X)\big]\big)^2\Big]\Big]}_{\text{разброс}}$$

Bias-variance decomposition

- Можно показать, что ошибка метода обучение 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 бэггинге

Можно показать, что в бэггинге:

- 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 бэггинг 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):

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

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:

$$\frac{1}{\ell}\sum_{i=1}^{\ell}\Big(b_2(x_i) - \big(y_i - b_1(x_i)\big)\Big)^2 \to \min_{b_2}$$

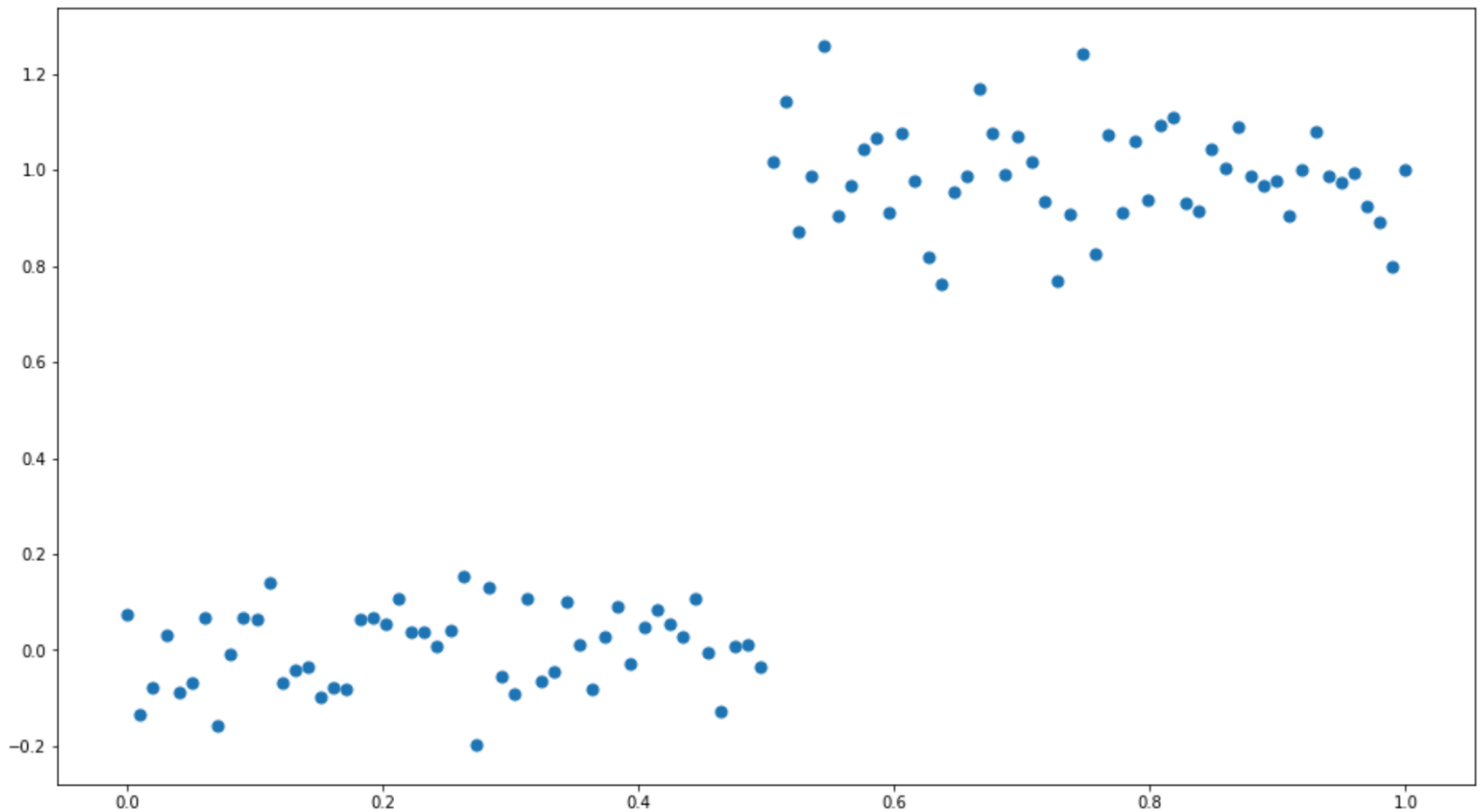
- b_1 тут уже фиксирован!

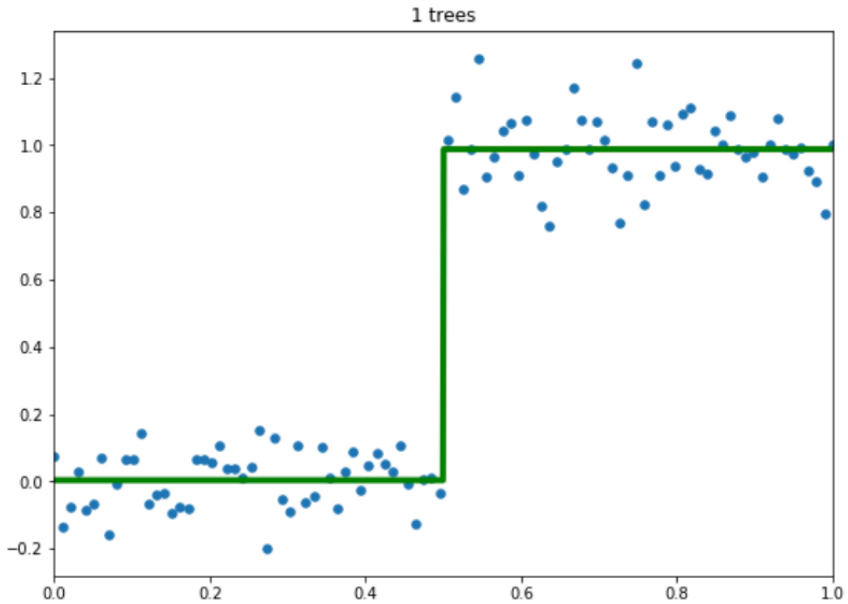
Бустинг для MSE

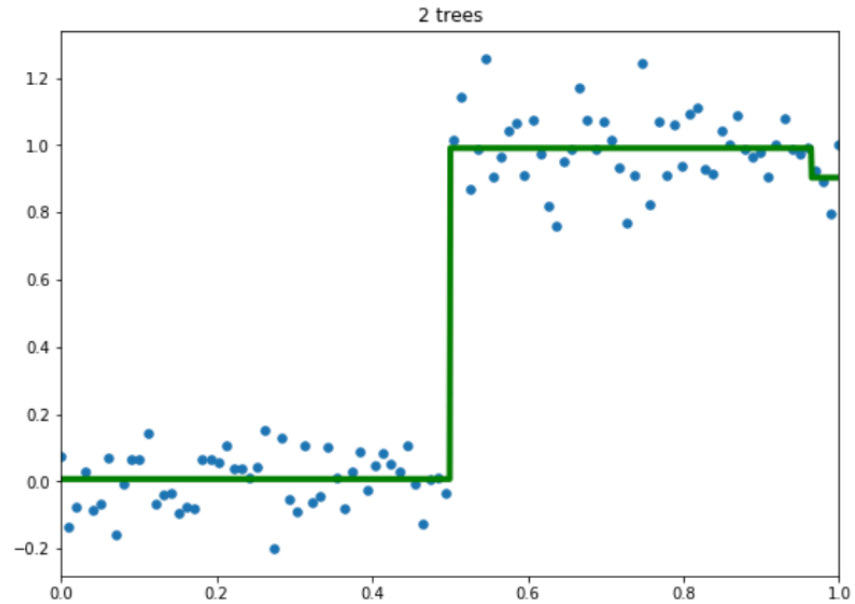
- И так далее:

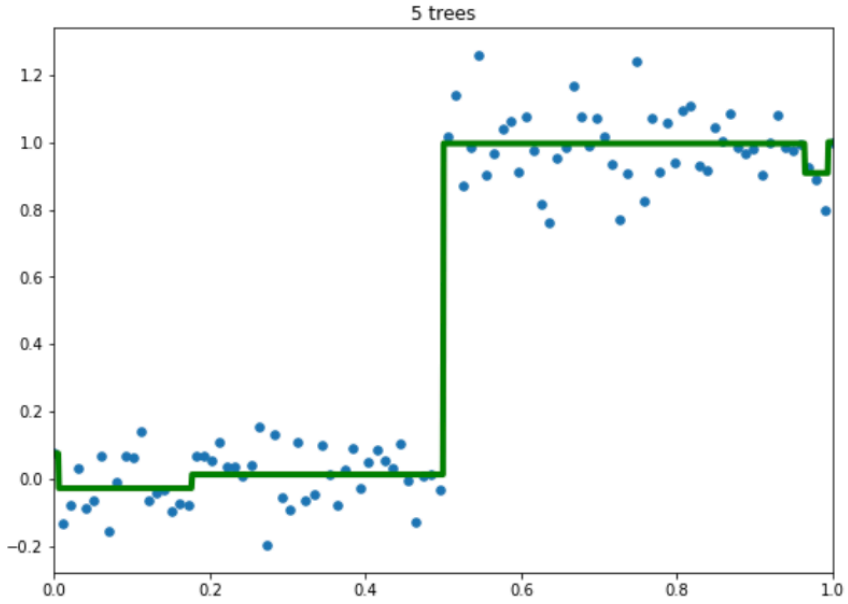
$$\frac{1}{\ell}\sum_{i=1}^{\ell}\Big(b_3(x_i) - \big(y_i - b_1(x_i) - b_2(x_i)\big)\Big)^2 \to \min_{b_3}$$

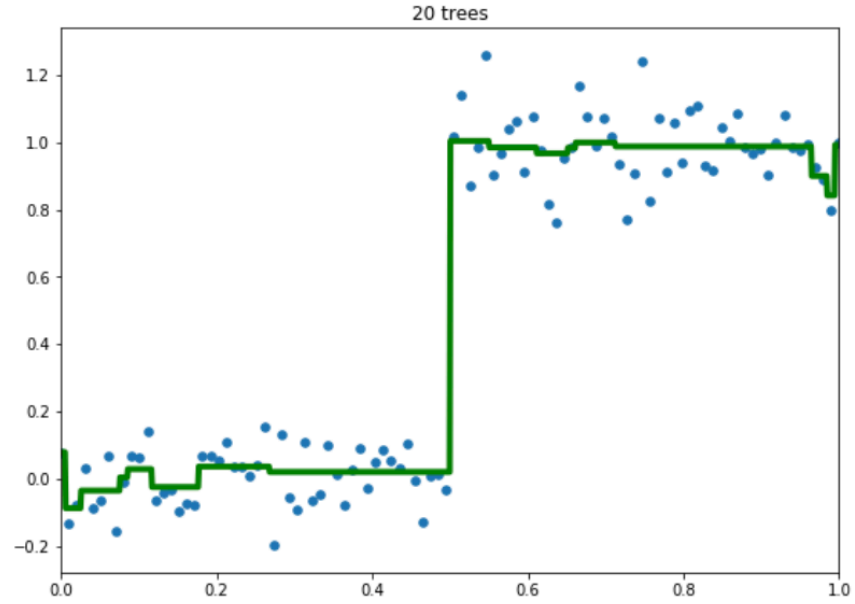
$$\frac{1}{\ell}\sum_{i=1}^{\ell}\Big(b_4(x_i) - \big(y_i - b_1(x_i) - b_2(x_i) - b_3(x_i)\big)\Big)^2 \to \min_{b_4}$$

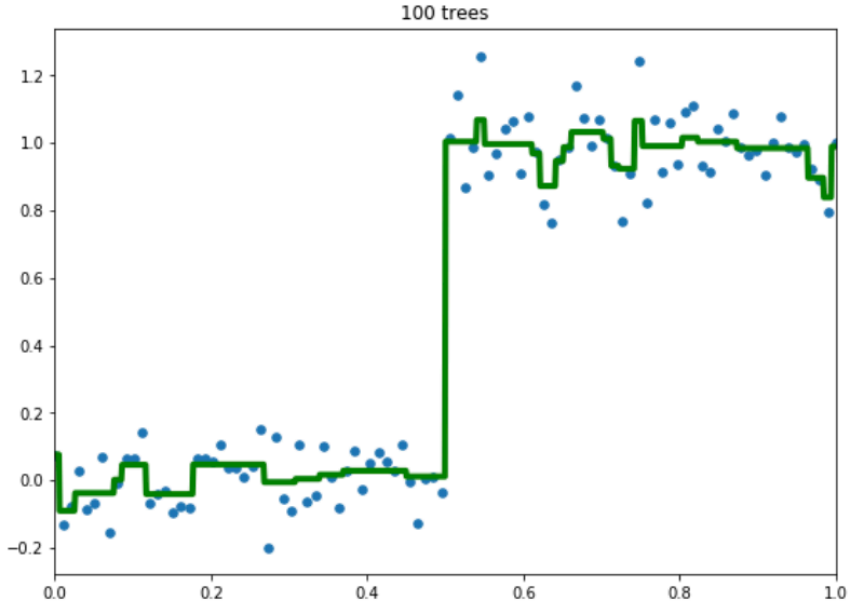

1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0

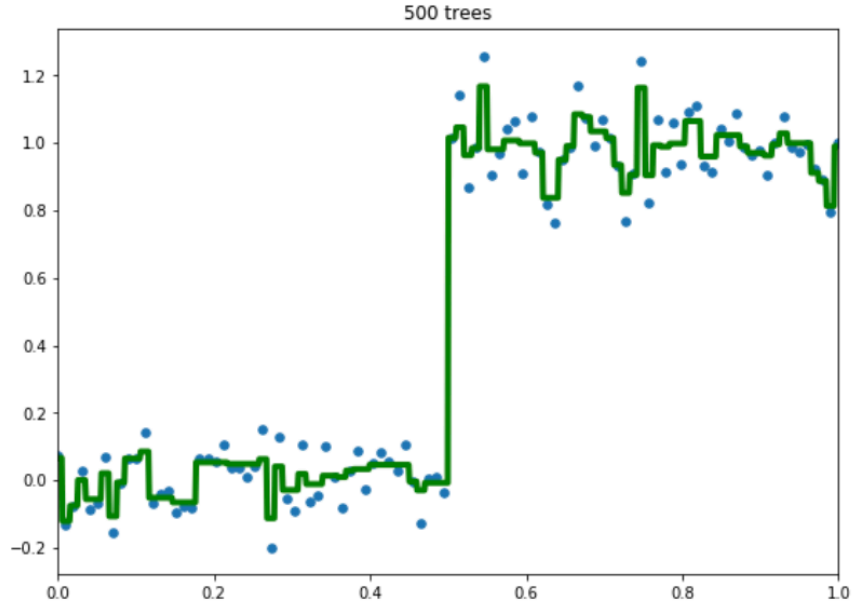

1 trees
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0

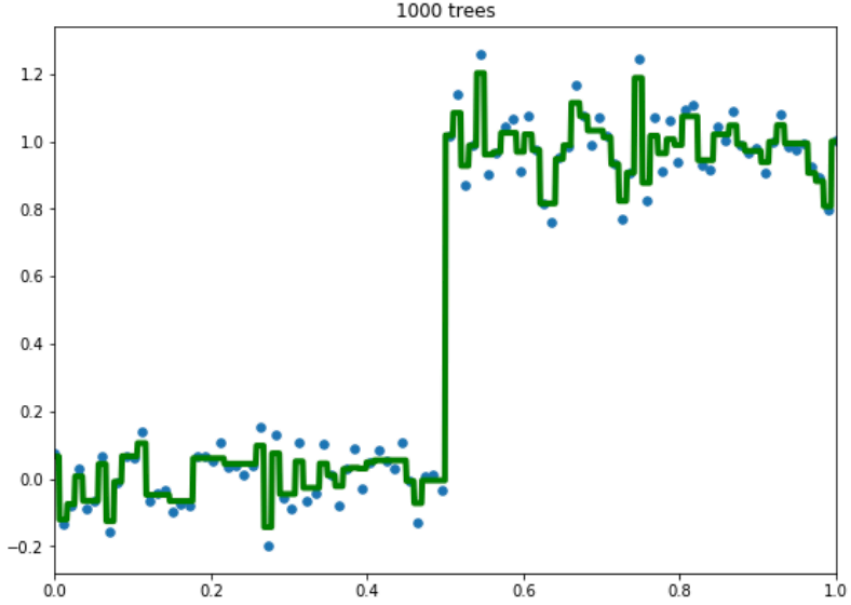

2 trees
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0

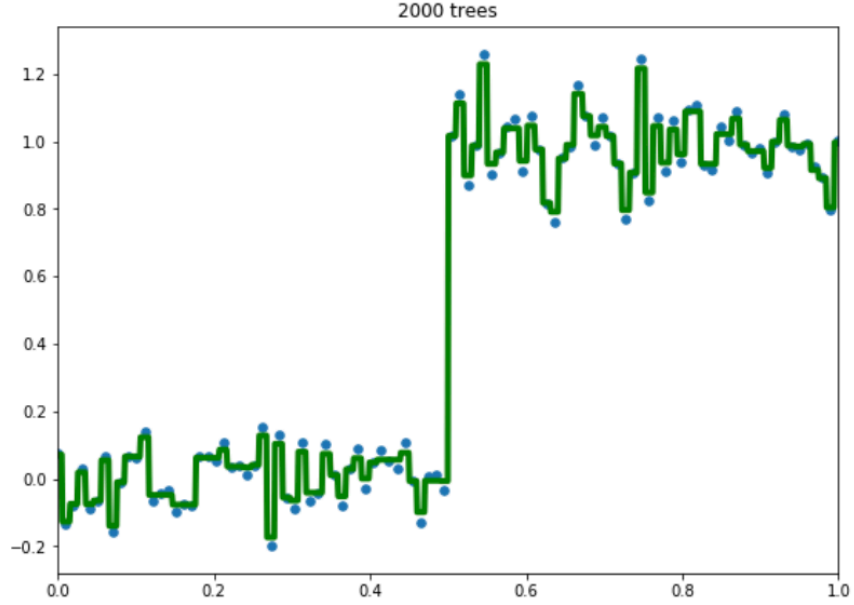

5 trees
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0


20 trees
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0

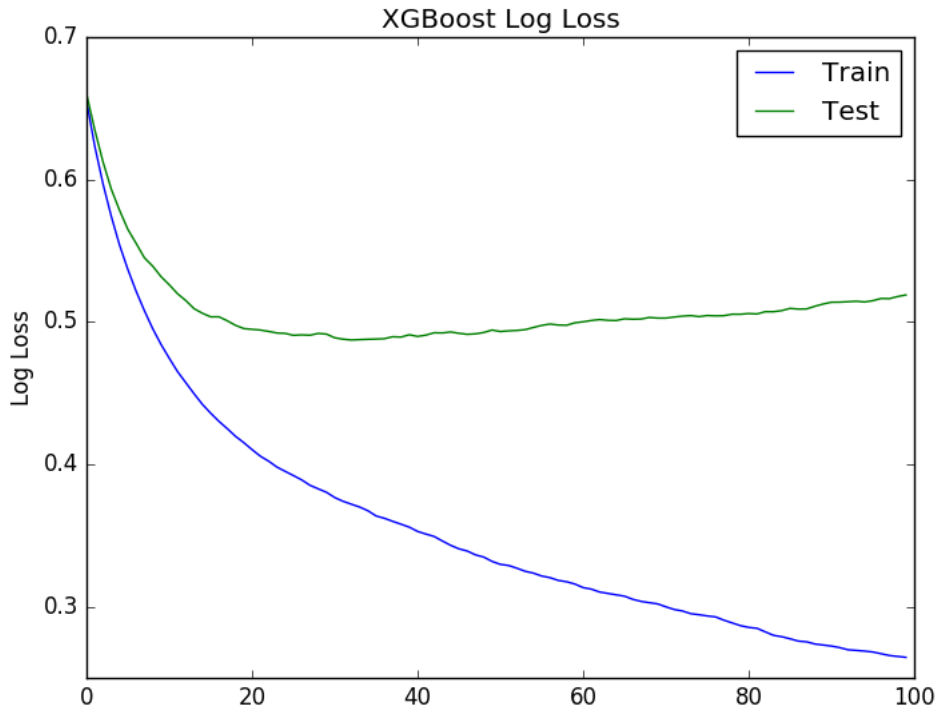

100 trees
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0


500 trees
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0


1000 trees
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0


2000 trees
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0

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: бэггинг **не меняет** смещение и **понижает** разброс
- Поэтому базовые модели — глубокие деревья

Градиентный бустинг в общем случае

- Задача для MSE:

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

- Задача обучения в общем виде:

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

Градиентный бустинг в общем случае

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

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

Градиентный бустинг в общем случае

- Задача обучения N-й модели для MSE:

$$\frac{1}{\ell}\sum_{i=1}^{\ell}(y_i - (a_{N-1}(x_i) + b_N(x_i)))^2 \to \min_{b_N}$$

- Взяв производную одного слагаемого и приравняв нулю, получим:

$$b_N(x_i) \approx y_i - a_{N-1}(x_i)$$

- В общем случае так не получится:

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

Градиентный бустинг в общем случае

- Например, для логистической функции потерь

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

Градиентный бустинг в общем случае

Можно показать, что задача

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

примерно совпадает с задачей

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

Где

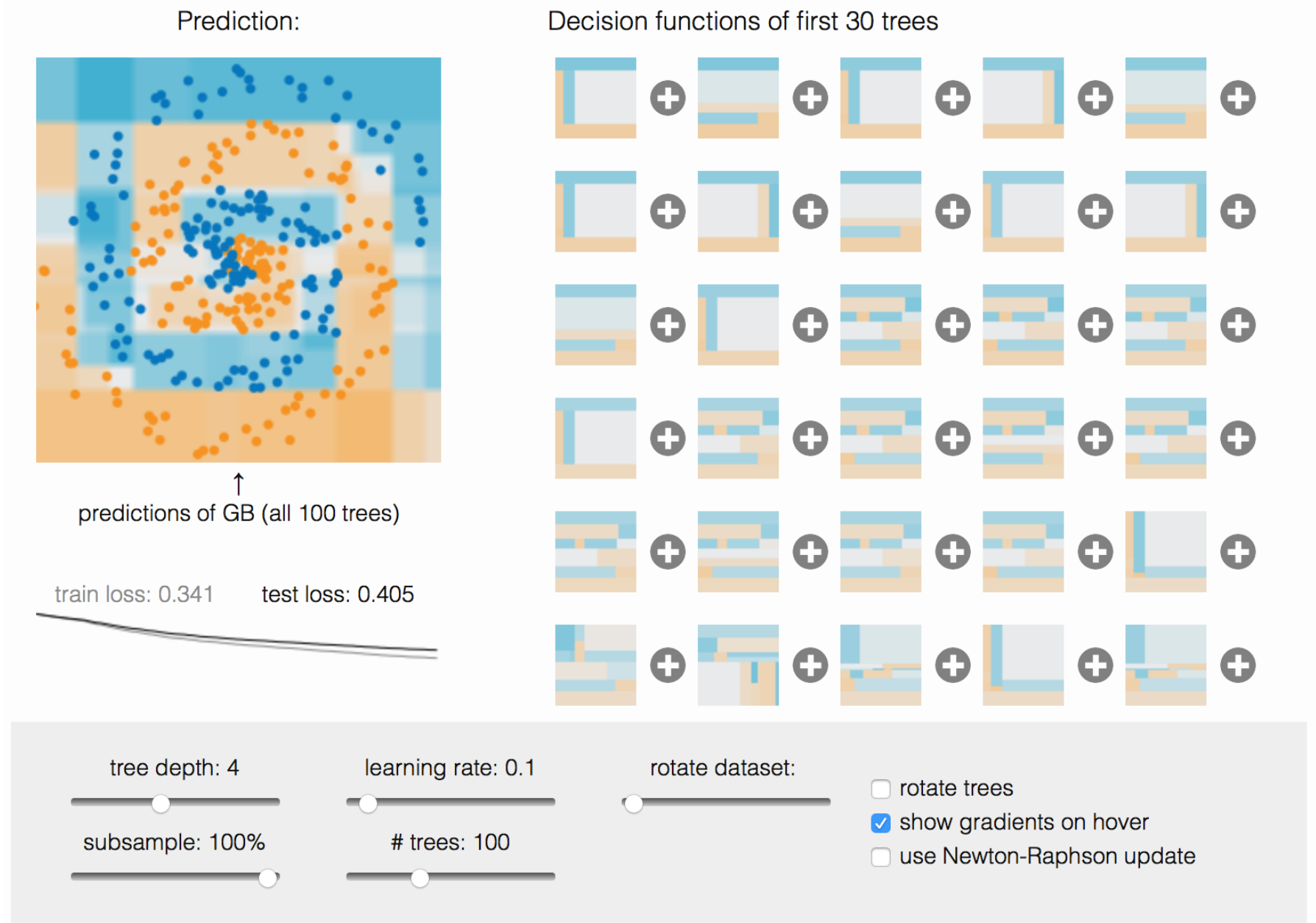
$$s_i = -\frac{\partial L}{\partial z}\bigg|_{z=a_{N-1}(x_i)}$$

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

Градиентный бустинг

1. Обучить алгоритм $b_1(x)$ на выборке $X = \{(x_i, y_i)\}_{i=1}^{l}$
2. Для $n = 2, \ldots, N$:
3. Вычислить сдвиги для объектов $s_i = -\frac{\partial L}{\partial z}\Big|_{z=a_{n-1}(x_i)}$
4. Обучить алгоритм $b_n(x)$ на выборке $\{(x_i, s_i)\}_{i=1}^{l}$ и добавить его в композицию



http://arogozhnikov.github.io/2016/07/05/gradient_boosting_playground.html

Обучение градиентного бустинга

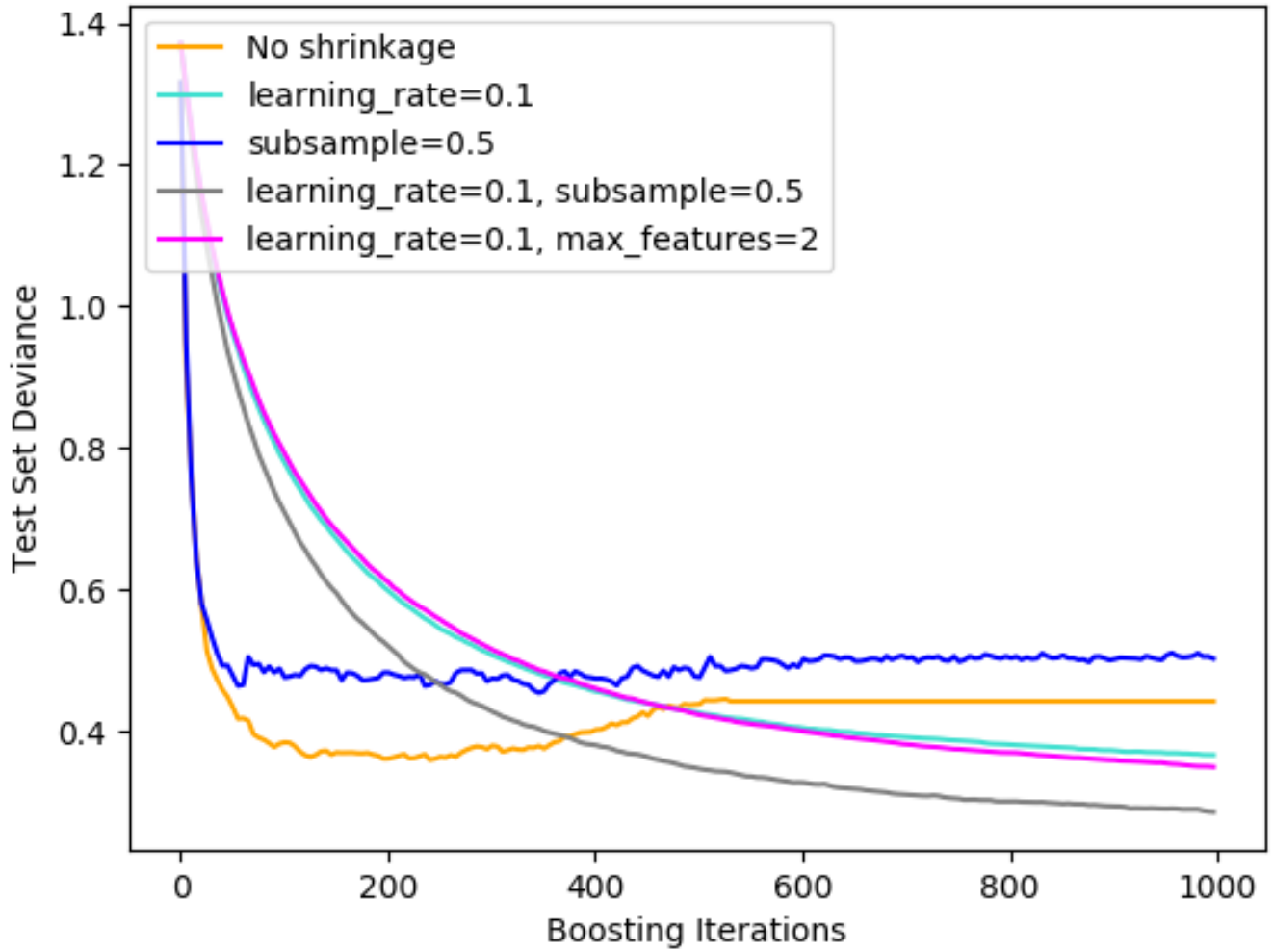
- Основные гиперпараметры:
  - Число деревьев
  - Размер шага
  - Глубина дерева
- В реализациях могут быть и другие важные настройки
  - Регуляризация
  - Семплирование объектов
  - и т.д.

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


No shrinkage
learning_rate=0.1
subsample=0.5
learning_rate=0.1, subsample=0.5
learning_rate=0.1, max_features=2
Test Set Deviance
Boosting Iterations
1.4
1.2
1.0
0.8
0.6
0.4
0
200
400
600
800
1000

Резюме

- В градиентном бустинге обучаем каждую следующую модель на ошибки предыдущих
- Лучше использовать неглубокие древья в качестве базовых моделей
- Быстро переобучается — используем длину шага для борьбы