Введение в анализ данных

Лекция 10

Градиентный бустинг

Евгений Соколов

esokolov@hse.ru

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

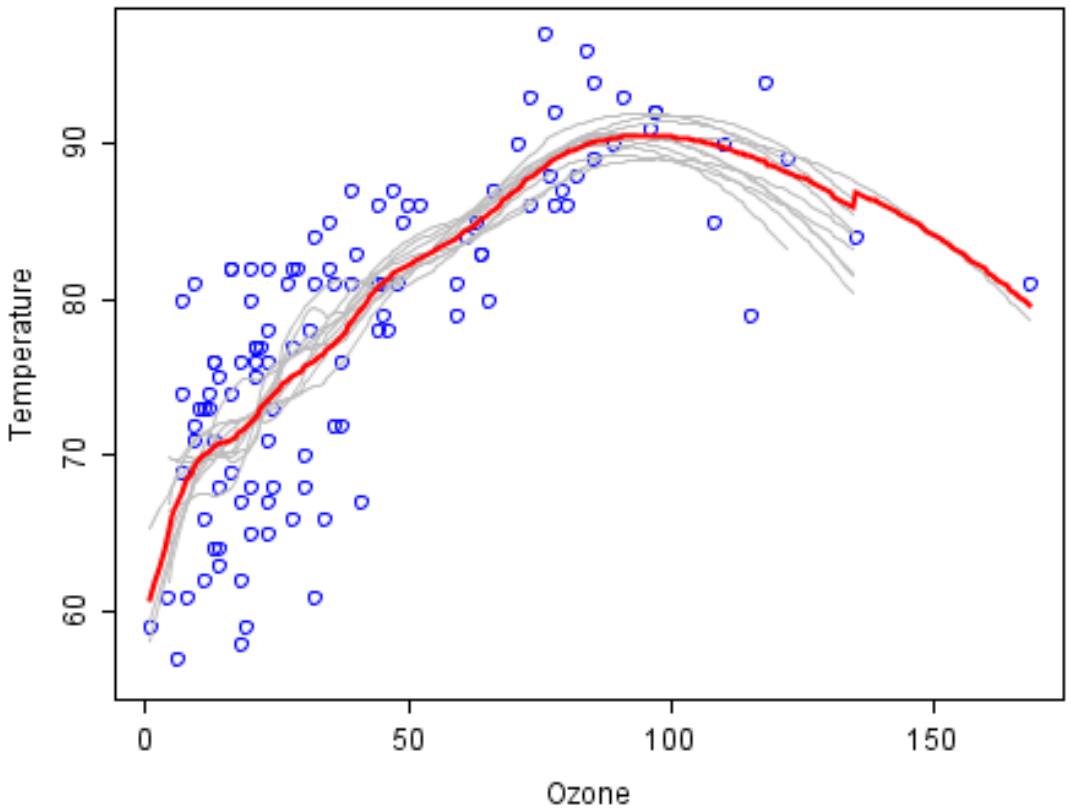
$$L(\mu) = \underbrace{\mathbb{E}_{x,y}\Big[\big(y - \mathbb{E}[y \mid x]\big)^2\Big]}_{\text{шум}} +$$

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

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 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):

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

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:

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

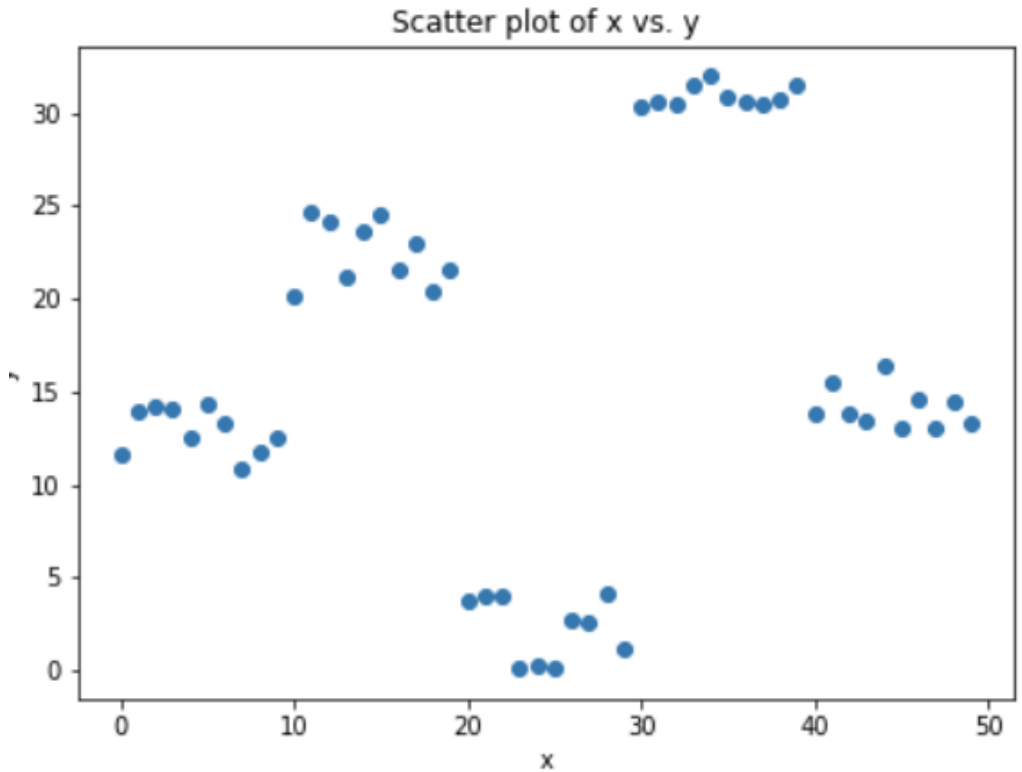
- b_1 тут уже фиксирован!

Бустинг для MSE

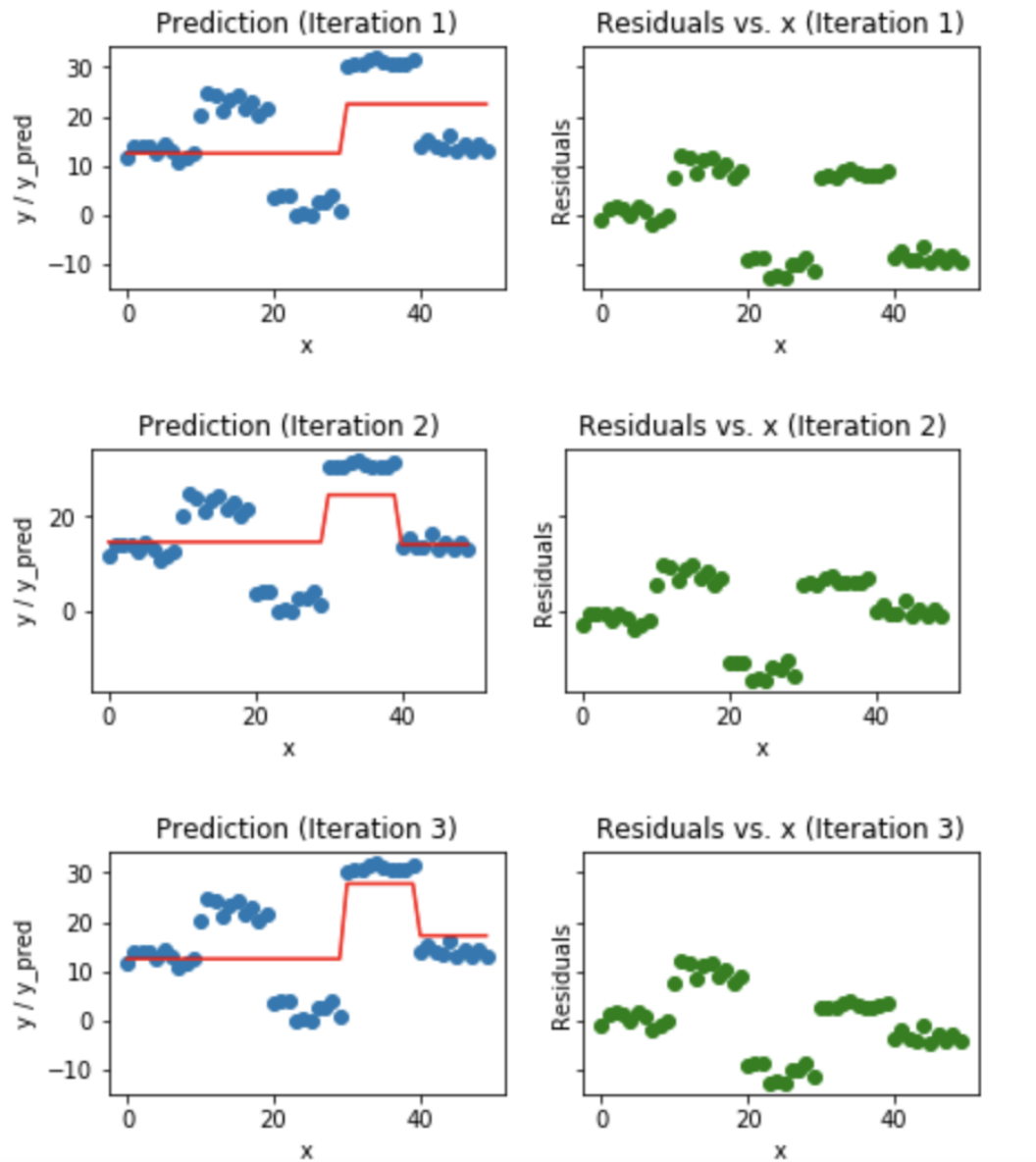
- И так далее:

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

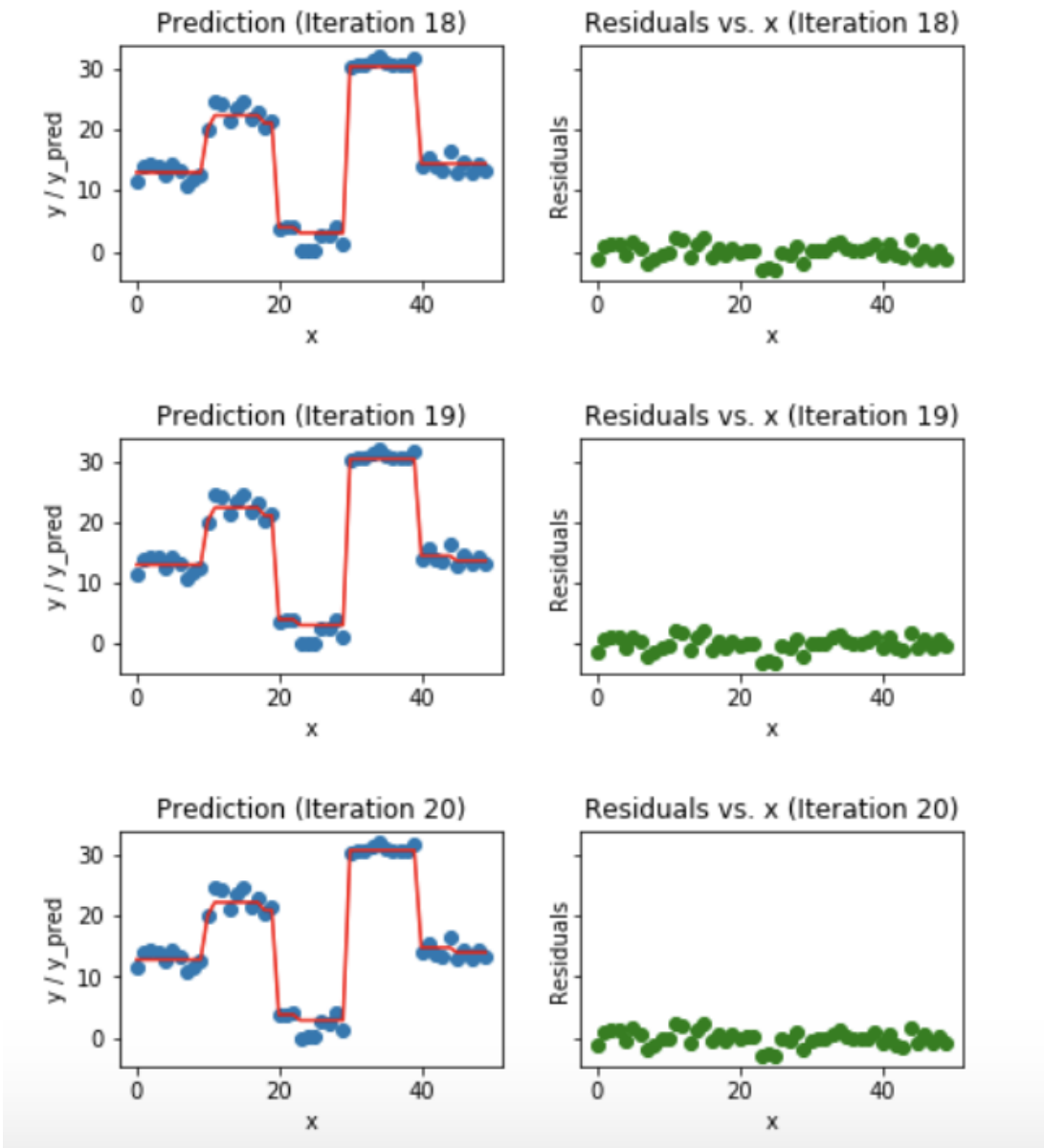
$$\frac{1}{\ell}\sum_{i=1}^{\ell}\big(b_4(x_i) - (b_1(x_i) + b_2(x_i) + b_3(x_i) - y_i)\big)^2 \to \min_{b_4}$$



https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d

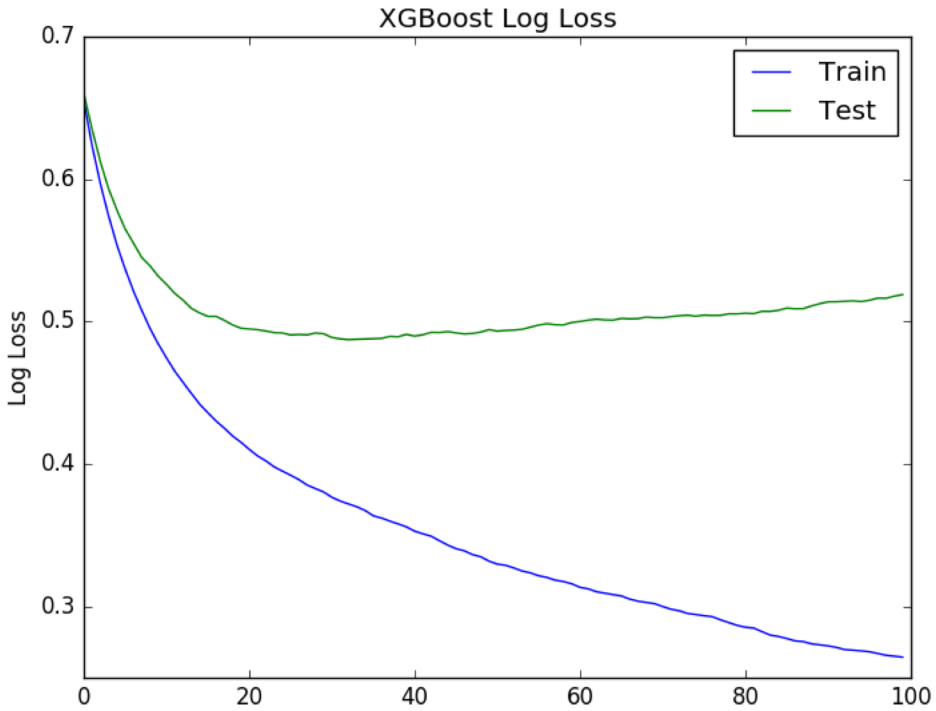


https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d



https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d

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:

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

Градиентный бустинг в общем случае

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

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

- Задача обучения N-й модели:

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

Градиентный бустинг в общем случае

- Для MSE есть важное свойство:

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

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

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

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

Градиентный бустинг в общем случае

- 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

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

примерно совпадает с задачей

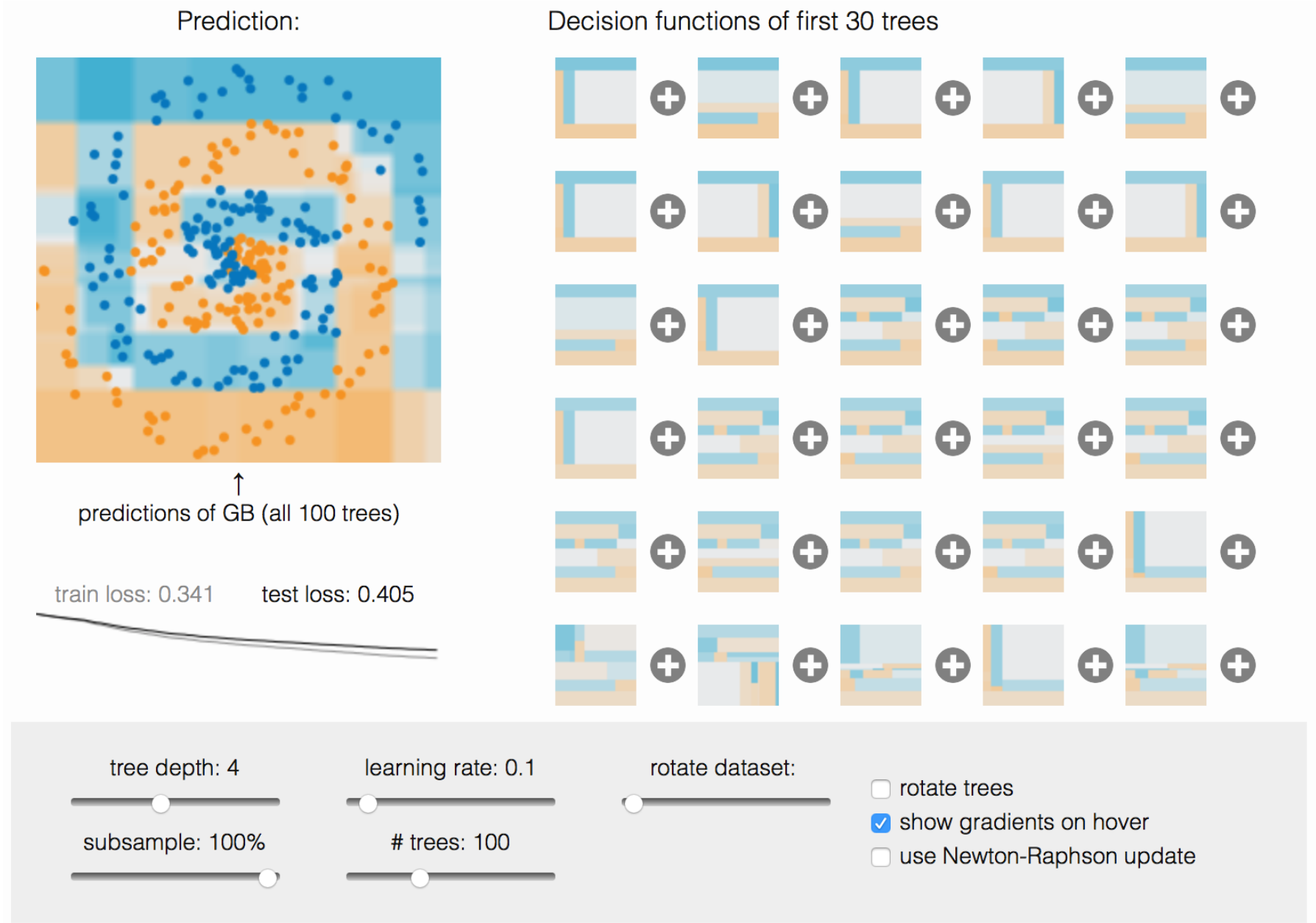
$$\sum_{i=1}^{\ell} (b_N(x_i) - s_i)^2 \to \min_{b_N}$$

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



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

Обучение градиентного бустинга

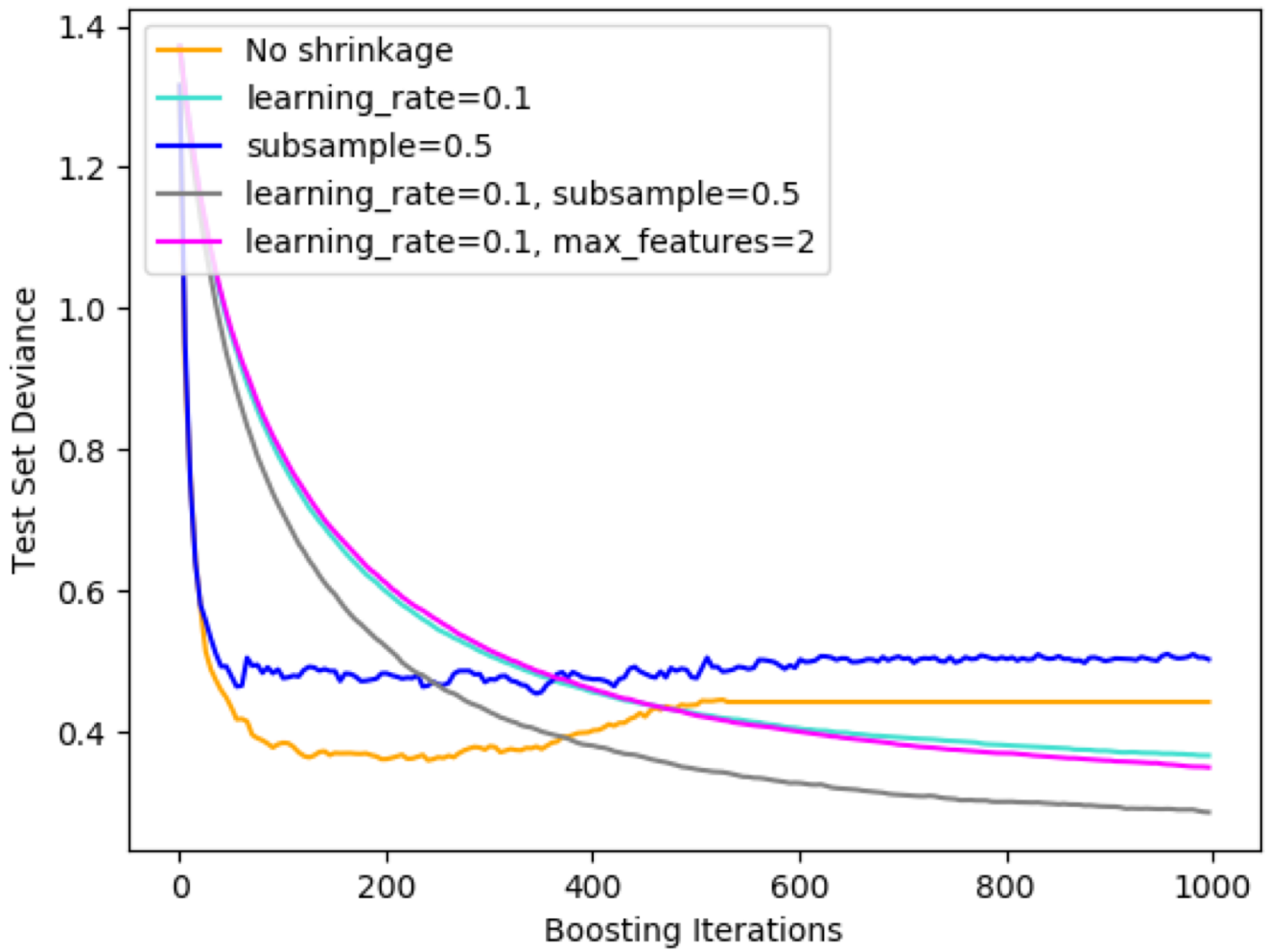
- Основные гиперпараметры:
  - Число деревьев
  - Размер шага
  - Глубина дерева
- В имплементациях могут быть и другие важные настройки
  - Регуляризация
  - Семплирование объектов
  - и т.д.

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


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

Обучение с учителем (заключение)

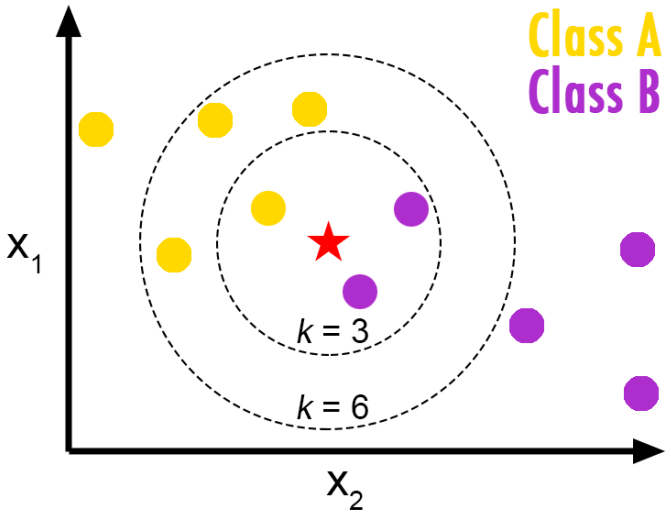
Обучение с учителем

- В нашем курсе: классификация или регрессия
- $X = (x_i, y_i)_{i=1}^{\ell}$ — обучающая выборка
- $a(x)$ — алгоритм, модель

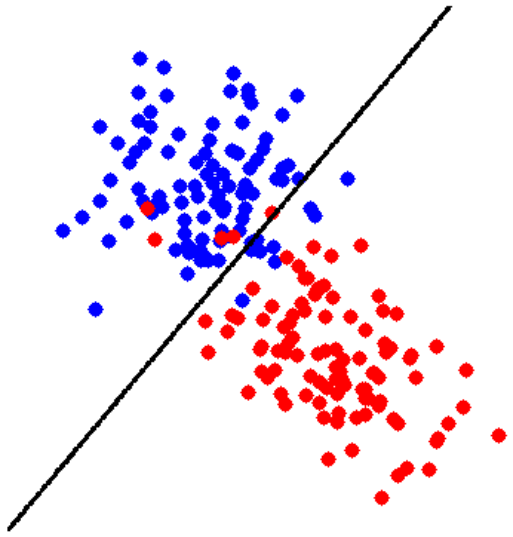
- «С учителем» — т.е. на обучающей выборке известны ответы y_i

Обучение с учителем

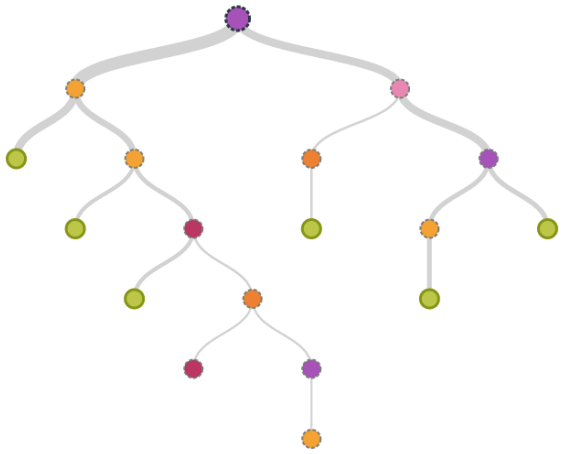
Метод k ближайших соседей



Линейные модели

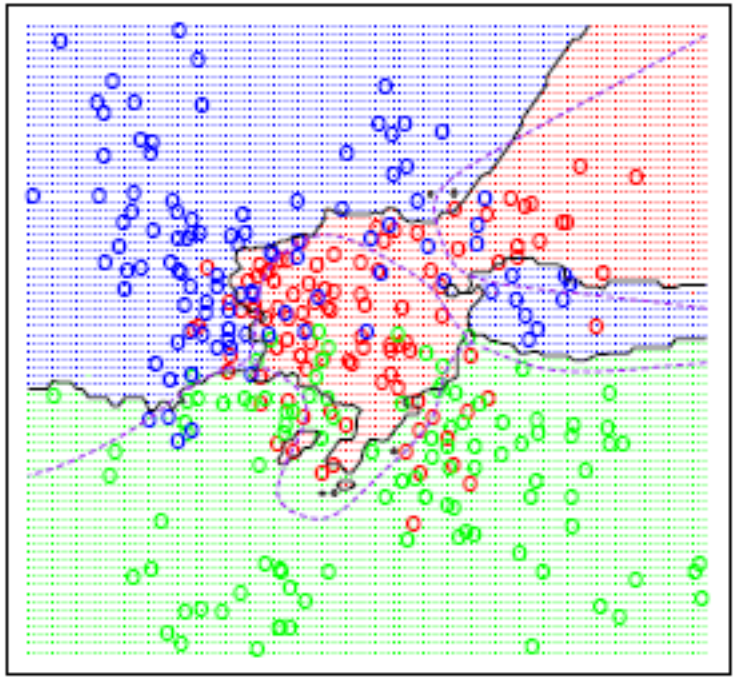


Решающие деревья



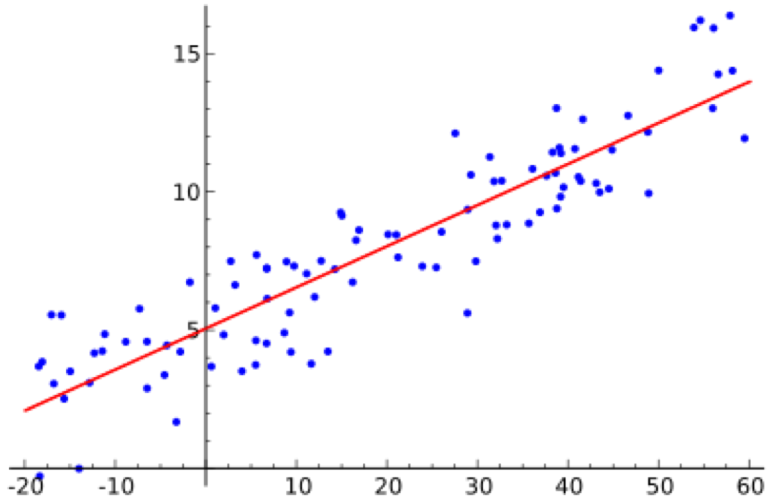
Метод k ближайших соседей

- (+) Очень мало параметров
- (+) Может восстанавливать сложные закономерности
- (-) Нередко показывает плохое качество
- (-) Приличных результатов можно добиться при подборе метрики



Линейные модели

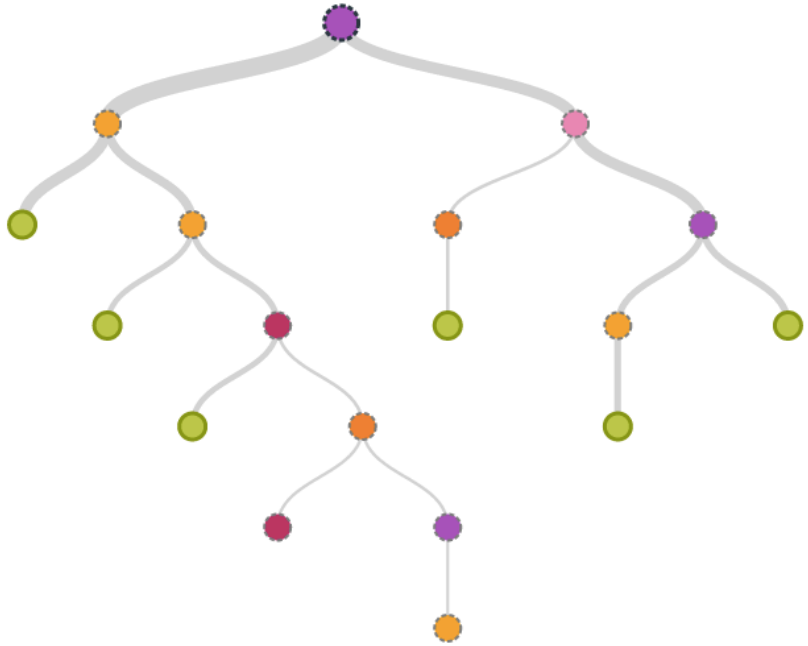
- (+) Легко контролировать переобучение (регуляризация)
- (+) Быстро обучаются даже на огромных объёмах данных
- (+) Хорошо работают при большом числе признаков (например, на категориальных признаках)
- (-) Восстанавливают всего лишь линейные закономерности



Решающие деревья

- (+) Могут дать нулевую ошибку на любой обучающей выборке
- (+) Можно интерпретировать
- (-) Очень легко переобучаются

- (+) Хорошо объединяются в композиции



Обучение с учителем

- Мы изучили основные типы моделей
- Измерять качество в регрессии и классификации тоже научились

- Важные этапы — подготовка данных (откуда их взять?) и разработка признаков
- Пример задачи: автоответ на письма
  - по мотивам статьи «Smart Reply: Automated Response Suggestion for Email», KDD 2016

Автоответ на письма

Сейчас 25% писем-ответов содержат меньше 20 токенов

Требования к автоответу:

- Высокое качество с точки зрения языка и смысла
- Разнообразие
  - Показывать несколько разных вариантов
  - «Yes, I will be there» и «I’ll be there»
- Сохранение приватности переписки пользователей

Задачи машинного обучения

Триггеринг:

- Понять, нужен ли для данного письма автоответ
- Письма со сложным вопросом
  - «Where do you want to go today?»
- Письма, на которые ответ не нужен вообще

Задачи машинного обучения

Выбор наиболее подходящих ответов:

- Классификация на K классов
- K — число допустимых ответов

Задачи не про машинное обучение

- Как собрать обучающую выборку для задачи триггеринга?
- Какие автоответы являются допустимыми?
- На каких признаках обучать классификаторы?
- Как добиться разнообразия ответов?

Триггеринг

Данные:

- Положительные примеры — письма, на которые ответили с мобильного устройства
- Отрицательные примеры — письма, не которые не ответили вообще
- 238 миллионов объектов

Допустимые ответы

Как не надо:

- Your the best!
- Thanks hon
- Yup
- Got it thx
- Leave me alone

Допустимые ответы

Как не надо (все предлагаемые ответы — одинаковые по смыслу):

Yes, I'll be there.
Yes, I will be there.
I'll be there.
Yes, I can.
What time?
I'll be there!
I will be there.
Sure, I'll be there.
Yes, I can be there.
Yes!

Допустимые ответы

- Взяли несколько миллионов наиболее частых ответов пользователей
- Кластеризовали их
- Выбрали из каждого кластера пять представителей и проверили допустимость силами асессоров

Разнообразие ответов

- Из каждого кластера выбирается представитель с максимальной оценкой вероятности от классификатора

- Есть смещение в сторону позитивных ответов
- Если топ-3 кандидатов позитивные, то третий заменяется на наиболее вероятный негативный ответ

Задачи не про машинное обучение

- Как собрать обучающую выборку для задачи триггеринга?
- Какие автоответы являются допустимыми?
- На каких признаках обучать классификаторы?
- Как добиться разнообразия ответов?