

Машинное обучение

Лекция 7
Композиции алгоритмов

Михаил Гущин

mhushchyn@hse.ru

НИУ ВШЭ, 2022



НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

На прошлой лекции

Построение **решающего дерева**:

- ▶ Даны данные X, y
- ▶ Находим **предикат** $\{x_j > t\}$ для вершины дерева:

$$\Delta I_{node} = I_{node} - \left(I_{left} \frac{N_{left}}{N_{node}} + I_{right} \frac{N_{right}}{N_{node}} \right) \rightarrow \max_{j,t}$$

- ▶ Строим две **дочерние вершины**: правую и левую

$$R_r(j, t) = \{x | x_j > t\}$$

$$R_l(j, t) = \{x | x_j \leq t\}$$

- ▶ **Рекурсивно** находим предикаты для всех дочерних вершин
- ▶ Если выполняются **критерии останова**, останавливаемся

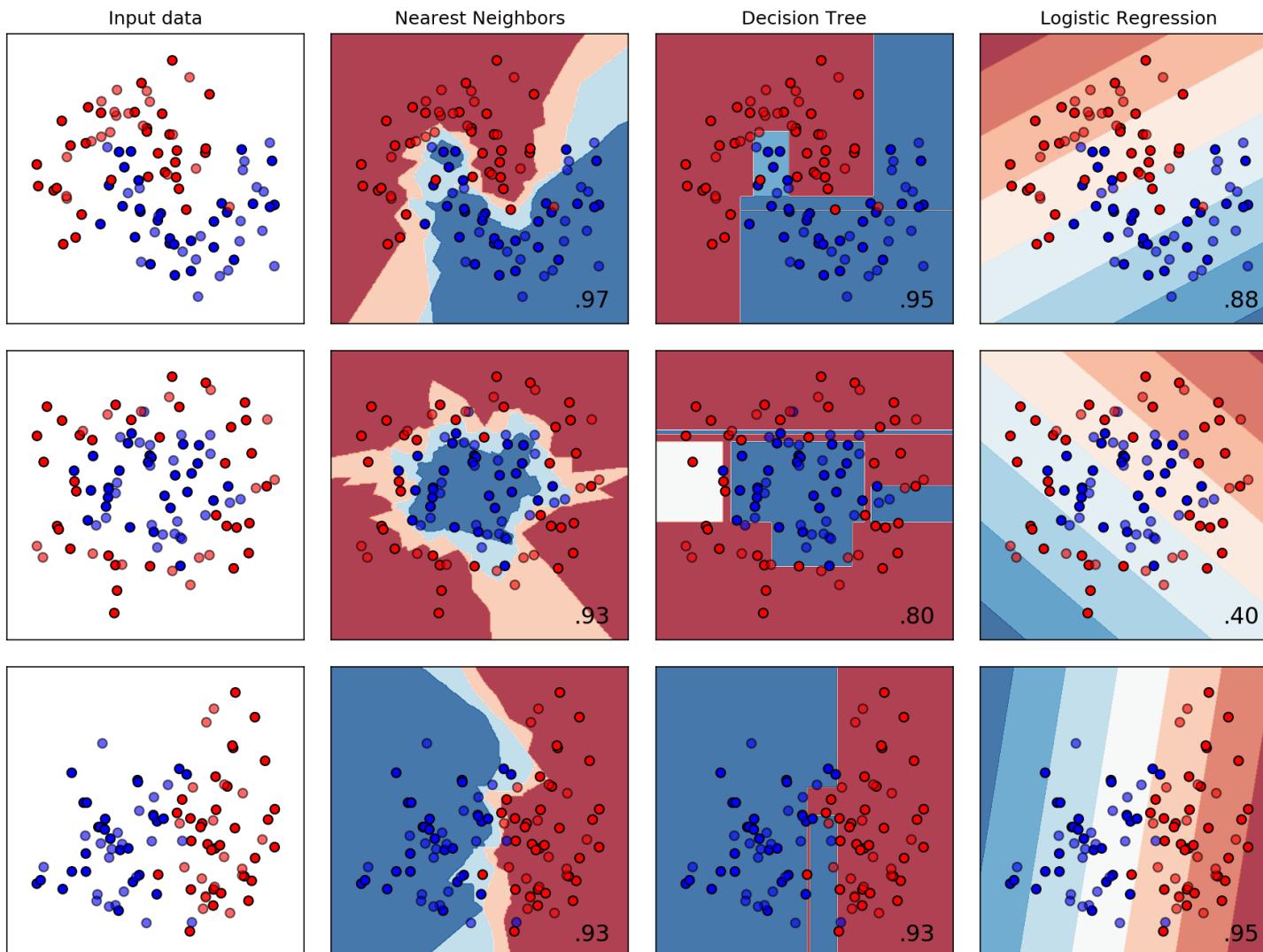
План

- ▶ Постановка задачи
- ▶ Бэггинг
- ▶ Случайный лес решающих деревьев
- ▶ Bias-Variance decomposition
- ▶ Градиентный бустинг

Постановка задачи



Классификаторы



Задача

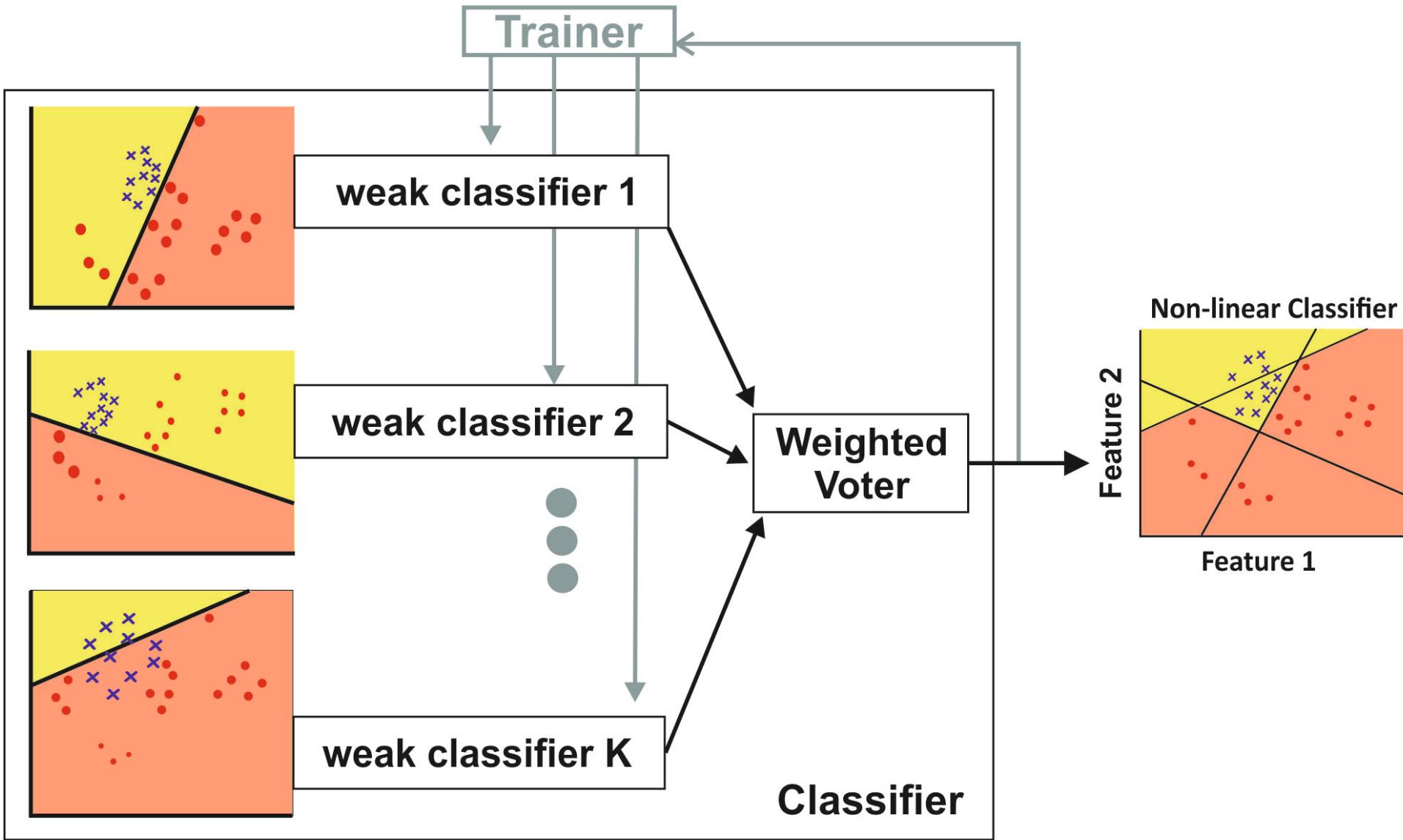
► **Нам дано:**

- Набор данных X, y
- Несколько классификаторов

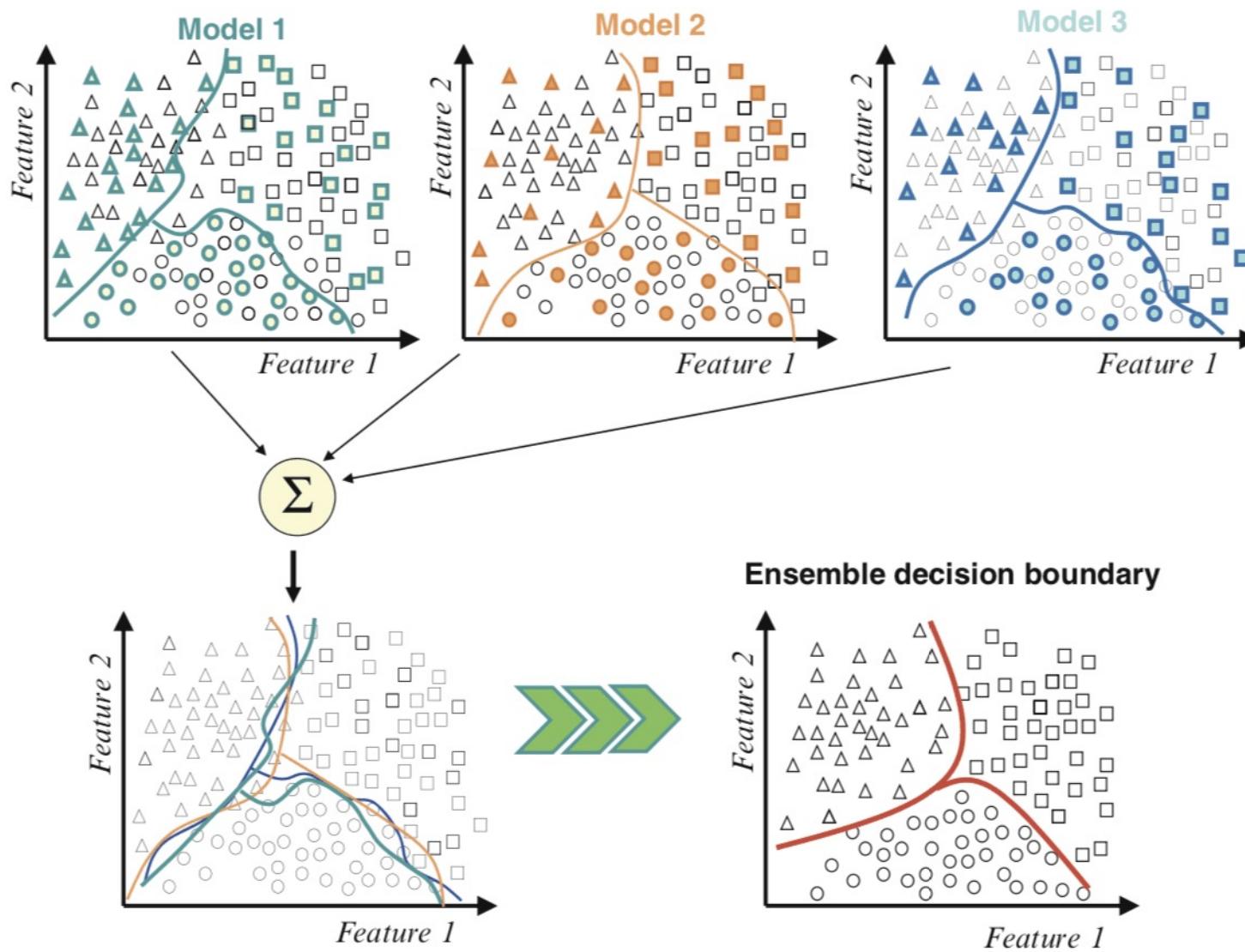
► **Наши задачи:**

- Как улучшить прогнозы классификаторов?
- Можно ли объединить разные классификаторы в одну модель?

Пример



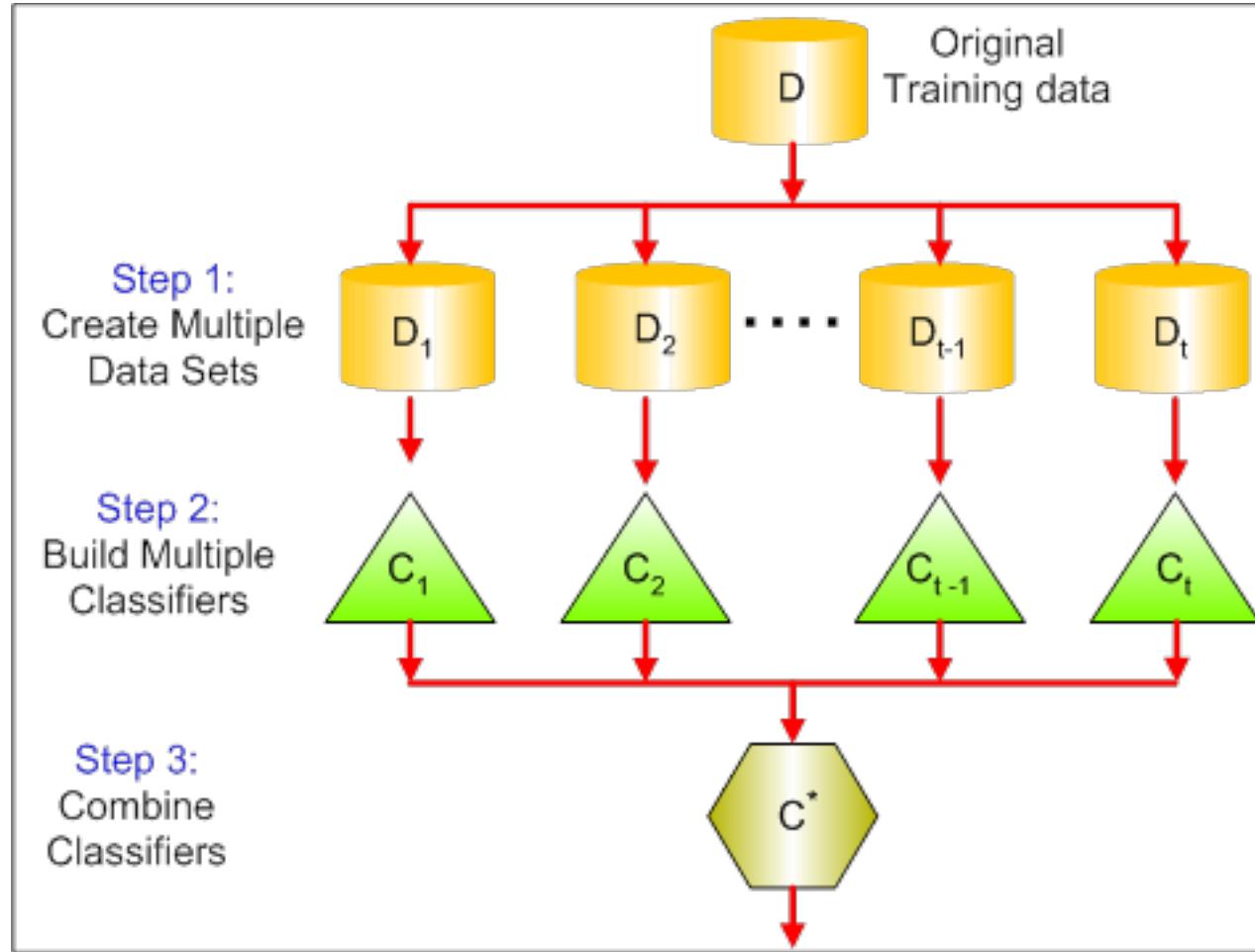
Пример



Бэггинг (bagging)



Бэггинг (bagging)



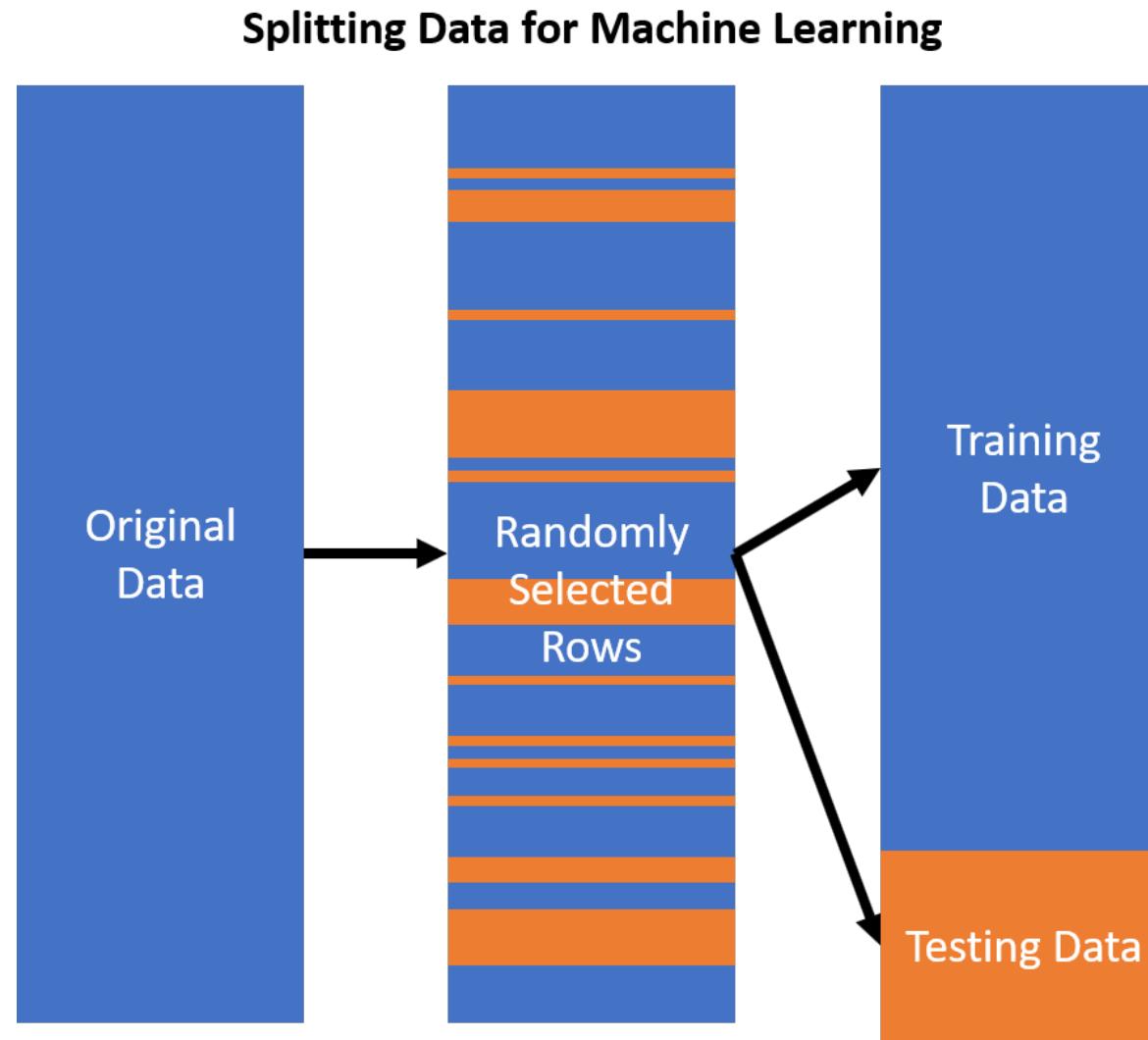
Подвыборки

- ▶ Как создать разные подвыборки имеющихся данных?

Два наиболее популярных способа:

- ▶ Случайная подвыборка без повторений (train-test split)
 - Случайным образом выбираем объекты из всей выборки
 - Размер подвыборки меньше самой выборки
- ▶ Бутстррап
 - Случайная подвыборка с повторениями
 - Размер подвыборки совпадает с размером всей выборки

Случайная подвыборка



Бутстрэп (bootstrap)

Original Dataset

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
-------	-------	-------	-------	-------	-------	-------	-------	-------	----------

Bootstrap 1

x_8	x_6	x_2	x_9	x_5	x_8	x_1	x_4	x_8	x_2
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_3	x_7	x_{10}
-------	-------	----------

Bootstrap 2

x_{10}	x_1	x_3	x_5	x_1	x_7	x_4	x_2	x_1	x_8
----------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_6	x_9
-------	-------

Bootstrap 3

x_6	x_5	x_4	x_1	x_2	x_4	x_2	x_6	x_9	x_2
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_3	x_7	x_8	x_{10}
-------	-------	-------	----------

Training Sets

Test Sets



This work by Sebastian Raschka is licensed under a
Creative Commons Attribution 4.0 International License.

Алгоритм бэггинга

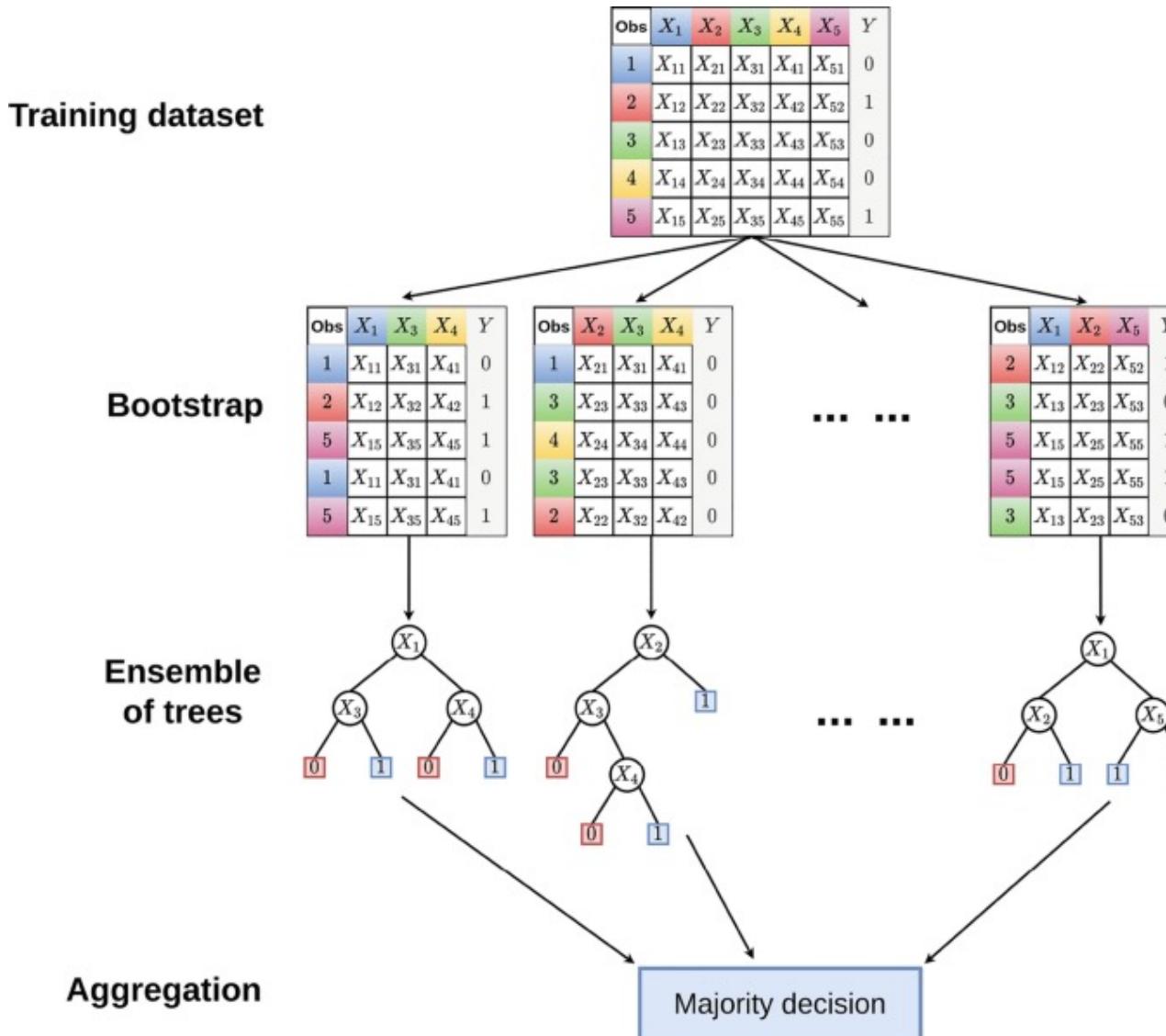
- ▶ Даны выборка данных X, y
- ▶ Для $k = 1 \dots K$:
 - Методом **бутстрата** генерируем подвыборку $X^{(k)}, y^{(k)}$
 - Обучаем модель классификации или регрессии $b_k(x)$ на $X^{(k)}, y^{(k)}$
- ▶ Собираем композицию моделей:

$$\hat{y}(x) = \frac{1}{K} \sum_{k=1}^K b_k(x)$$

Случайный лес решающих
деревьев



Случайный лес (random forest)

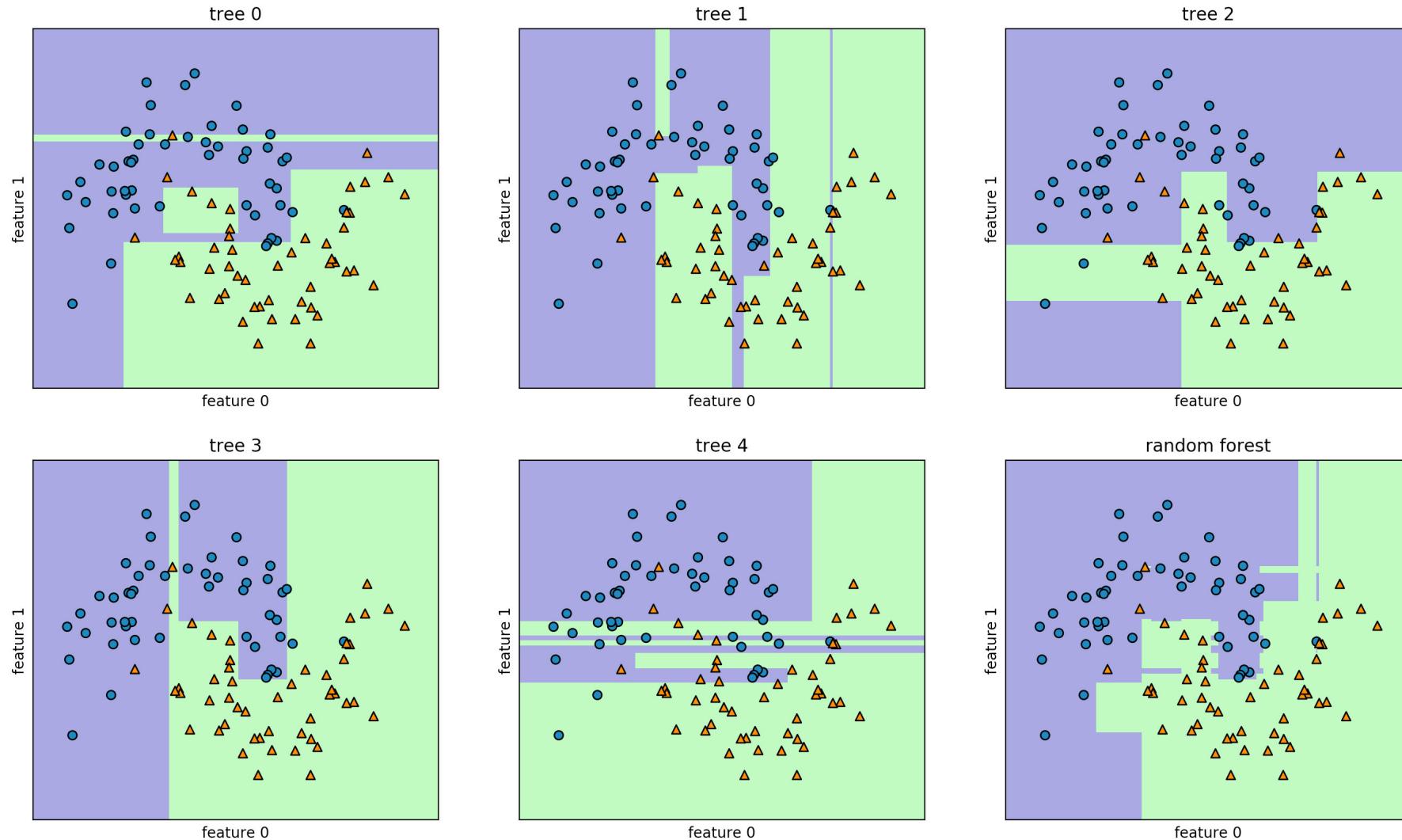


Алгоритм случайного леса

- ▶ Даны выборка данных $X \in R^{(n \times d)}, y^n$
- ▶ Для $k = 1 \dots K$:
 - Методом **бутстрата** генерируем подвыборку $X^{(k)}, y^{(k)}$
 - Обучаем решающее дерево классификации или регрессии $b_k(x)$ на $X^{(k)}, y^{(k)}$
 - При каждом разбиении дерева выбирается ***m* случайных признаков из *d*.**
Оптимальное разбиение ищется только среди них.
- ▶ Собираем композицию моделей:

$$\hat{y}(x) = \frac{1}{K} \sum_{k=1}^K b_k(x)$$

Пример



Бутстрэп (bootstrap)

Original Dataset

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
-------	-------	-------	-------	-------	-------	-------	-------	-------	----------

Bootstrap 1

x_8	x_6	x_2	x_9	x_5	x_8	x_1	x_4	x_8	x_2
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_3	x_7	x_{10}
-------	-------	----------

Bootstrap 2

x_{10}	x_1	x_3	x_5	x_1	x_7	x_4	x_2	x_1	x_8
----------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_6	x_9
-------	-------

Bootstrap 3

x_6	x_5	x_4	x_1	x_2	x_4	x_2	x_6	x_9	x_2
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

x_3	x_7	x_8	x_{10}
-------	-------	-------	----------

Training Sets

Test Sets



This work by Sebastian Raschka is licensed under a
Creative Commons Attribution 4.0 International License.

Out-of-Bag (OOB) ошибка

- ▶ Строим деревья на бутстррап подвыборках
- ▶ Объекты, которые не попали в подвыборку, можно использовать для тестирования дерева
- ▶ Для каждого объекта x_i можно найти деревья, которые были обучены без него, и посчитать по их прогнозам Out-of-Bag ошибку:

$$OOB = \sum L(y_i, \frac{1}{\sum_{k=1}^K [x_i \notin X^{(k)}]} \sum_{k=1}^K [x_i \notin X^{(k)}] b_k(x_i))$$

- где $L(y, z)$ – функция потерь или метрика качества

Важные замечания

- ▶ Бэггинг можно применять для любых алгоритмов
- ▶ Лучше всего он работает на слабых моделях
- ▶ Как правило, в композиции объединяют переобученные модели

Bias-Variance decomposition



Вопросы

- ▶ Почему композиции алгоритмов работают?
- ▶ Почему прогноз композиции лучше, чем прогнозы отдельных моделей?
- ▶ Почему используют переобученные модели?

Задача

- ▶ Рассмотрим задачу регрессии с функцией потерь MSE
- ▶ Пусть ответ $y(x)$ для заданного x – некоторая случайная величина:

$$y(x) = f(x) + \epsilon$$

– где $\epsilon \sim N(0, \sigma^2)$

- ▶ Обозначим прогноз нашей модели $a(x)$
- ▶ Посчитаем мат. ожидание ошибки прогноза для заданного x :

$$\text{Error} = E[(a(x) - y(x))^2]$$

Расчет ошибки

$$Error = E \left[(a(x) - y(x))^2 \right] =$$

$$= E[(a(x) - f(x) - \epsilon + E[a(x)] - E[a(x)])^2] =$$

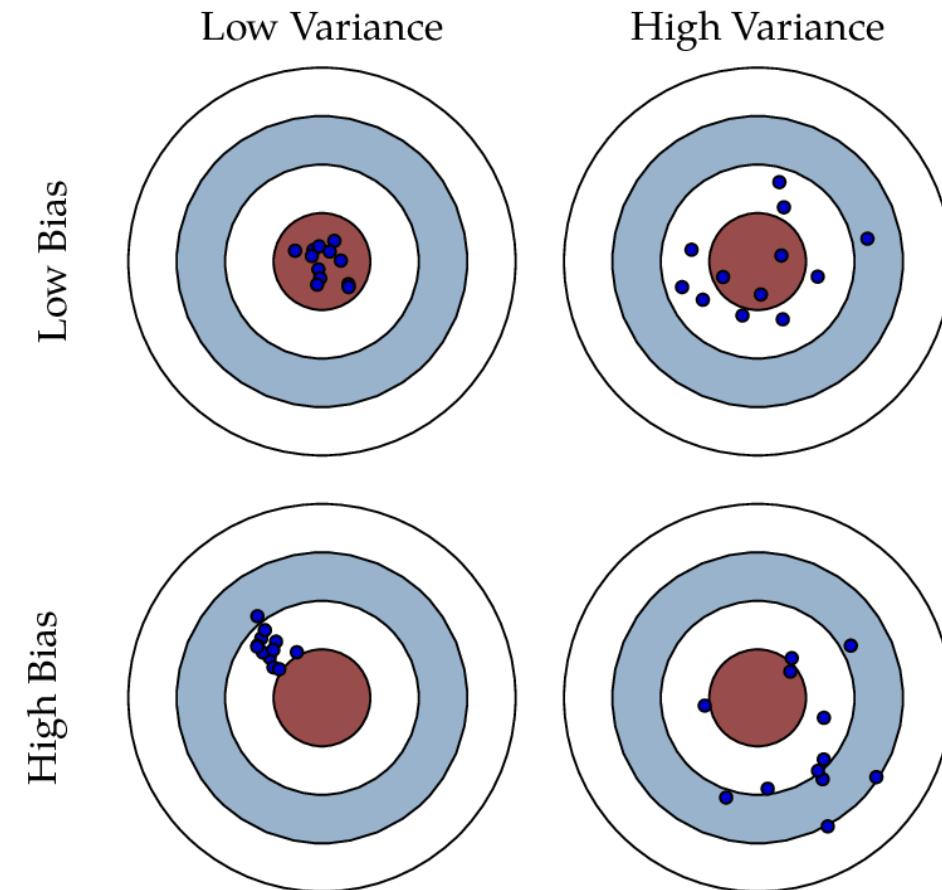
$$= E[(a(x) - E[a(x)])^2] + (E[a(x)] - f(x))^2 + \sigma^2 =$$

$$= \text{Variance} + \text{Bias}^2 + \text{Noise}$$

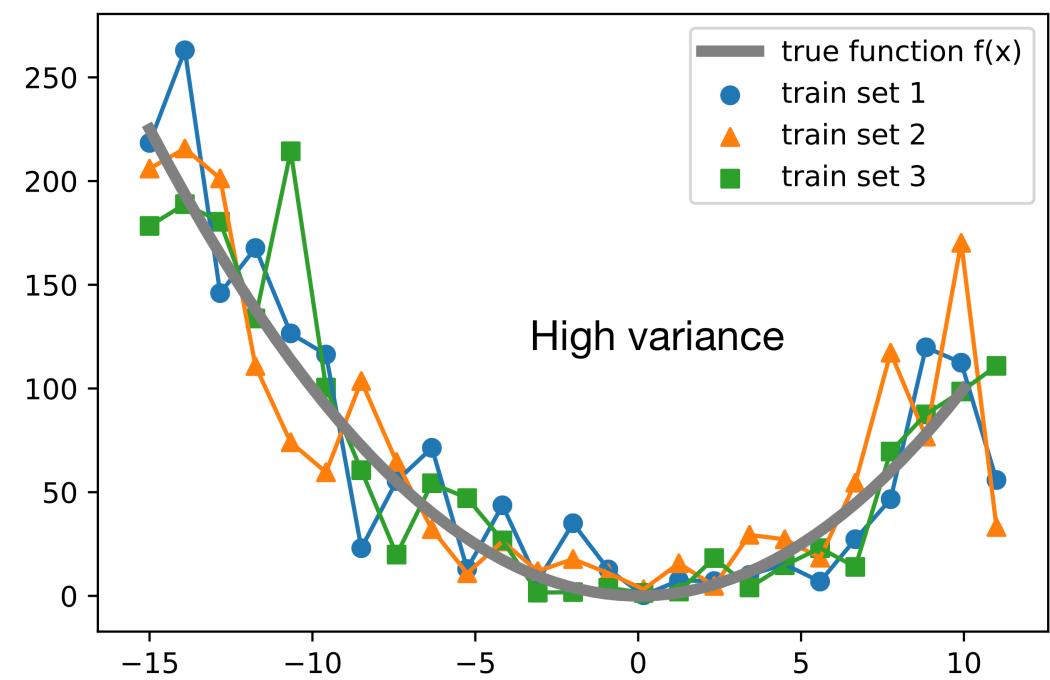
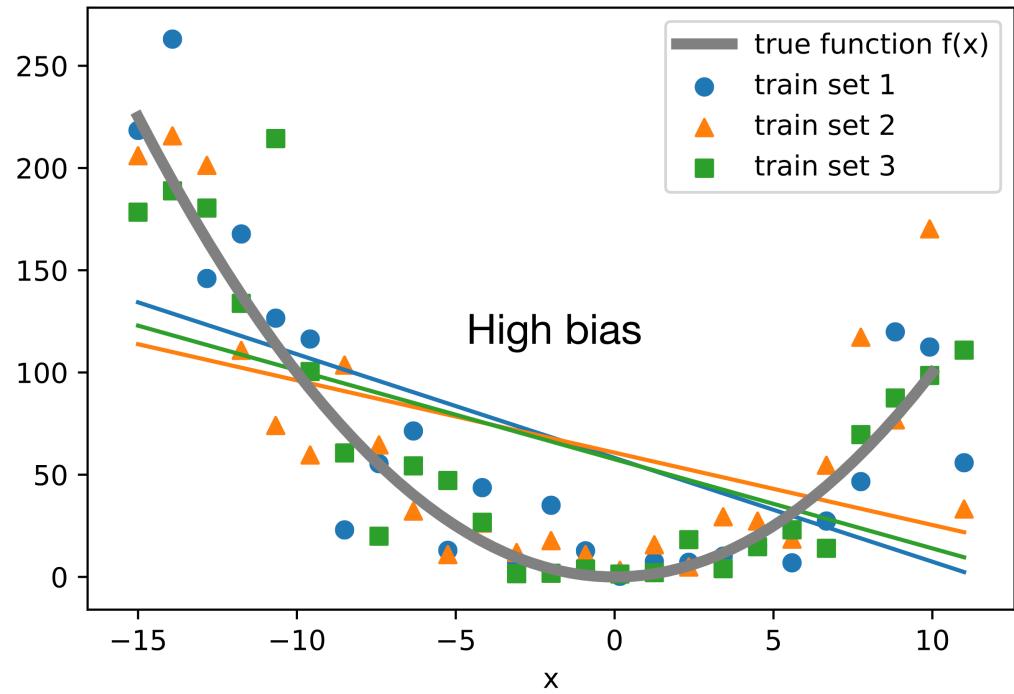
Bias-Variance decomposition

- ▶ **Variance (разброс)** – разброс ответов обученных алгоритмов относительно среднего ответа.
- ▶ **Bias (смещение)** – отклонение среднего ответа алгоритма от идеального ответа
- ▶ **Noise (шум)** – шум в данных

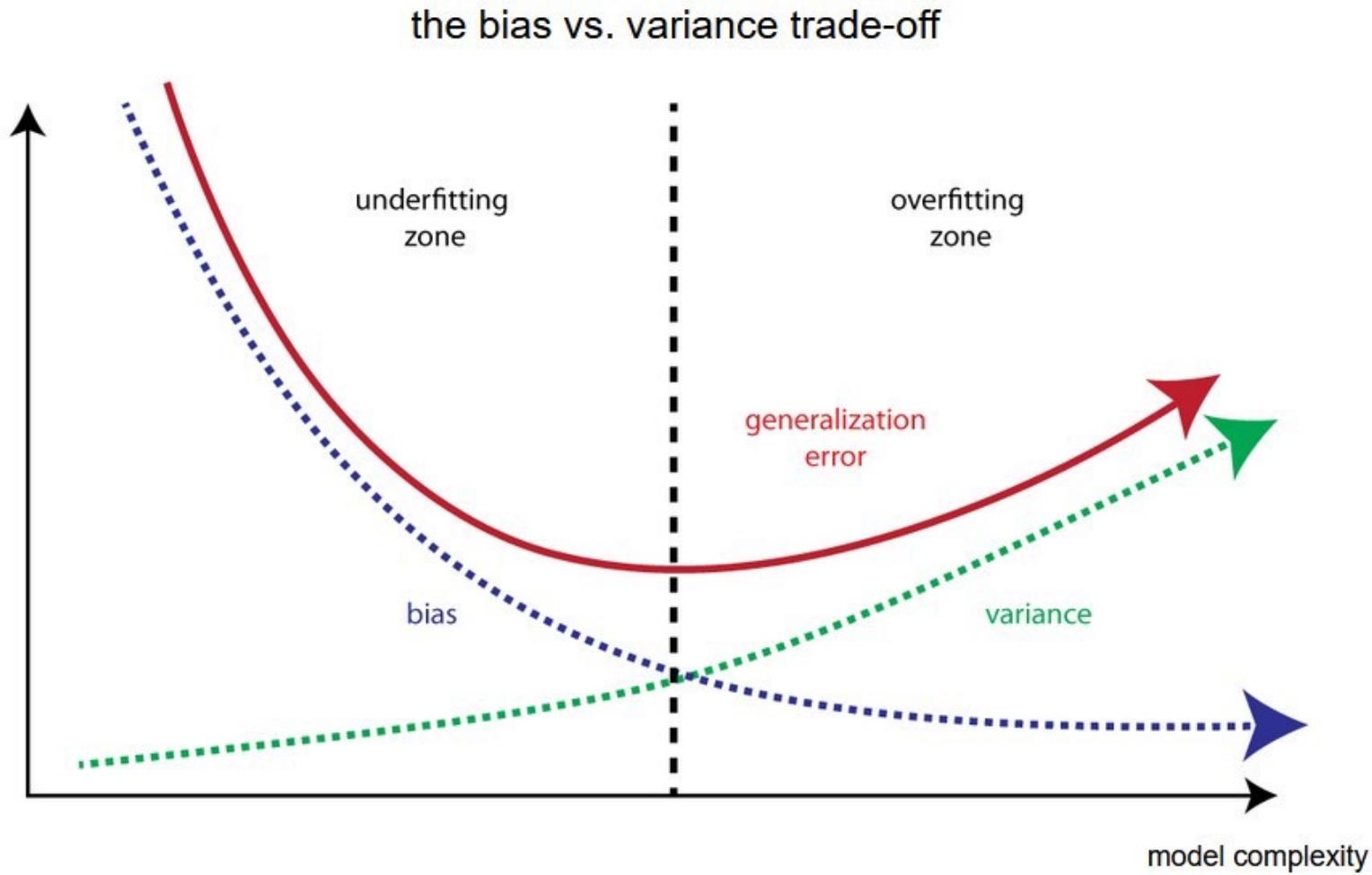
Пример



Пример



Переобучение



Композиции алгоритмов

$$Error = \mathbf{E}[(a(x) - E[a(x)])^2] + (E[a(x)] - f(x))^2 + \sigma^2$$

- ▶ Пусть $a(x)$ - композиция алгоритмов
- ▶ Тогда $a(x) \approx E[a(x)]$
- ▶ Тогда ошибка для композиции:

$$Error \approx \mathbf{0} + (E[a(x)] - f(x))^2 + \sigma^2$$

Градиентный бустинг



Задача

- ▶ Даны выборка данных $X \in R^{(n \times d)}, y^n$
- ▶ Будем строить композицию K моделей:

$$a_K(x) = b_0(x) + \sum_{k=1}^K \gamma_k b_k(x)$$

- где $b_0(x)$ – начальный прогноз. Например, константа.
- ▶ Хотим, чтобы $a_K(x)$ минимизировала нашу функцию потерь:

$$L(y, a_K(X)) \rightarrow \min_{\gamma, b}$$

Вопросы

- ▶ Как минимизировать функцию потерь?
- ▶ Как обучать модели в композиции?

Градиентный спуск

- ▶ Есть функция $L(w)$, минимум которой хотим найти
- ▶ Пусть w_0 - начальный вектор параметров
- ▶ Тогда **градиентный спуск** состоит в повторении:

$$w^{(k)} = w^{(k-1)} - \eta \nabla L(w^{(k-1)})$$

- η – длина шага градиентного спуска (**learning rate**) (мы сами его задаем)
- k – номер итерации
- ▶ Получаем, что **с каждой итерацией к весам добавляется поправка**:

$$\Delta w = w^{(k)} - w^{(k-1)} = -\eta \nabla L(w^{(k-1)})$$

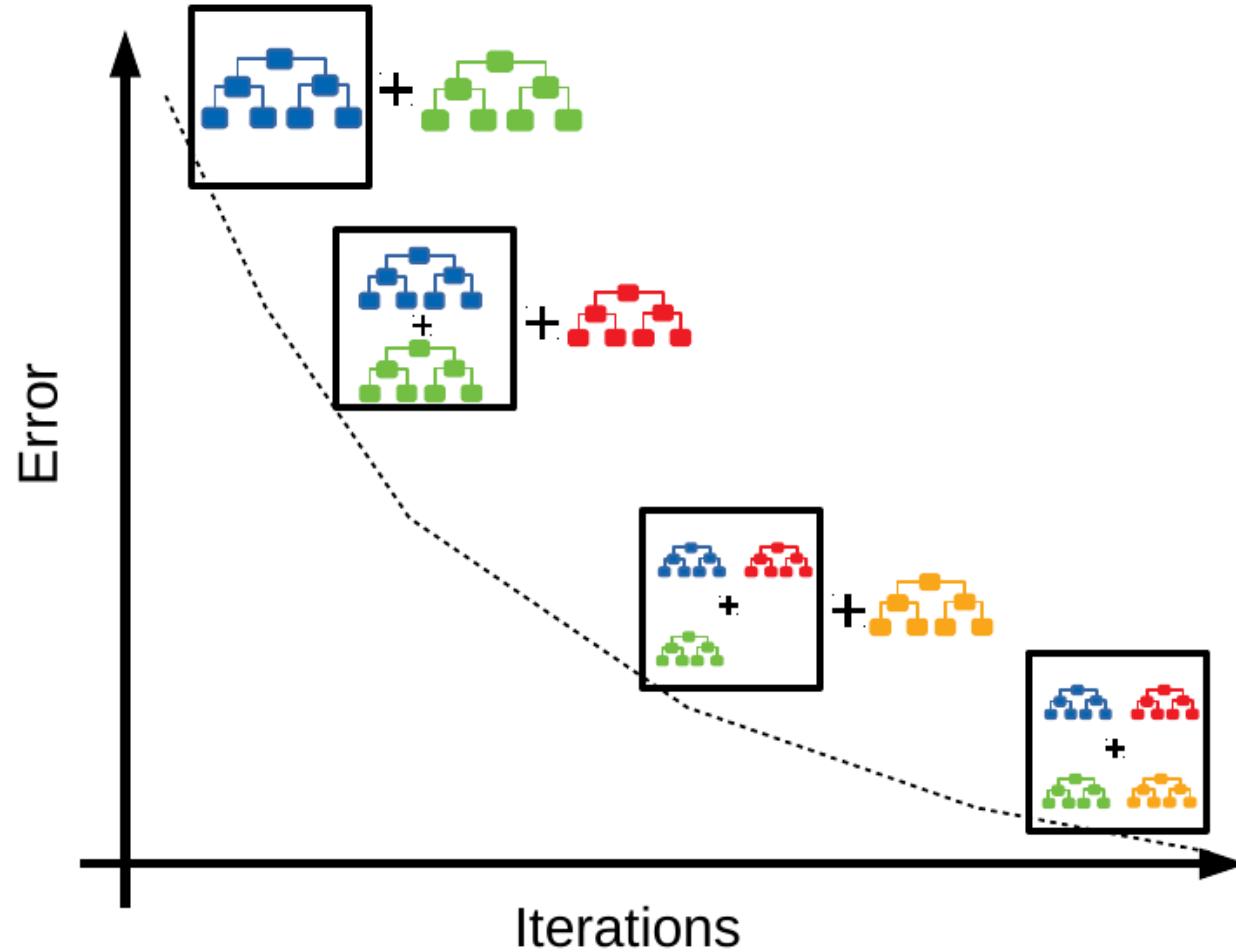
Идея решения

- ▶ Вместо $w^{(k)}$ подставим $a_k(X)$
- ▶ Тогда, для минимизации функции потерь нужно, чтобы

$$a_k(X) - a_{k-1}(X) = -\eta \nabla L(a_{k-1}(X))$$

- ▶ Как обучить новую модель $b_k(X)$, чтобы $a_k(X) = a_{k-1}(X) + \gamma_k b_k(X)$?

Идея решения



Алгоритм градиентного бустинга

- ▶ Даны выборка данных $X \in R^{(n \times d)}, y^n$
- ▶ Делаем начальный прогноз $a_0(X) = b_0(X)$
- ▶ Для каждого $k = 1 \dots K$:
 - Методом **бутстрата** генерируем подвыборку $X^{(k)}, y^{(k)}$
 - Считаем вектор производных функции потерь (**остатки, сдвиги**):

$$s = - \frac{dL(y^{(k)}, z)}{dz} \Big|_{z=a_{k-1}(X^{(k)})}$$

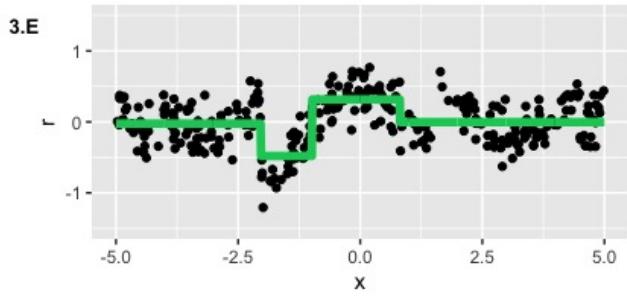
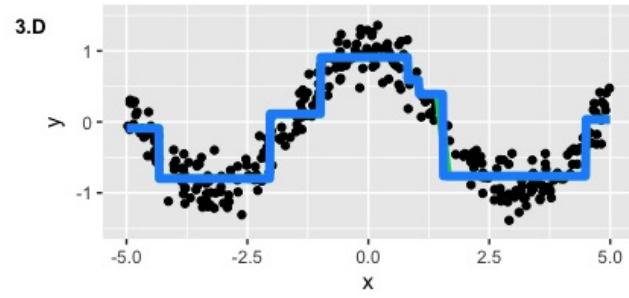
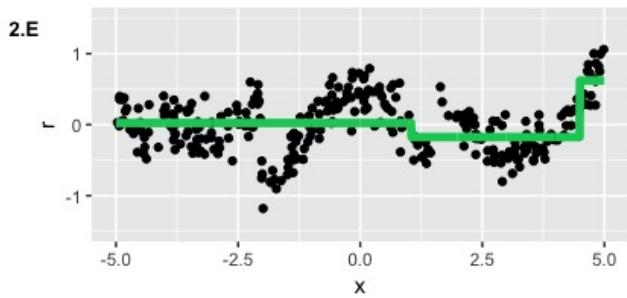
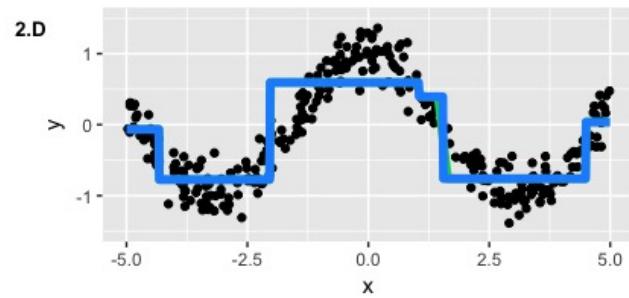
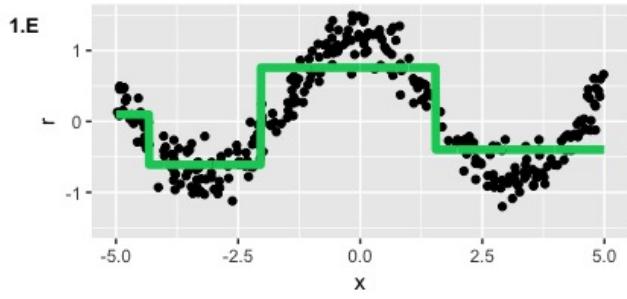
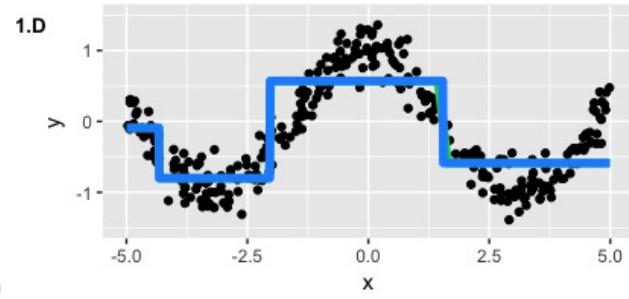
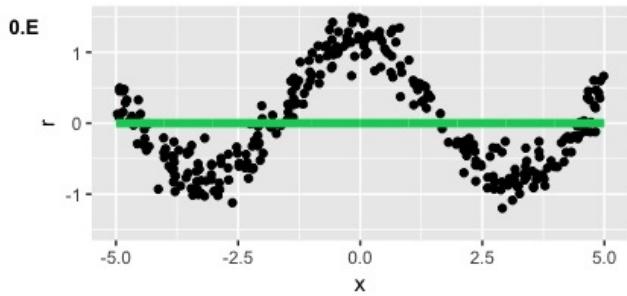
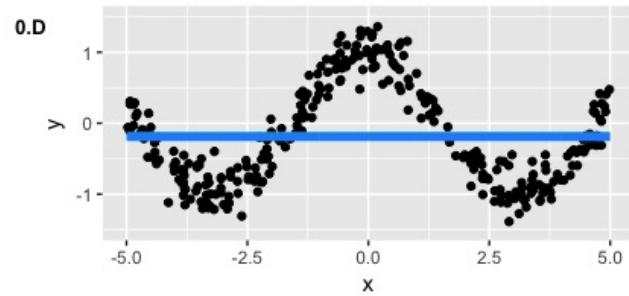
- Обучаем модель **регрессии** $b_k(X^{(k)})$:

$$(b_k(X^{(k)}) - s)^T (b_k(X^{(k)}) - s) \rightarrow \min_{b_k}$$

- Обновляем композицию алгоритмов:

$$a_k(X^{(k)}) = a_{k-1}(X^{(k)}) + \gamma_k b_k(X^{(k)})$$

Пример



$$s = -\frac{dL(y, z)}{dz} \Big|_{z=a_{k-1}(X)}$$

$$a_k(X) = a_{k-1}(X) + \gamma_k b_k(X)$$

Градиентный бустинг с оптимальным шагом

- ▶ Делаем начальный прогноз $a_0(X) = b_0(X)$
- ▶ Для каждого $k = 1 \dots K$:
 - Методом **бутстрата** генерируем подвыборку $X^{(k)}, y^{(k)}$
 - Считаем вектор **остатков (сдвигов)** $s = -\Delta L(y^{(k)}, a_{k-1}(X^{(k)}))$:
 - Обучаем модель **регрессии** $b_k(X^{(k)})$:
 - Обновляем композицию алгоритмов:

$$L \left(y, a_{k-1}(X^{(k)}) + \gamma_k b_k(X^{(k)}) \right) \rightarrow \min_{\gamma_k}$$
$$a_k(X^{(k)}) = a_{k-1}(X^{(k)}) + \eta \gamma_k b_k(X^{(k)})$$

- γ_k - оптимальный шаг градиентного бустинга
- $\eta \in [0, 1)$ – коэффициент сокращения шага (механизм регуляризации)

Сокращение шага

Сокращение шага. На практике оказывается, что градиентный бустинг очень быстро строит композицию, ошибка которой на обучении выходит на асимптоту, после чего начинает настраиваться на шум и переобучаться. Это явление можно объяснить одной из двух причин:

- Если базовые алгоритмы очень простые (например, решающие деревья небольшой глубины), то они плохо приближают вектор антиградиента. По сути, добавление такого базового алгоритма будет соответствовать шагу вдоль направления, сильно отличающегося от направления наискорейшего убывания. Соответственно, градиентный бустинг может свестись к случайному блужданию в пространстве.
- Если базовые алгоритмы сложные (например, глубокие решающие деревья), то они способны за несколько шагов бустинга идеально подогнаться подирующую выборку — что, очевидно, будет являться переобучением, связанным с излишней сложностью семейства алгоритмов.

Источник: <https://github.com/esokolov/ml-course-hse/blob/master/2018-fall/lecture-notes/lecture09-ensembles.pdf>

Важные замечания

- ▶ Неважно какую задачу вы решаете, классификацию или регрессию, в градиентном бустинге **всегда** используются модели **регрессии**
- ▶ Обычно в градиентном бустинге используют решающие деревья
- ▶ Популярные библиотеки:
 - sklearn
 - XGBoost
 - CatBoost
 - LightGBM

Заключение



Резюме

- ▶ Даны выборка данных X, y
- ▶ Для $k = 1 \dots K$:
 - Методом **бутстрата** генерируем подвыборку $X^{(k)}, y^{(k)}$
 - Обучаем модель классификации или регрессии $b_k(x)$ на $X^{(k)}, y^{(k)}$
- ▶ Собираем композицию моделей:

$$\hat{y}(x) = \frac{1}{K} \sum_{k=1}^K b_k(x)$$

Вопросы

- ▶ Что такое композиция алгоритмов машинного обучения? Покажите, что в предположении некоррелированных ошибок базовых алгоритмов, ошибка композиции будет в N раз меньше, чем средняя ошибка базовых алгоритмов, где N - число базовых алгоритмов.
- ▶ Что такое бэггинг? Что такое случайный лес? Что такое out-of-bag ошибка, для чего она используется?
- ▶ Опишите алгоритм построения композиции методом градиентного бустинга. Что такое сдвиги (остатки)?
- ▶ Что такое сокращение шага в градиентном спуске и для чего оно используется?