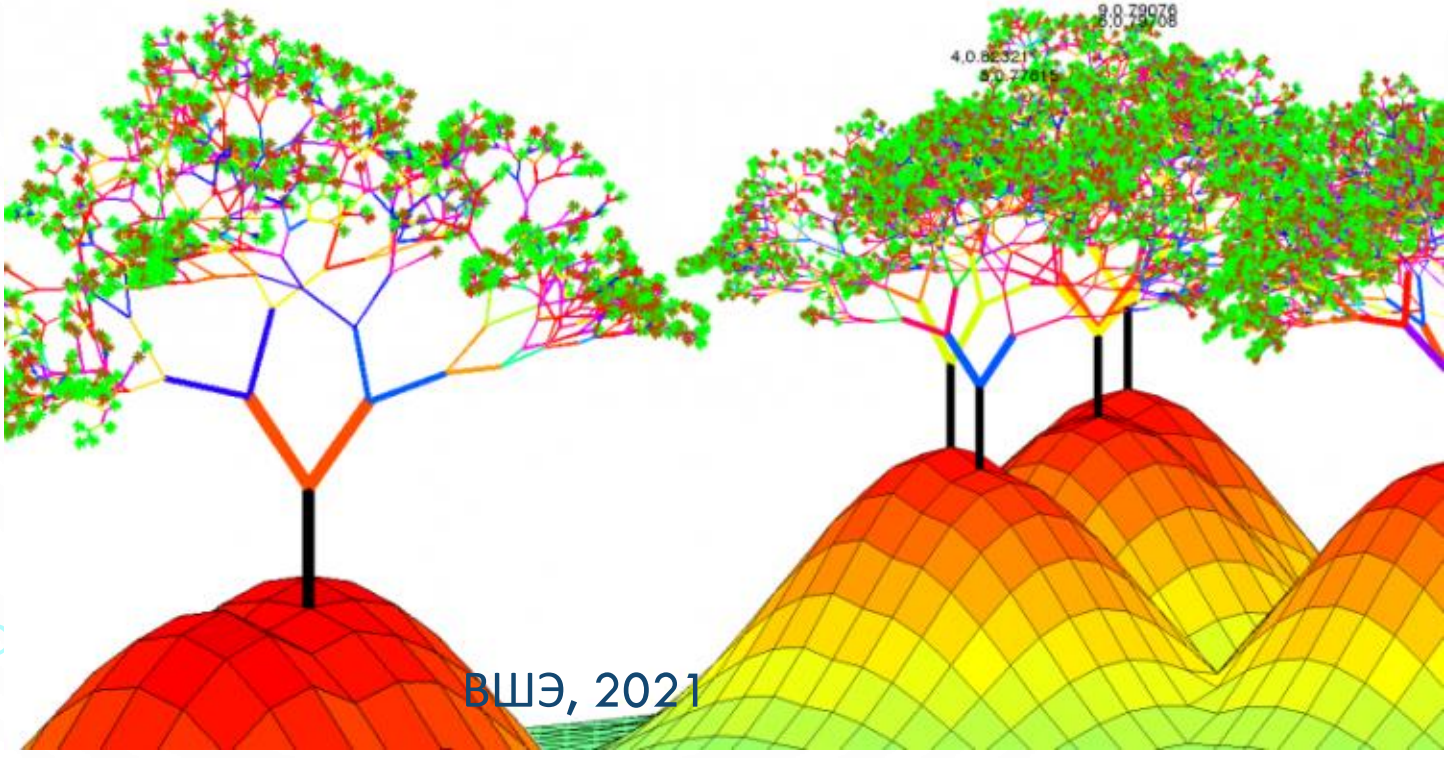
Занятие 9
Ансамбли моделей.

Елена Кантонистова

elena.kantonistova@yandex.ru

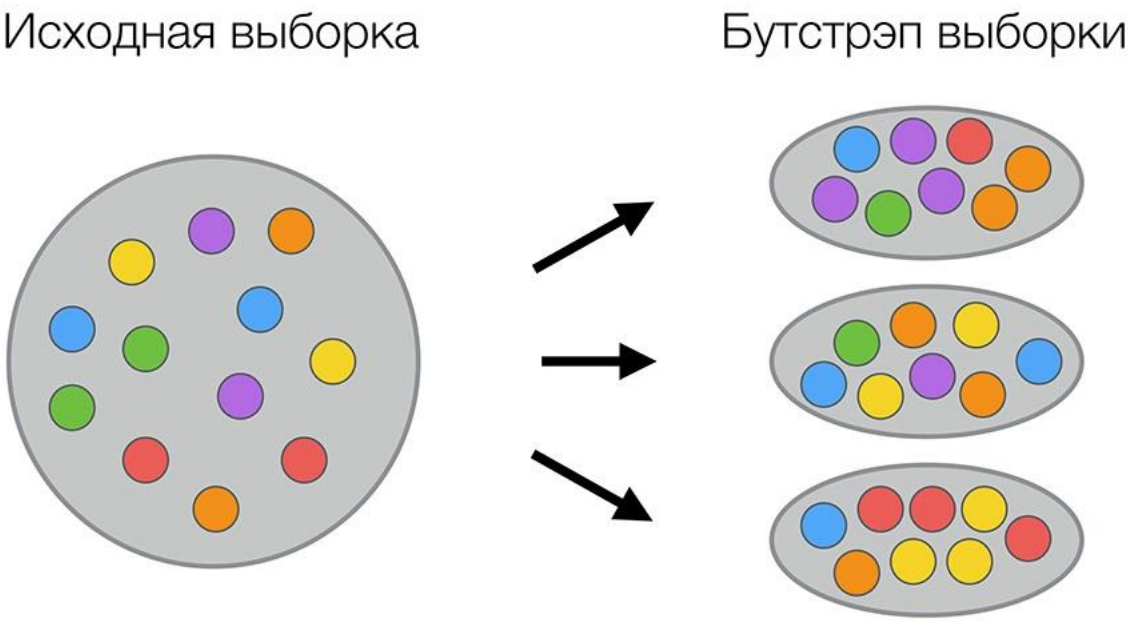


ВШЭ, 2021

БУТСТРЭП

Дана выборка X. Решаем задачу регрессии.

- **Бутстрэп:** равномерно возьмем из выборки X l объектов с возвращением (т.е. в новой выборке будут повторяющиеся объекты). Получим выборку X_1.
- Повторяем процедуру N раз, получаем выборки $X_1, \dots, X_N$.



БЭГГИНГ (BOOTSTRAP AGGREGATION)

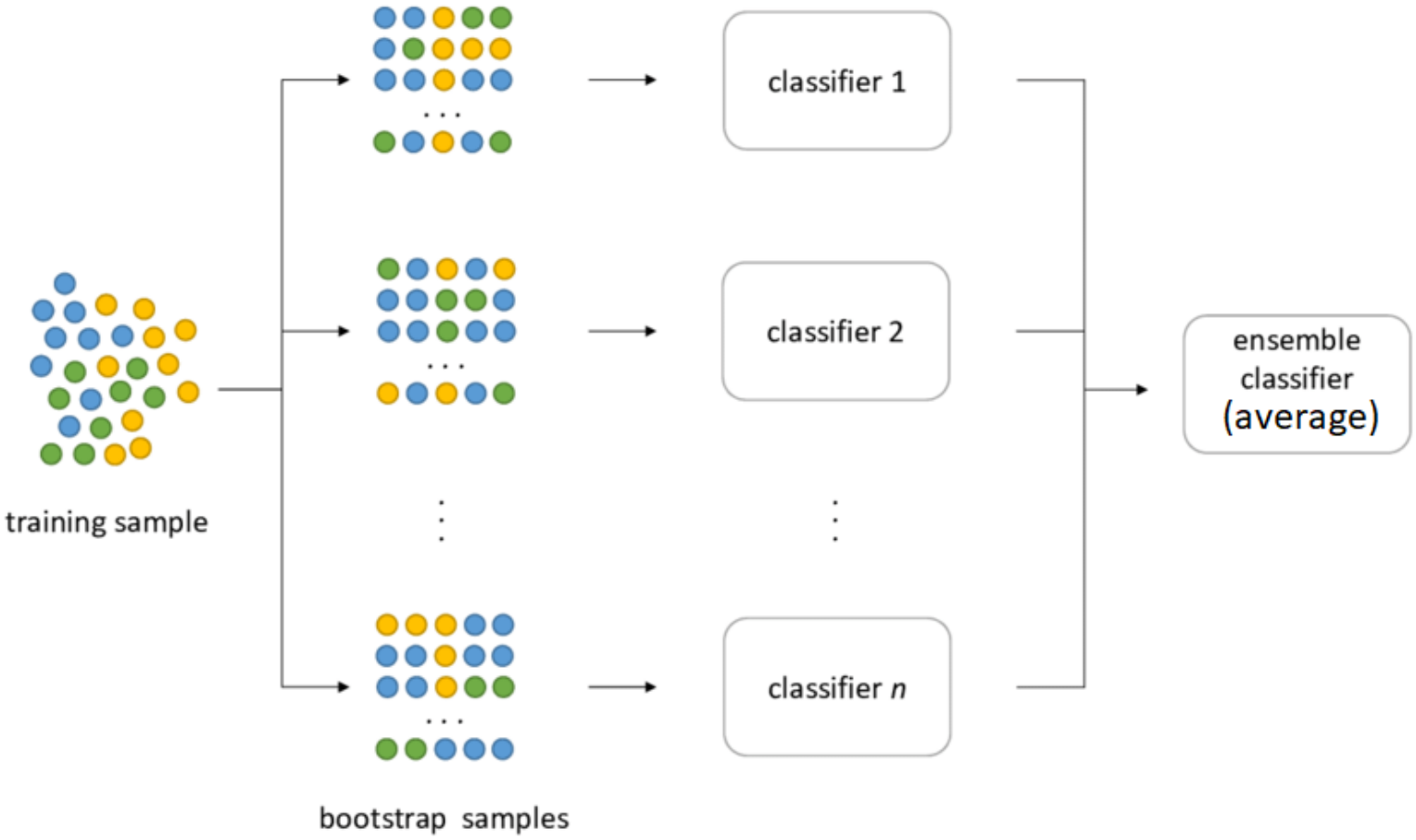
С помощью бутстрэпа мы получили выборки $X_1, \dots, X_N$.

- Обучим по каждой из них линейную модель регрессии – получим базовые алгоритмы $b_1(x), \dots, b_N(x)$.
- Построим новую функцию регрессии:

$$a(x) = \frac{1}{N}\sum_{j=1}^{N} b_j(x)$$

БЭГГИНГ (BOOTSTRAP AGGREGATION)

$$a(x) = \frac{1}{N}\sum_{j=1}^{N} b_j(x)$$



БЭГГИНГ (BOOTSTRAP AGGREGATION)

$$\boldsymbol{a(x) = \frac{1}{N}\sum_{j=1}^{N} b_j(x)}$$

Утверждение. *Если алгоритмы $b_1(x), \ldots, b_N(x)$ некоррелированы, то среднеквадратичная ошибка алгоритма $a(x)$, полученного при помощи бэггинга, в N раз меньше среднеквадратичной ошибки исходных алгоритмов $b_j(x)$.*

РАЗЛОЖЕНИЕ ОШИБКИ (BIAS-VARIANCE DECOMPOSITION

Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

- *Модель переобучена?*
- *Модель плохо предсказывает целевую переменную?*
- *В самих данных много неточностей (шумов)*

РАЗЛОЖЕНИЕ ОШИБКИ (BIAS-VARIANCE DECOMPOSITION)

Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

Утверждение: *ошибку модели $a(x)$ можно представить в виде*

$$\mathbf{Err(x) = Bias^2\big(a(x)\big) + Var\big(a(x)\big) + \sigma^2.}$$

РАЗЛОЖЕНИЕ ОШИБКИ (BIAS-VARIANCE DECOMPOSITION)

Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

Утверждение: *ошибку модели* $a(x)$ *можно представить в виде*

$$\mathbf{Err(x) = Bias^2\big(a(x)\big) + Var\big(a(x)\big) + \sigma^2.}$$

- $\mathbf{Bias\big(a(x)\big)}$ - средняя ошибка по всем возможным наборам данных – **смещение**.

РАЗЛОЖЕНИЕ ОШИБКИ (BIAS-VARIANCE DECOMPOSITION)

Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

Утверждение: *ошибку модели $a(x)$ можно представить в виде*

$$\mathbf{Err(x) = Bias^2\big(a(x)\big) + Var\big(a(x)\big) + \sigma^2.}$$

- $\mathbf{Bias\big(a(x)\big)}$ - средняя ошибка по всем возможным наборам данных – **смещение**.

Смещение показывает, насколько в среднем модель хорошо предсказывает целевую переменную:

- ✓ ***маленькое смещение - хорошее предсказание***
- ✓ ***большое смещение – плохое предсказание***

РАЗЛОЖЕНИЕ ОШИБКИ (BIAS-VARIANCE DECOMPOSITION)

Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

Утверждение: *ошибку модели $a(x)$ можно представить в виде*

$$\mathbf{Err(x) = Bias^2\big(a(x)\big) + Var\big(a(x)\big) + \sigma^2.}$$

- $\mathbf{Var\big(a(x)\big)}$ - дисперсия ошибки, т.е. как сильно различается ошибка при обучении на различных наборах данных – **разброс**.

РАЗЛОЖЕНИЕ ОШИБКИ (BIAS-VARIANCE DECOMPOSITION)

Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

Утверждение: *ошибку модели $a(x)$ можно представить в виде*

$$\mathbf{Err(x) = Bias^2\big(a(x)\big) + Var\big(a(x)\big) + \sigma^2.}$$

- $\mathbf{Var\big(a(x)\big)}$ - дисперсия ошибки, т.е. как сильно различается ошибка при обучении на различных наборах данных – **разброс**.

Большой разброс означает, что ошибка очень чувствительна к изменению обучающей выборки, т.е.:

✓ ***большой разброс – сильно переобученная модель***

РАЗЛОЖЕНИЕ ОШИБКИ (BIAS-VARIANCE DECOMPOSITION)

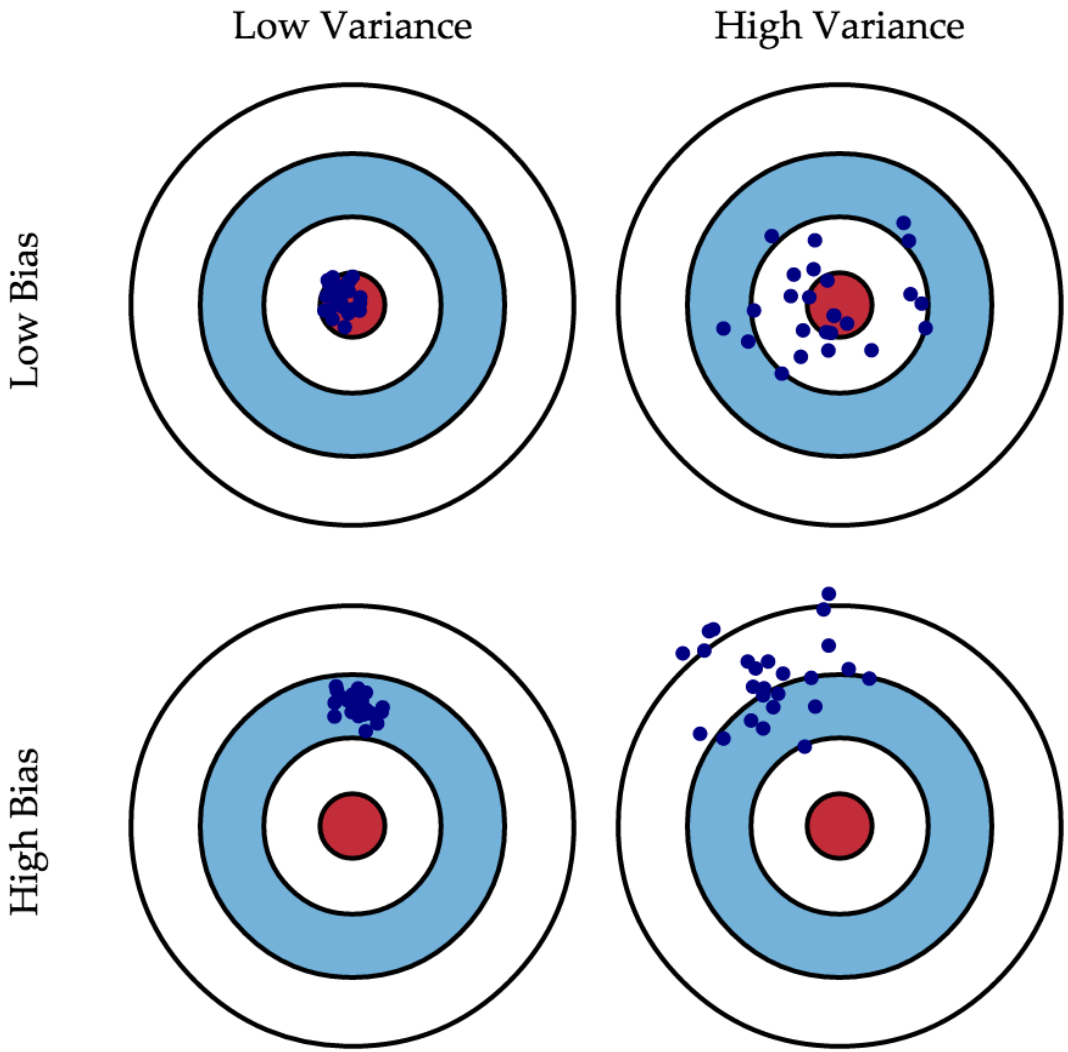
Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

Утверждение: *ошибку модели $a(x)$ можно представить в виде*

$$\mathbf{Err(x) = Bias^2\big(a(x)\big) + Var\big(a(x)\big) + \sigma^2.}$$

- $\mathbf{Bias\big(a(x)\big)}$ - средняя ошибка по всем возможным наборам данных – **смещение**.
- $\mathbf{Var\big(a(x)\big)}$ - дисперсия ошибки, т.е. как сильно различается ошибка при обучении на различных наборах данных – **разброс**.
- $\boldsymbol{\sigma^2}$ - неустранимая ошибка – **шум**.

СМЕЩЕНИЕ И РАЗБРОС

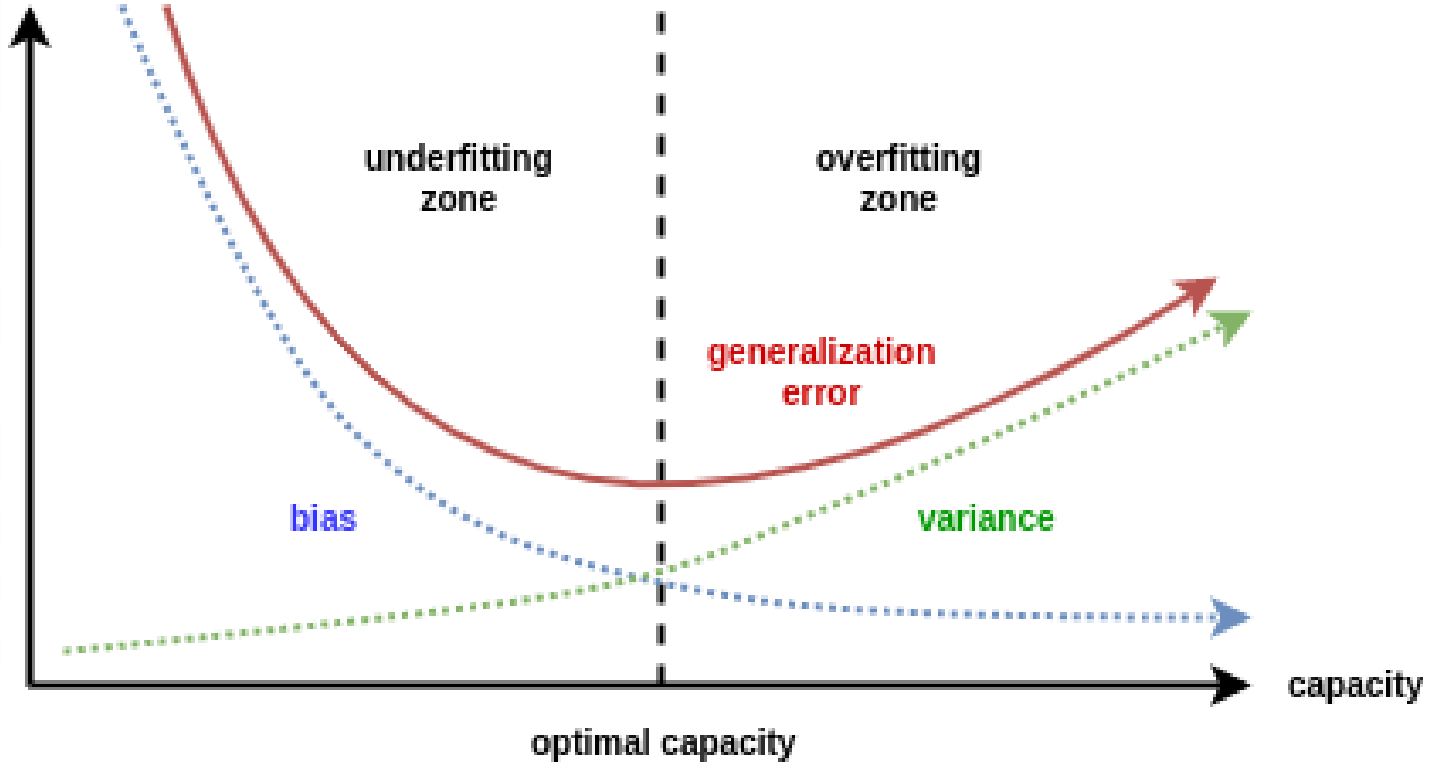


BIAS-VARIANCE TRADE OFF

- У простой модели (например, линейная регрессия) обычно большое смещение и маленький разброс
- Чем сложнее модель (чем больше у неё настраиваемых параметров), тем меньше у неё смещение и тем больше разброс

BIAS-VARIANCE TRADE OFF

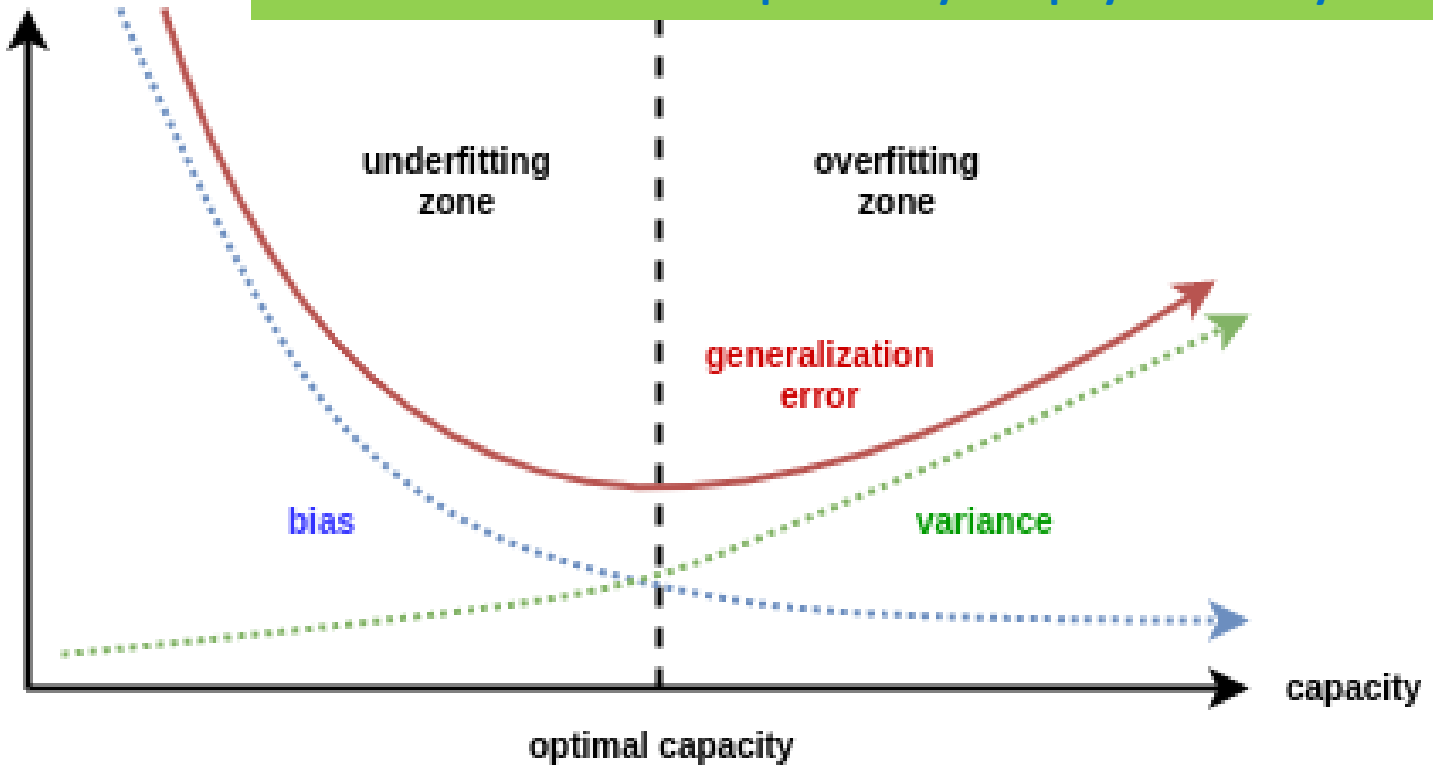
- У простой модели (например, линейная регрессия) обычно большое смещение и маленький разброс
- Чем сложнее модель (чем больше у неё настраиваемых параметров), тем меньше у неё смещение и тем больше разброс



BIAS-VARIANCE TRADE OFF

- У простой модели (например, линейная регрессия) обычно большое смещение и маленький разброс
- Чем сложнее модель (чем больше у неё настраиваемых параметров), тем меньше у неё смещение и тем больше разброс

Цель: подобрать оптимальную по сложности модель, чтобы минимизировать суммарную ошибку.



СМЕЩЕНИЕ И РАЗБРОС У БЭГГИНГА

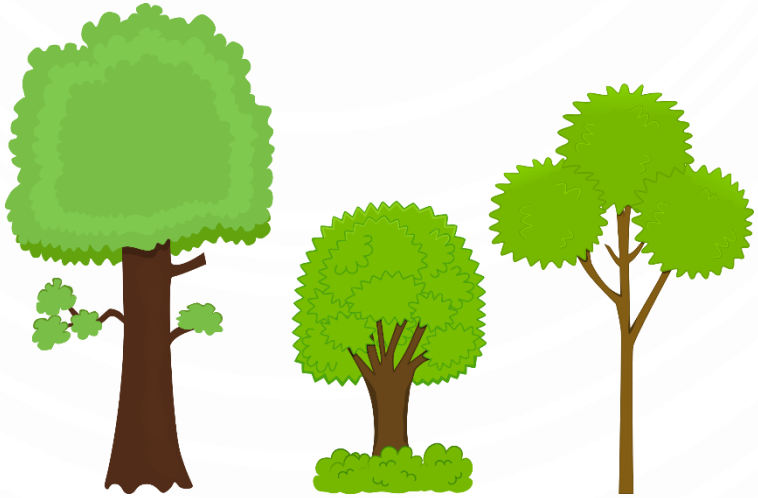
Утверждение.

1) ***Бэггинг не ухудшает смещенность модели****, т.е. смещение* $a_N(x)$ *равно смещению одного базового алгоритма.*

2) Если базовые алгоритмы некоррелированы, то ***дисперсия бэггинга*** $a_N(x)$ ***в N раз меньше дисперсии отдельных базовых алгоритмов.***

СЛУЧАЙНЫЙ ЛЕС (RANDOM FOREST)

- Возьмем в качестве базовых алгоритмов для бэггинга ***решающие деревья***, т.е. каждое случайное дерево $b_i(x)$ построено по своей подвыборке X_i.
- В каждой вершине дерева будем искать ***разбиение не по всем признакам, а по подмножеству признаков***.
- Дерево строится до тех пор, пока в листе не окажется n_{min} объектов.

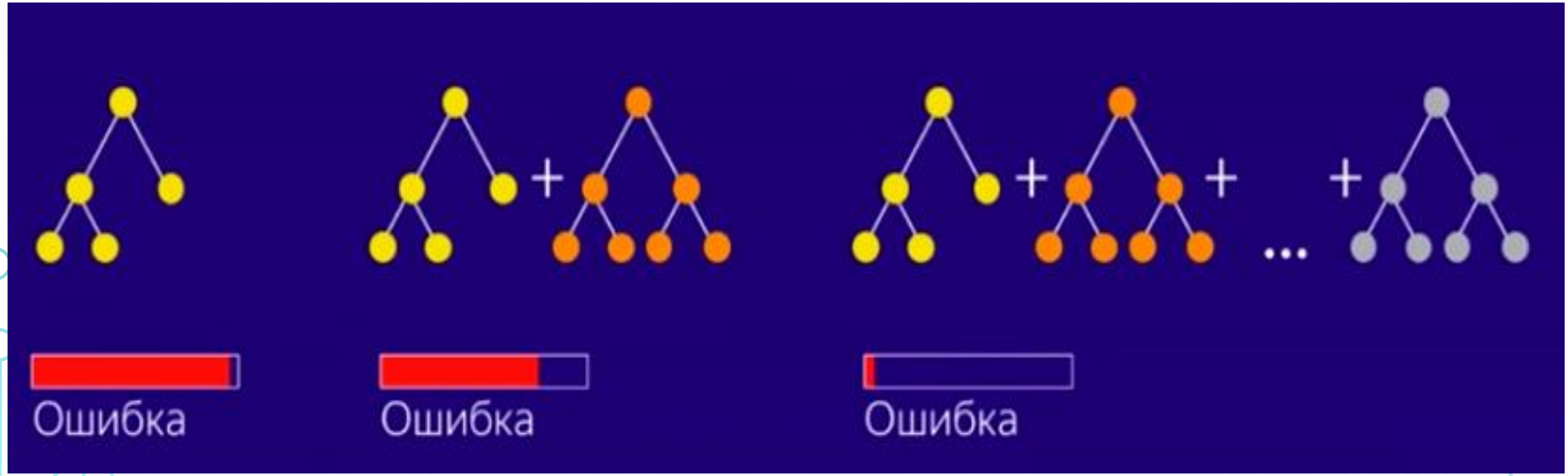


БУСТИНГ

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

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 объекте x:

$$s = y - b_1(x)$$

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 объекте x:

$$s = y - b_1(x)$$

Следующий алгоритм должен настраиваться на эту ошибку, т.е. ***целевая переменная для следующего алгоритма – это вектор ошибок s*** (а не исходный вектор y)

БУСТИНГ В ЗАДАЧЕ РЕГРЕССИИ

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

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

Шаг 2: Ищем алгоритм $b_2(x)$, настраивающийся на ошибки s 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

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

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

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

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

Следующий алгоритм $b_3(x)$ будем выбирать так, чтобы он минимизировал ошибку предыдущей композиции (т.е. $b_1(x) + b_2(x)$) и т.д.

ГРАДИЕНТНЫЙ БУСТИНГ

Пусть $L(y, z)$ – произвольная дифференцируемая функция потерь. Строим алгоритм $a_N(x)$ вида

$$\boldsymbol{a_L(x) = \sum_{n=1}^{L} \gamma_n b_n(x)}$$

ГРАДИЕНТНЫЙ БУСТИНГ

Пусть $L(y, z)$ – произвольная дифференцируемая функция потерь. Строим алгоритм $a_N(x)$ вида

$$a_L(x) = \sum_{n=1}^{L} \gamma_n b_n(x),$$

где на N-м шаге

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

$$s_i^{(N)} = y_i - a_{N-1}(x_i)?$$

ГРАДИЕНТНЫЙ БУСТИНГ

Пусть $L(y, z)$ – произвольная дифференцируемая функция потерь. Строим алгоритм $a_N(x)$ вида

$$a_L(x) = \sum_{n=1}^{L} \gamma_n b_n(x),$$

где на N-м шаге

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

$$\cancel{s_i^{(N)} = y_i - a_{N-1}(x_i)} \qquad s_i^{(N)} = -\frac{\partial L}{\partial z}$$

ГРАДИЕНТНЫЙ БУСТИНГ

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

Строим алгоритм $a_N(x)$ вида

$$a_L(x) = \sum_{n=1}^{L} \gamma_n b_n(x),$$

где на N-м шаге

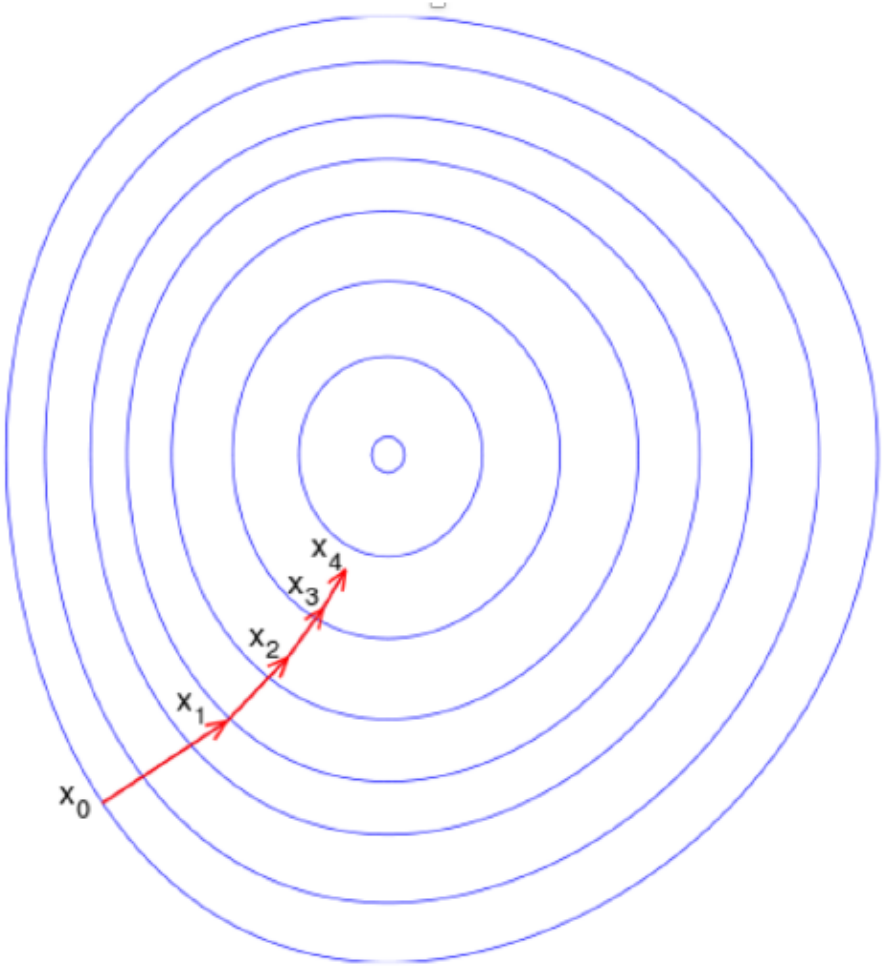
$$b_N(x) = \underset{b \in A}{\operatorname{argmin}} \sum_{i=1}^{l} \left(b(x_i) - s_i^{(N)} \right)^2,$$

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

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



БУСТИНГ: ВЫБОР БАЗОВЫХ АЛГОРИТМОВ

- Если базовые алгоритмы очень простые, то они плохо приближают антиградиент функции потерь, т.е. градиентный бустинг может свестись к случайному блужданию.
- Если базовые алгоритмы сложные, то за несколько шагов бустинг подгонится под обучающую выборку, и получим переобученный алгоритм.

БУСТИНГ: ВЫБОР БАЗОВЫХ АЛГОРИТМОВ

- Если базовые алгоритмы очень простые, то они плохо приближают антиградиент функции потерь, т.е. градиентный бустинг может свестись к случайному блужданию.
- Если базовые алгоритмы сложные, то за несколько шагов бустинг подгонится под обучающую выборку, и получим переобученный алгоритм.

Чаще всего в качестве базовых алгоритмов используют ***решающие деревья***.

БУСТИНГ: ВЫБОР БАЗОВЫХ АЛГОРИТМОВ

- Если базовые алгоритмы очень простые, то они плохо приближают антиградиент функции потерь, т.е. градиентный бустинг может свестись к случайному блужданию.
- Если базовые алгоритмы сложные, то за несколько шагов бустинг подгонится под обучающую выборку, и получим переобученный алгоритм.

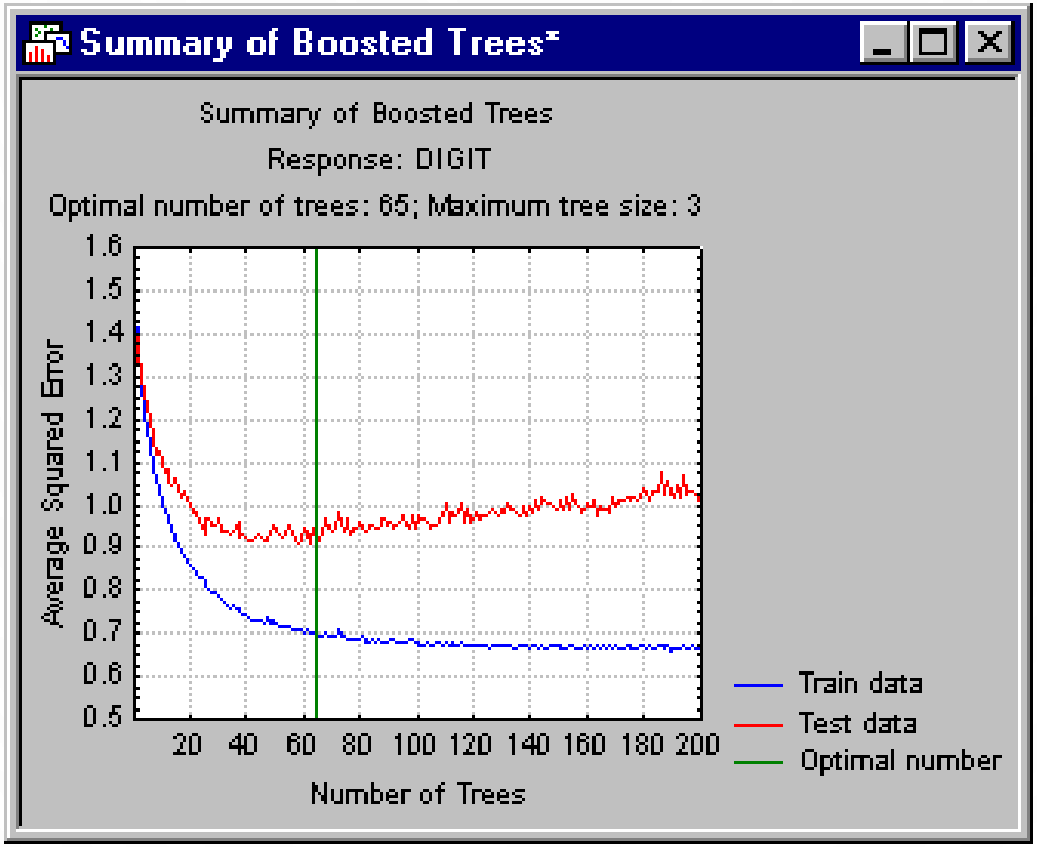
Чаще всего в качестве базовых алгоритмов используют ***решающие деревья***.

В таком случае *решающие деревья не должны быть очень маленькими, а также очень глубокими*.

Оптимальная глубина – от 3 до 6 (зависит от задачи).

КОЛИЧЕСТВО ИТЕРАЦИЙ БУСТИНГА

Так как на каждом шаге бустинга целенаправленно уменьшается ошибка на тренировочной выборке, то если процесс не остановить, то мы достигнем нулевой ошибки, а значит, переобучимся!



СМЕЩЕНИЕ И РАЗБРОС БУСТИНГА

- 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).