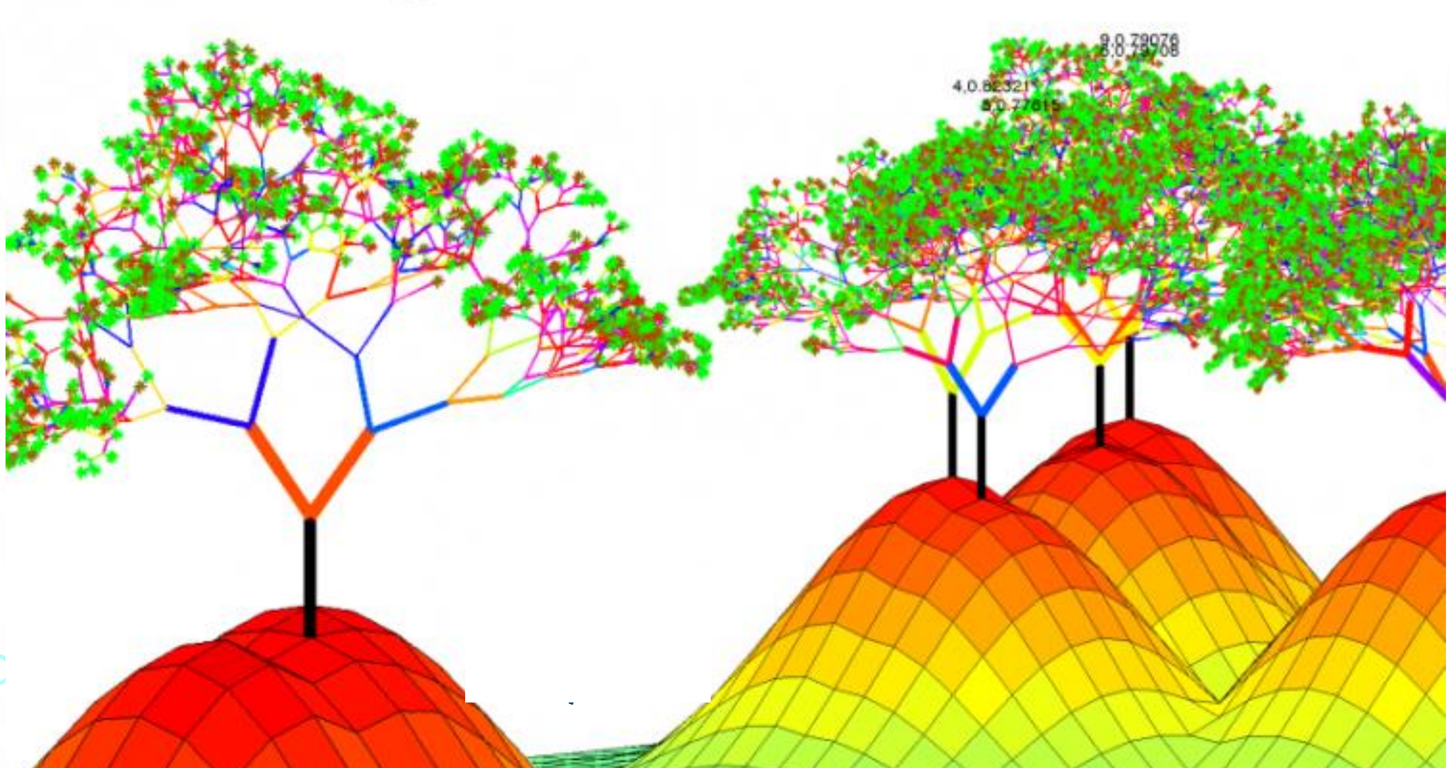
Лекция 10 Композиции алгоритмов.

Кантонистова Е.О.



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*

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

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*

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

- $\mathbf{Bias(a(x))}$ - 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)

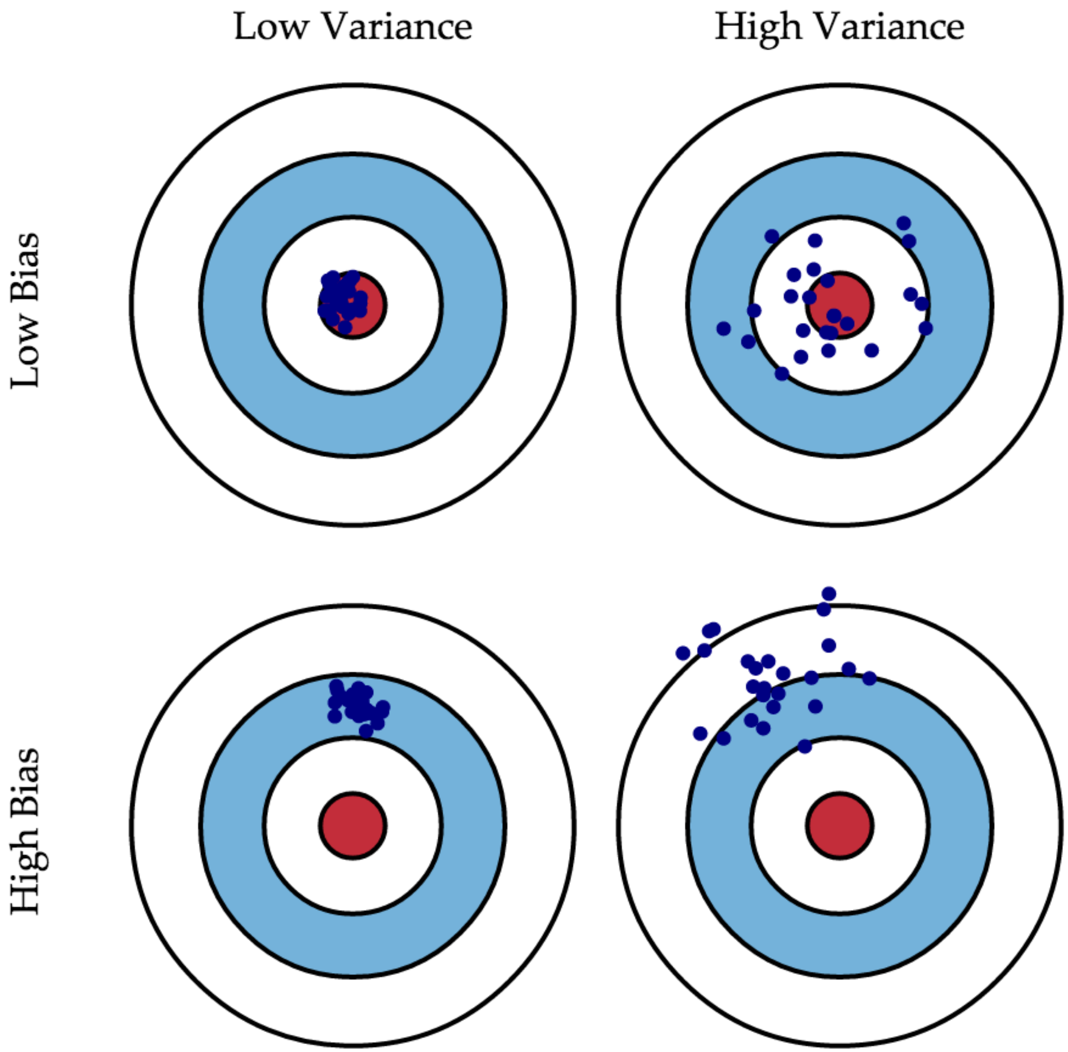
Зачастую для улучшения качества модели необходимо понять, из-за чего возникает ошибка в предсказаниях.

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

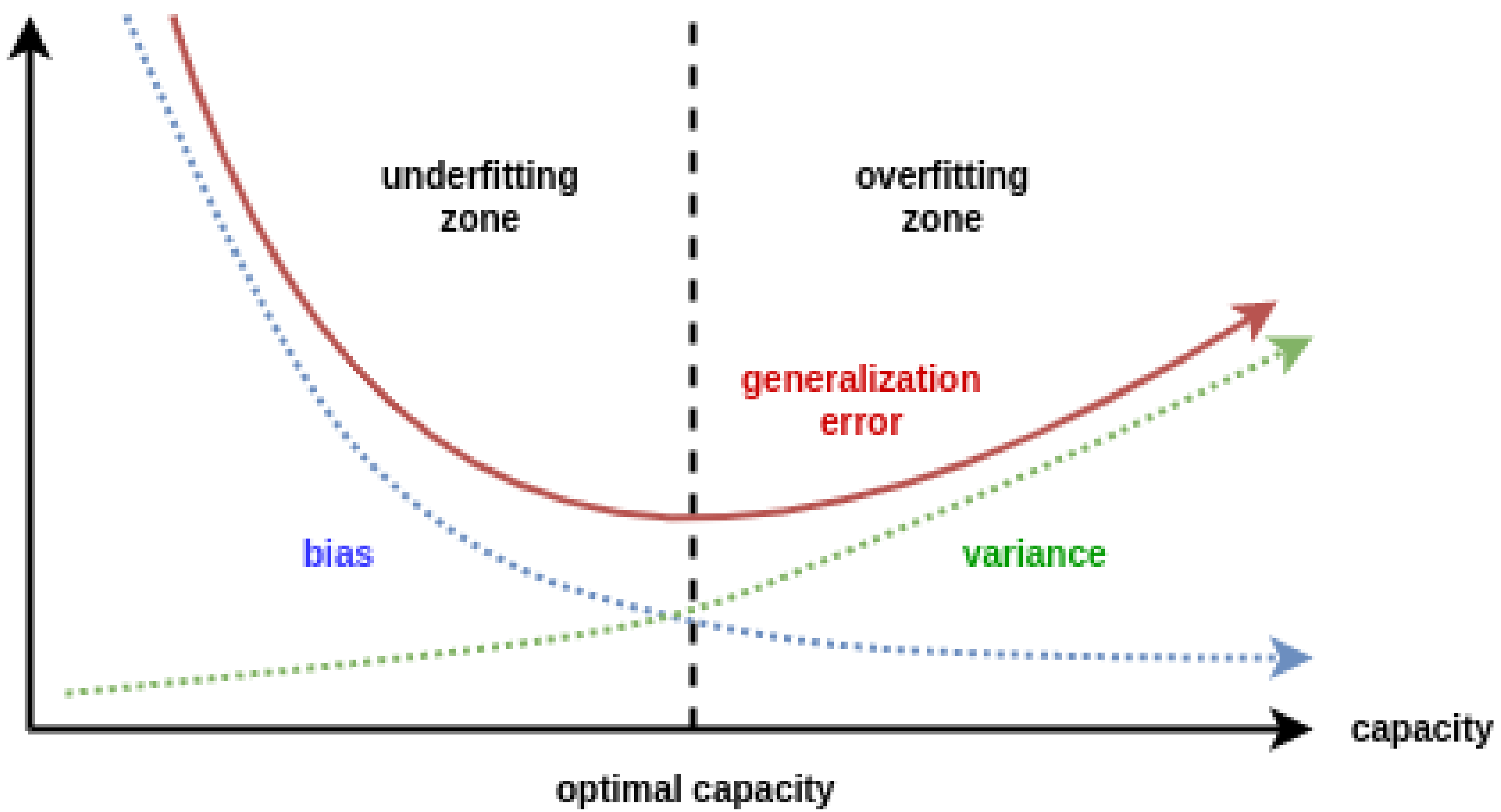
$$\mathbf{Err(x) = Bias^2\big(a(x)\big) + Var\big(a(x)\big) + \sigma^2}.$$

- $\mathbf{Bias\big(a(x)\big)}$ - 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**.
- $\boldsymbol{\sigma}^\mathbf{2}$ - неустранимая ошибка – **шум**.

СМЕЩЕНИЕ И РАЗБРОС



BIAS-VARIANCE TRADEOFF

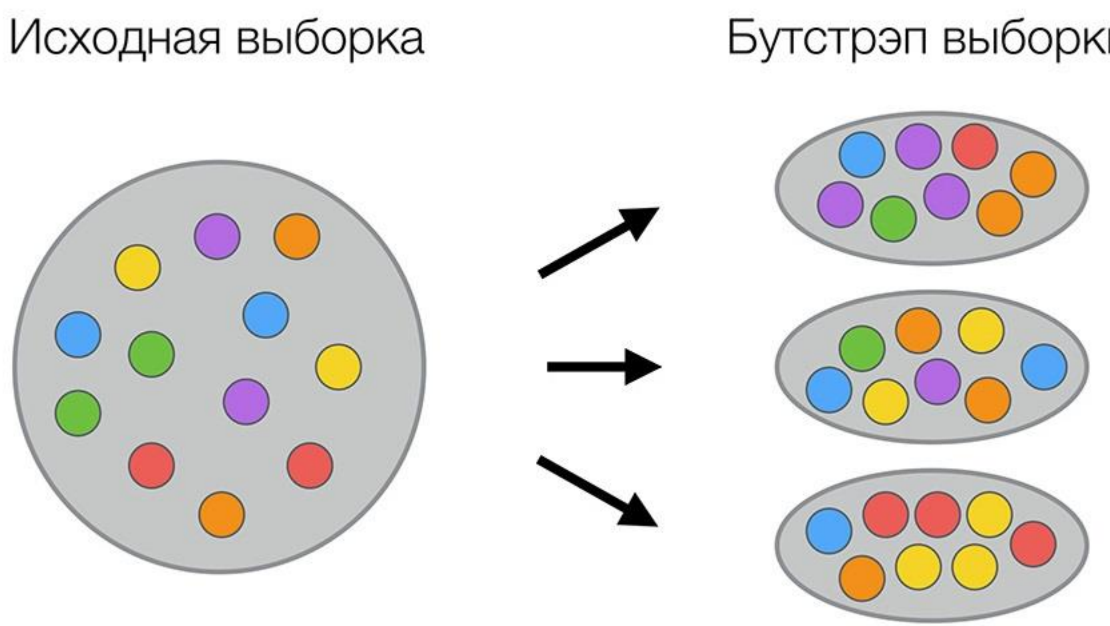


БУТСТРЭП

Дана выборка X.

<u>Бутстрэп:</u> 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 $X_1, \ldots, X_N$.



Бэггинг (Bootstrap aggregation)

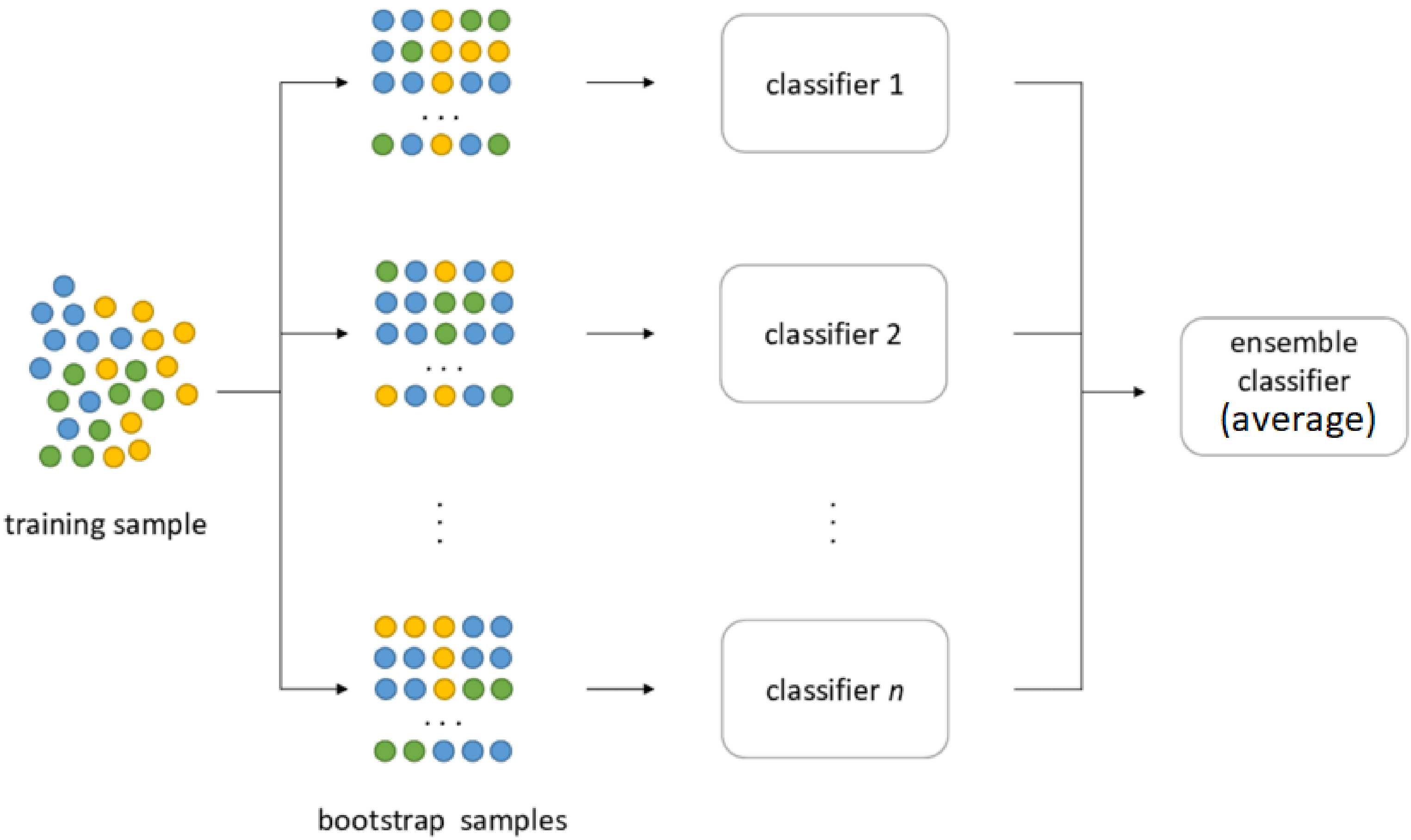
С помощью бутстрэпа мы получили выборки $X_1, \dots, X_N$.

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

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

БЭГГИНГ (BOOTSTRAP AGGREGATION)

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



СМЕЩЕНИЕ И РАЗБРОС У БЭГГИНГА

Бэггинг: $a_N(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x) = \frac{1}{N}\sum_{n=1}^{N} \tilde{\mu}(X)\,(x)$

(здесь $\tilde{\mu}(X) = \mu(\tilde{X})$ – алгоритм, обученный на подвыборке $\tilde{X}$)

Утверждение (с док-вом):

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.

RANDOM FOREST

Алгоритм 3.1. Random Forest

для $n = 1, \dots, N$

Сгенерировать выборку $\tilde{X}_n$ с помощью бутстрэпа

Построить решающее дерево $b_n(x)$ по выборке $\tilde{X}_n$:

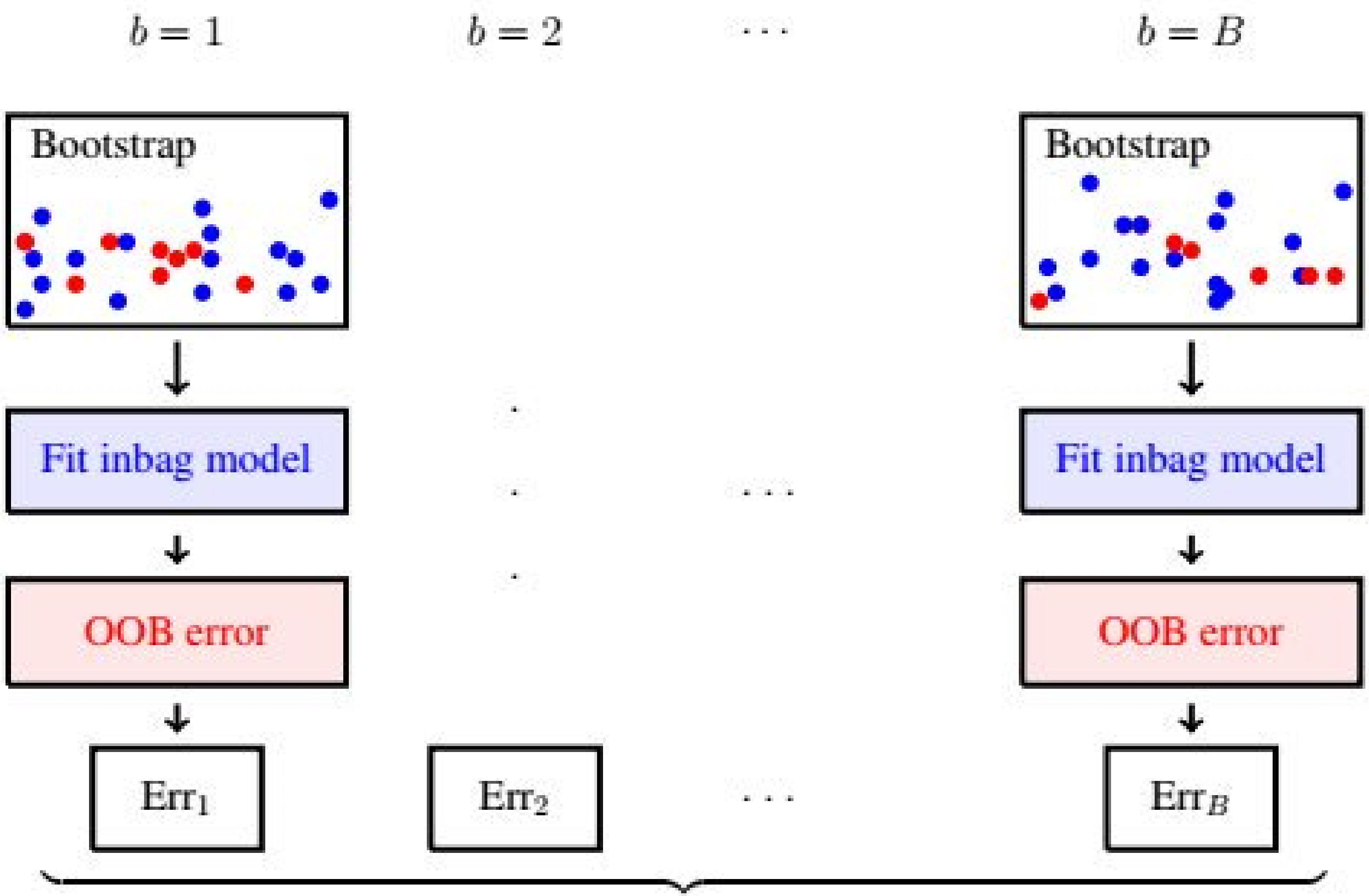
- дерево строится, пока в каждом листе не окажется не более $n_{\min}$ объектов
- при каждом разбиении сначала выбирается m случайных признаков из p, и оптимальное разделение ищется только среди них

Вернуть композицию $a_N(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x)$

RANDOM FOREST – ПРАКТИЧЕСКИЕ РЕКОМЕНДАЦИИ

- Если p – количество признаков, то при классификации обычно берут $m = [\sqrt{p}]$, а при регрессии - $m = [\frac{p}{3}]$ признаков
- При классификации обычно дерево строится, пока в листе не окажется $n_{min} = 1$ объект, а при регрессии $n_{min} = 5$

OUT-OF-BAG ОШИБКА



$$\mathrm{Err}_{\mathrm{oob}} = \frac{\mathrm{Err}_1 + \cdots + \mathrm{Err}_B}{B} = \frac{1}{B}\sum_{b=1}^{B} \mathrm{Err}_b$$

OUT-OF-BAG ОШИБКА

- Каждое дерево в случайном лесе обучается по некоторому подмножеству объектов
- Значит, для каждого объекта есть деревья, которые на этом объекте не обучались.

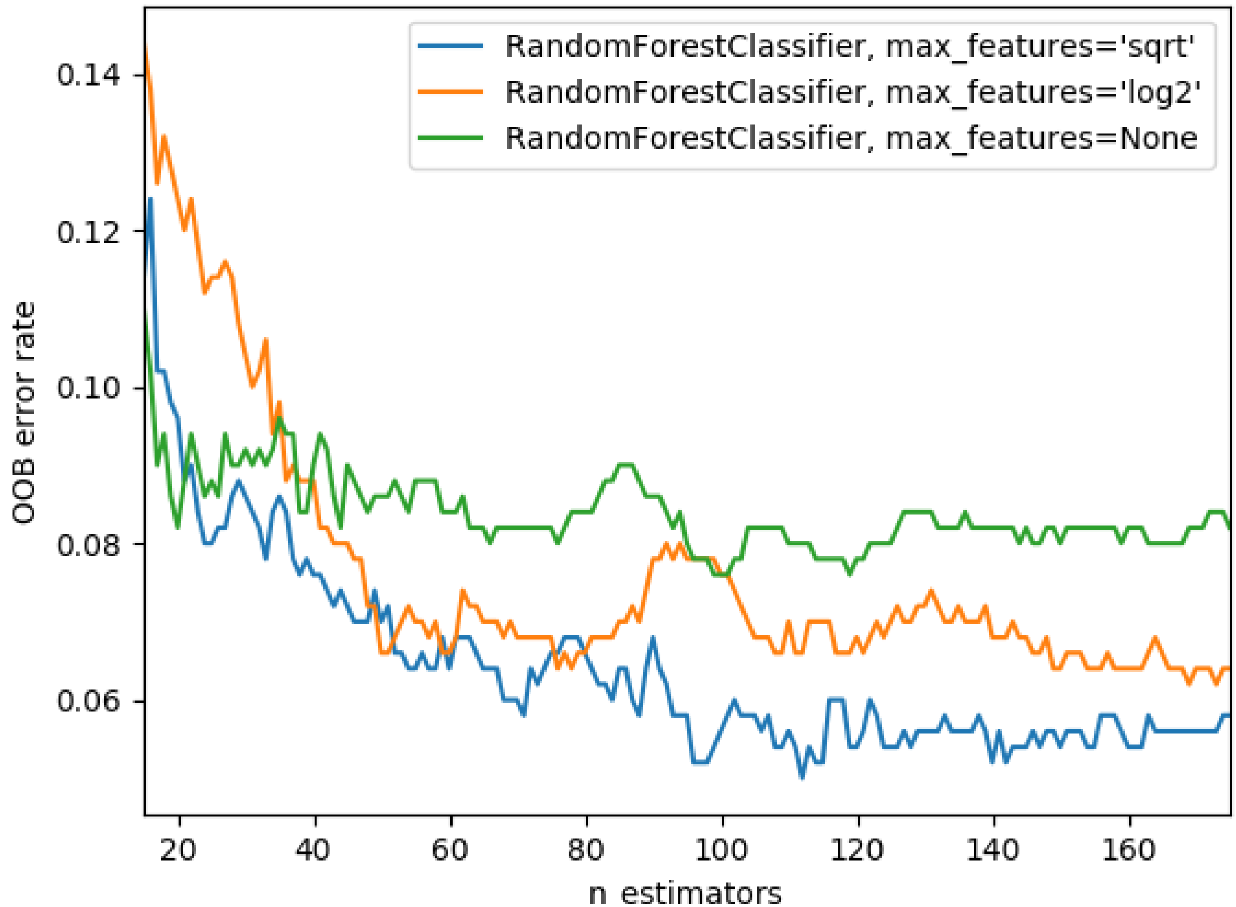
Out-of-bag ошибка:

$$OOB = \sum_{i=1}^{l} L\left(y_i, \frac{\sum_{n=1}^{N} [x_i \notin X_n]\, b_n(x_i)}{\sum_{n=1}^{N} [x_i \notin X_n]}\right)$$

Утверждение. *При $N \to \infty$ OOB оценка стремится к leave-one-out оценке.*

OOB-SCORE

По графику out-of-bag ошибки можно, например, подбирать количество деревьев в случайном лесе

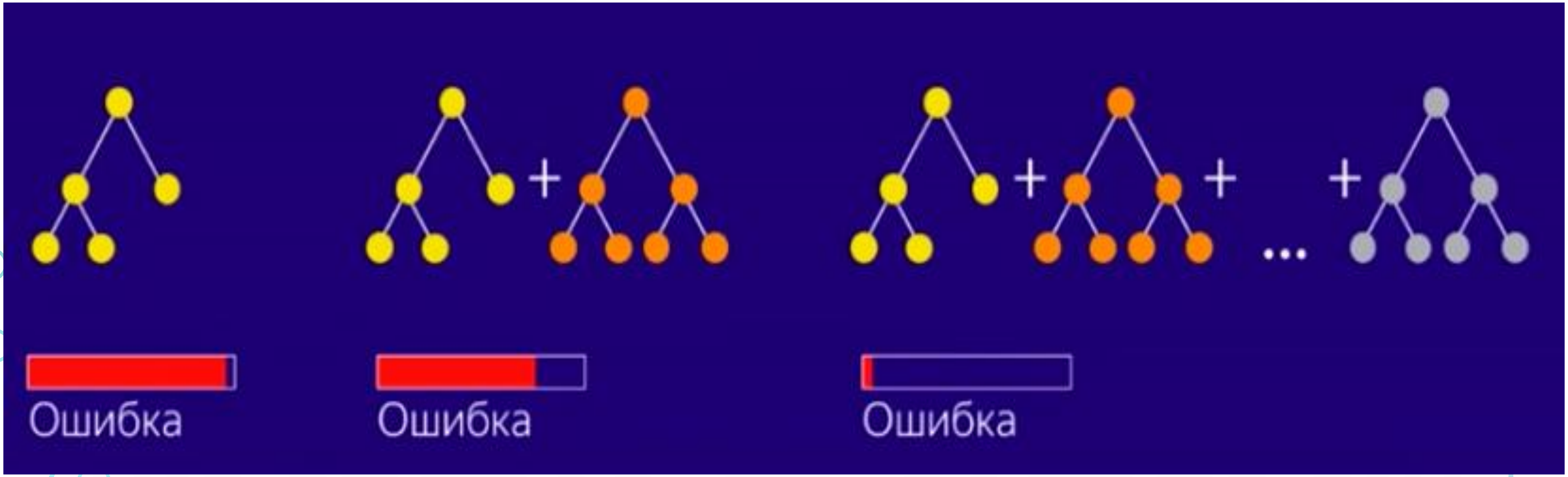


БУСТИНГ

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

БУСТИНГ

<u>Идея</u>: 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

<u>Шаг 1</u>: Ищем алгоритм $b_1(x)$, минимизирующий ошибку:

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

- 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:

$$\boldsymbol{s} = y - b_1(x)$$

Следующий алгоритм должен настраиваться на эту ошибку, т.е. ***целевая переменная для следующего алгоритма – это вектор ошибок*** $\boldsymbol{s}$ (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:

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

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:

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

Следующий алгоритм $\mathrm{b}_3(\mathrm{x})$ будем выбирать так, чтобы он минимизировал ошибку предыдущей композиции (т.е. $\mathrm{b}_1(\mathrm{x}) + \mathrm{b}_2(\mathrm{x})$) и т.д.

БУСТИНГ В ЗАДАЧЕ РЕГРЕССИИ

Каждый следующий алгоритм настраиваем на ошибку предыдущих.

Шаг N: Ошибка: $\boldsymbol{s_i^{(N)}} = y_i - \sum_{n=1}^{N-1} b_n(x_i) = y_i - a_{N-1}(x_i)$

Ищем алгоритм $b_N(x)$:

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

СМЕЩЕНИЕ И РАЗБРОС БУСТИНГА

- Бустинг целенаправленно уменьшает ошибку, т.е. смещение у него маленькое.
- Алгоритм получается сложным, поэтому разброс может быть большим.

Значит, чтобы не переобучиться, в качестве базовых алгоритмов надо брать неглубокие деревья (глубины 3-6).