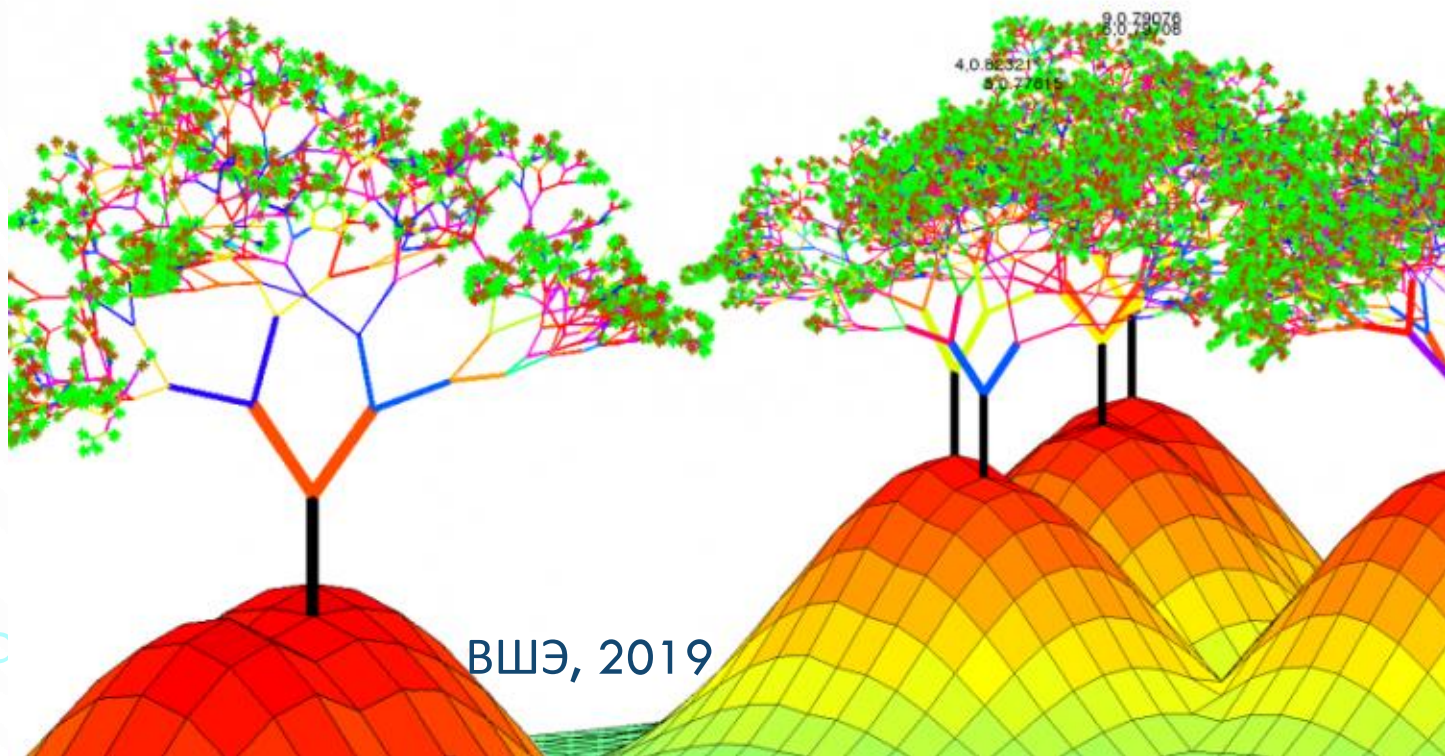


# Композиции алгоритмов. Бэггинг.

Кантонистова Е.О.

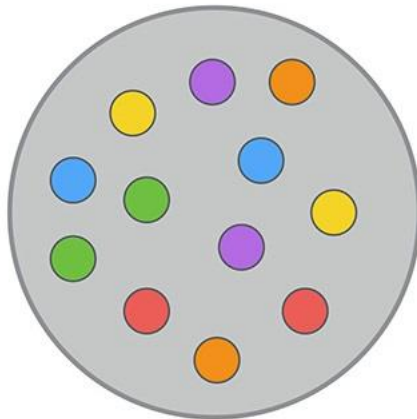


# БУТСТРЭП

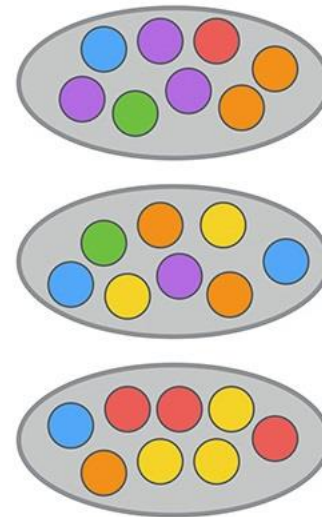
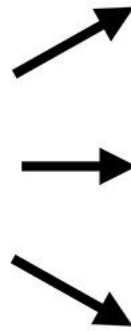
Дана выборка  $X$ . Решаем задачу регрессии.

- **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  $X_1, \dots, X_N$ .

Исходная выборка



Бутстрэп выборки



# 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)

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:

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

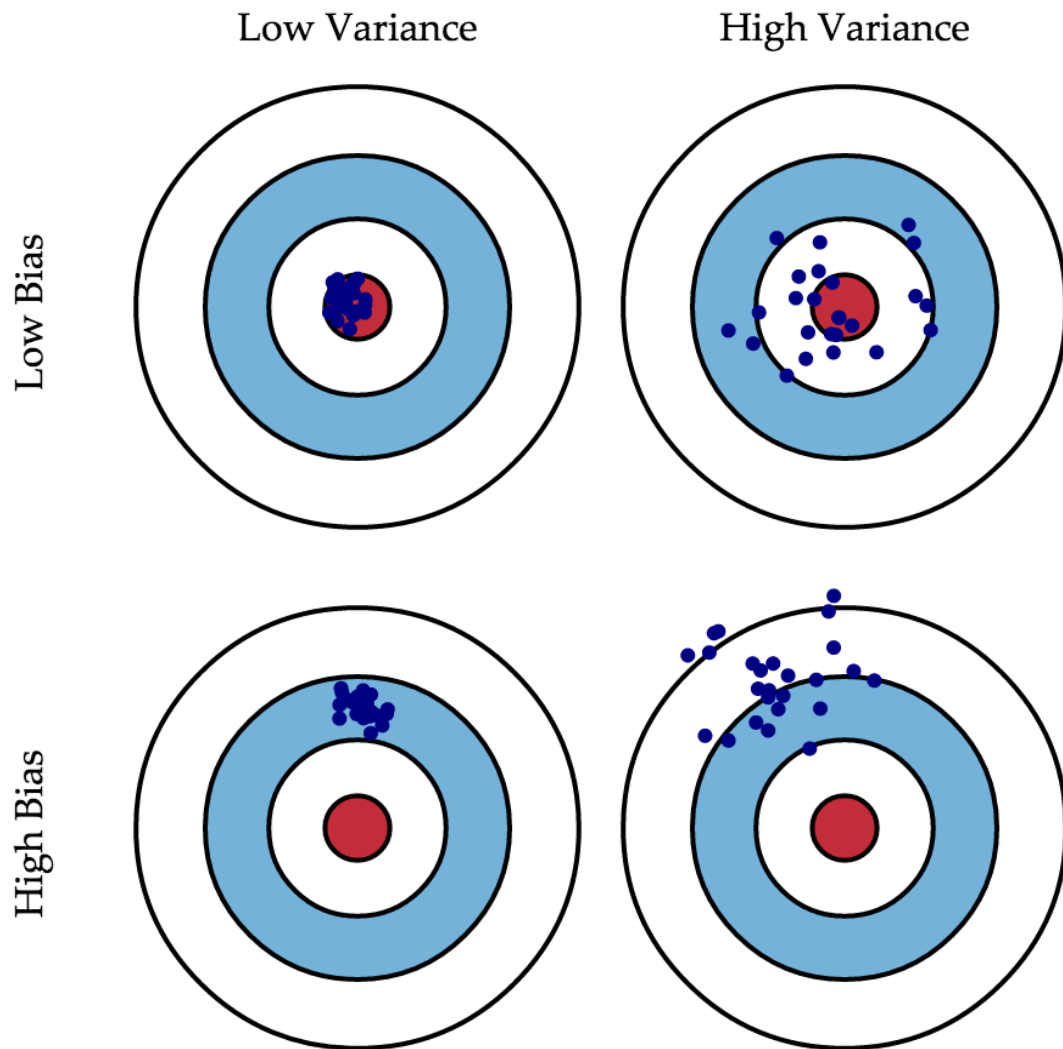
**Утверждение.** Если алгоритмы  $b_1(x), \dots, b_N(x)$  некоррелированы, то среднеквадратичная ошибка алгоритма  $a(x)$ , полученного при помощи бэггинга, в  $N$  раз меньше среднеквадратичной ошибки исходных алгоритмов  $b_j(x)$ .

# РАЗЛОЖЕНИЕ ОШИБКИ (BIAS-VARIANCE DECOMPOSITION)

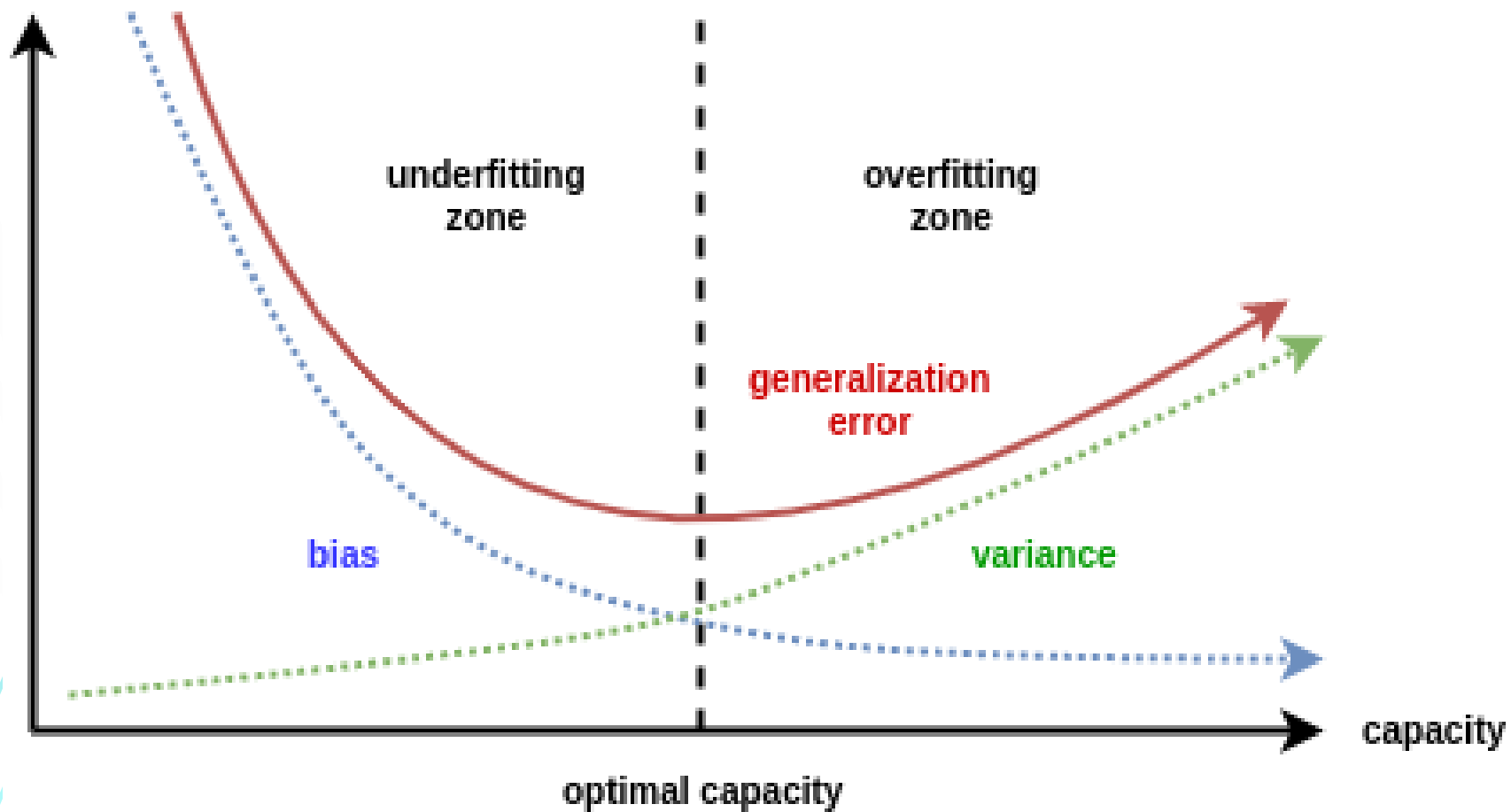
**Утверждение.**

$$\begin{aligned} L(\mu) = & \mathbb{E}_{x,y}[(y - \mathbb{E}[y|x])^2] \text{ (шум)} \\ & + \mathbb{E}_{x,y}[(\mathbb{E}_X[\mu(X)] - \mathbb{E}[y|x])^2] \text{ (смещение)} \\ & + \mathbb{E}_{x,y}[\mathbb{E}_X[(\mu(X) - \mathbb{E}_X[\mu(X)])^2]] \text{ (разброс)} \end{aligned}$$

# СМЕЩЕНИЕ И РАЗБРОС



# BIAS-VARIANCE TRADEOFF



# 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.

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

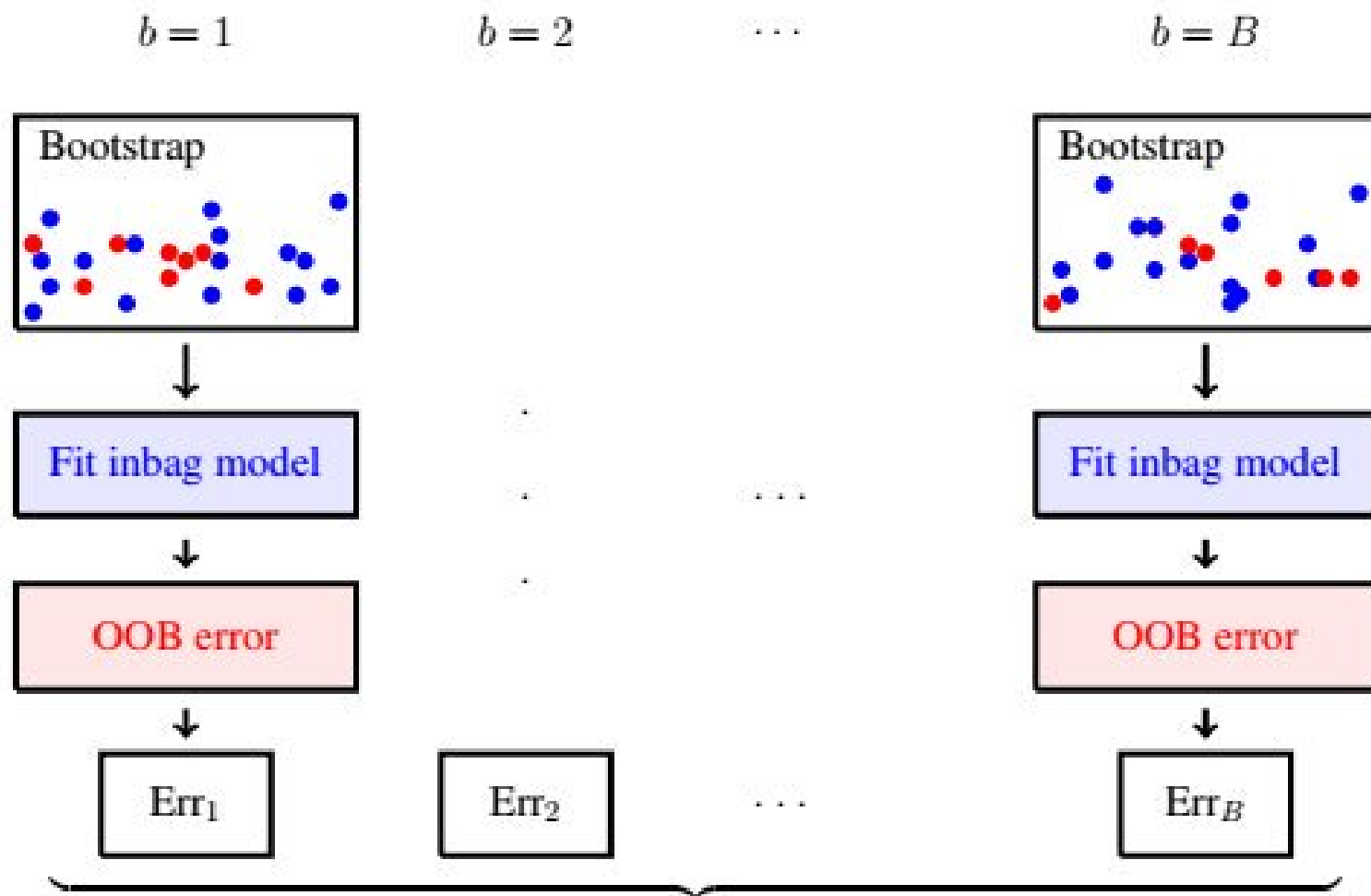
---

- 1: для  $n = 1, \dots, N$
  - 2: Сгенерировать выборку  $\tilde{X}_n$  с помощью бутстрэпа
  - 3: 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
  - 4: 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  $m = \lfloor \sqrt{p} \rfloor$ , а при регрессии -  $m = \lfloor \frac{p}{3} \rfloor$  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



# 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:

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

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

# OOB-SCORE

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

