

# Boosting

Algorithms

# AdaBoost.M1

1. Инициализируются веса:

$$w_i = \frac{1}{N}, i = 1, 2, \dots, N$$

2. С  $m=1$  по  $M$ :

а) Обучается модель  $G_m(x)$ , используя веса из шага 1

# AdaBoost.M1

b) Вычисляется ошибка:

$$err_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x))}{\sum_{i=1}^N w_i}$$

c) Вычисляется:

$$a_m = \log((1 - err_m)/err_m)$$

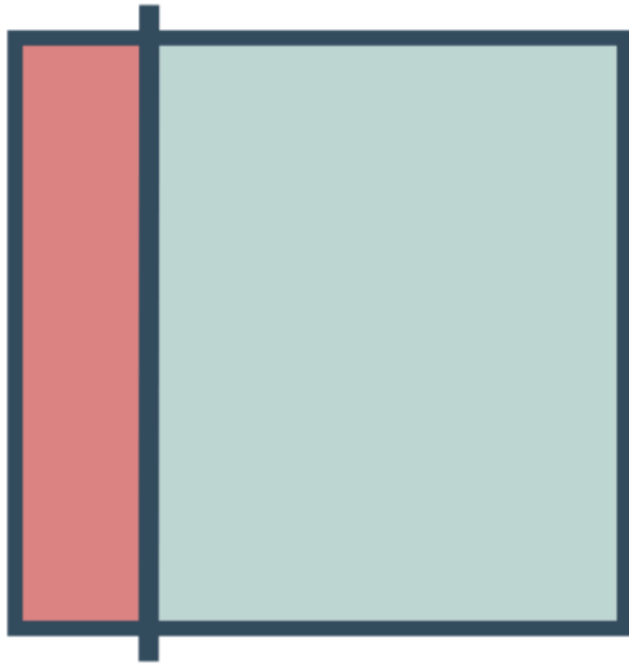
# AdaBoost.M1

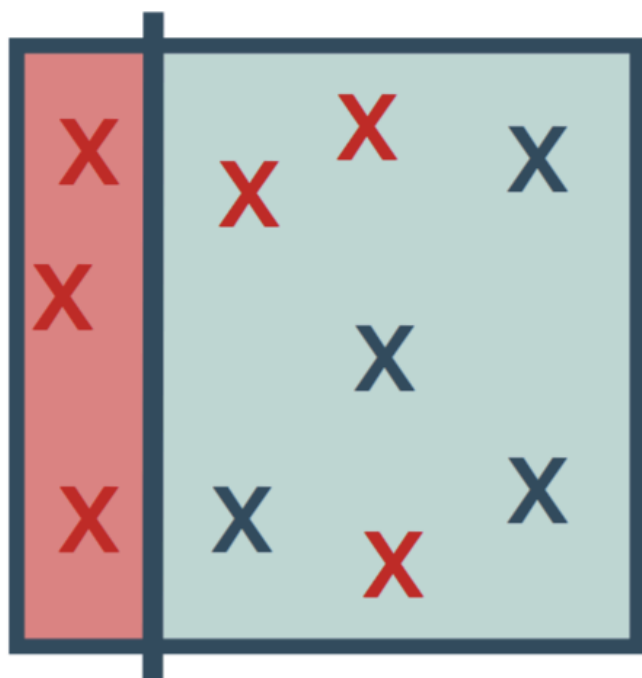
b) Присваивается новый вес:

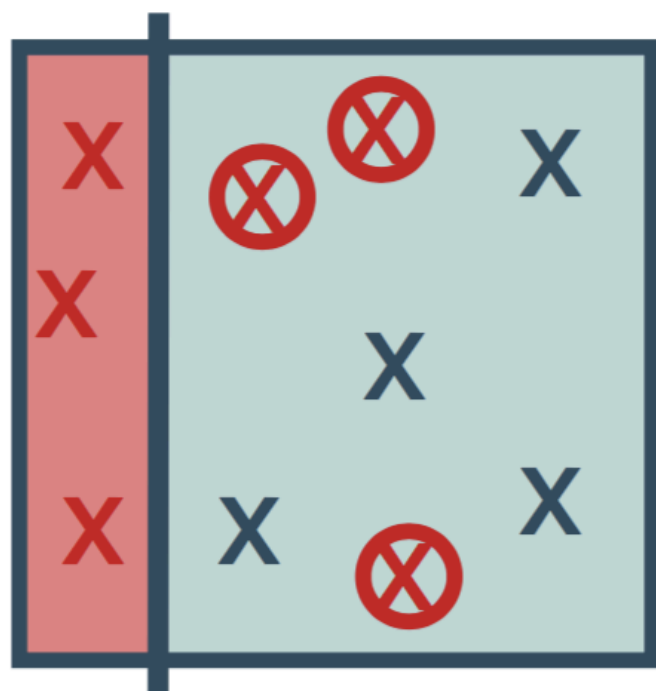
$$w_i \leftarrow w_i \cdot \exp[a_m \cdot I(y_i \neq G_m(x))], i = 1, 2, \dots, N$$

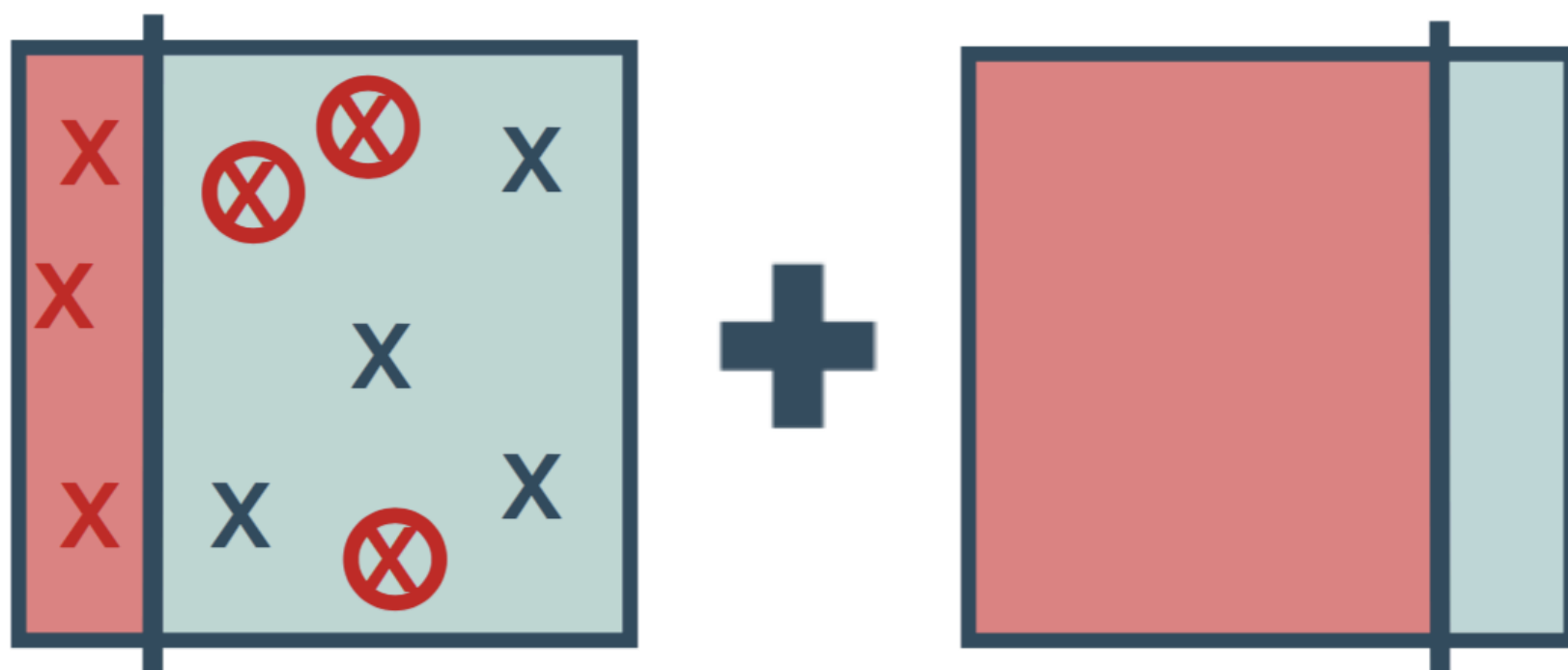
3.

$$H(x) = \text{sign} \left( \sum_{m=1}^M a_m G_m(x) \right)$$

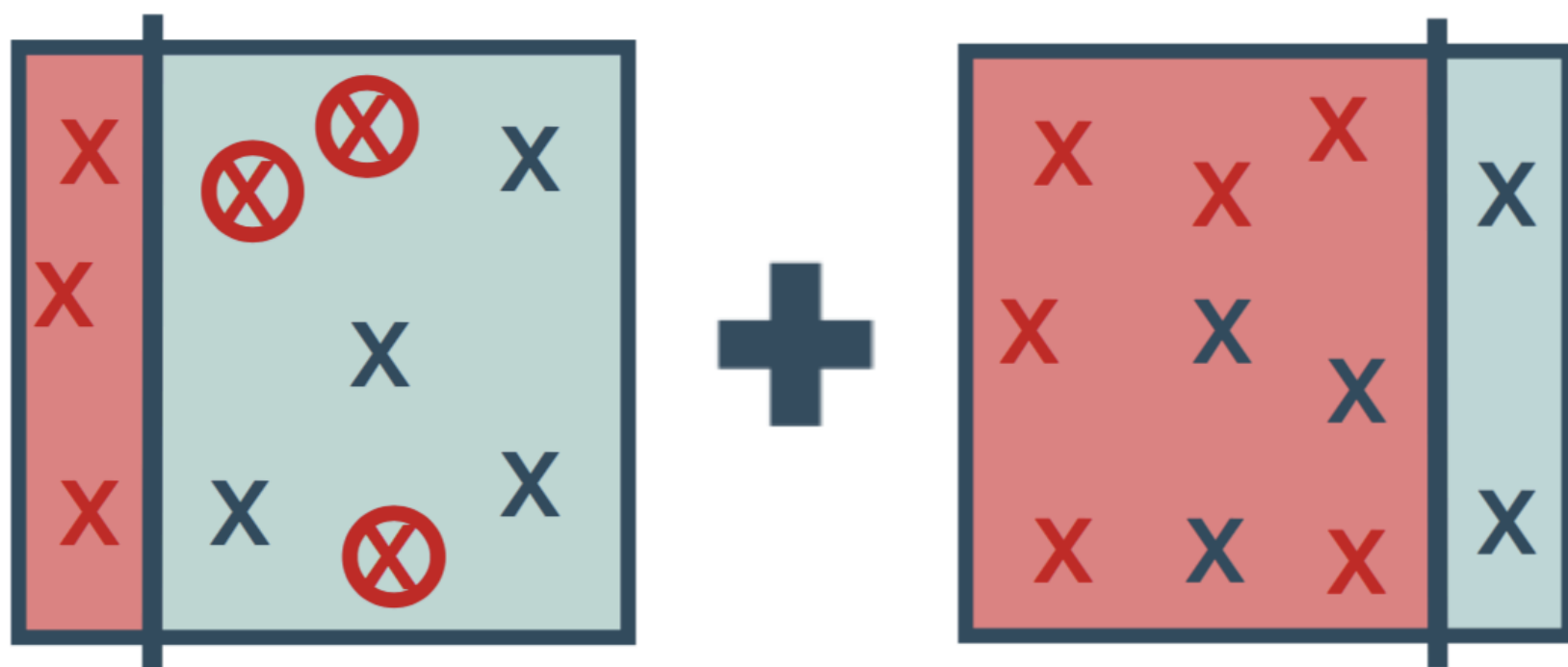


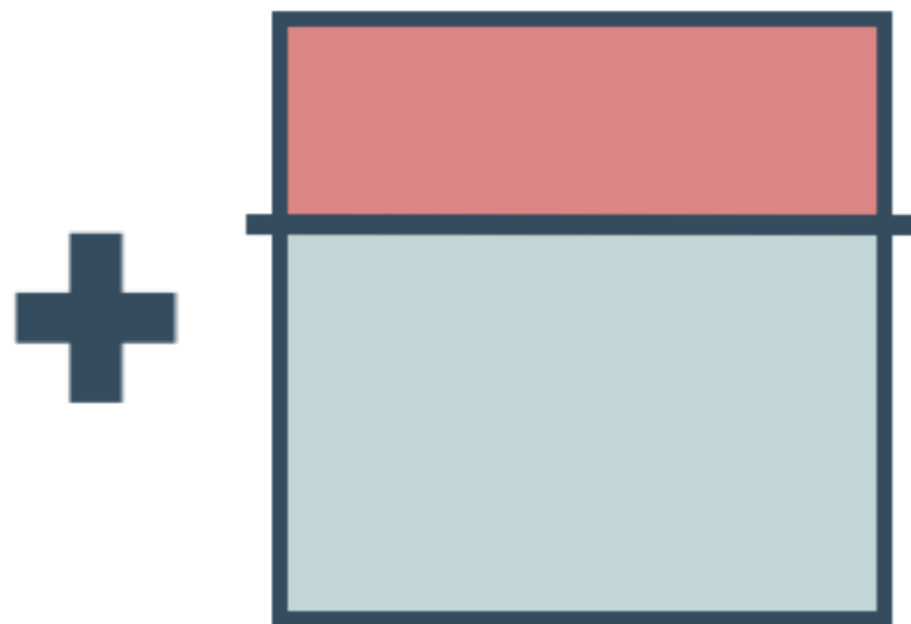
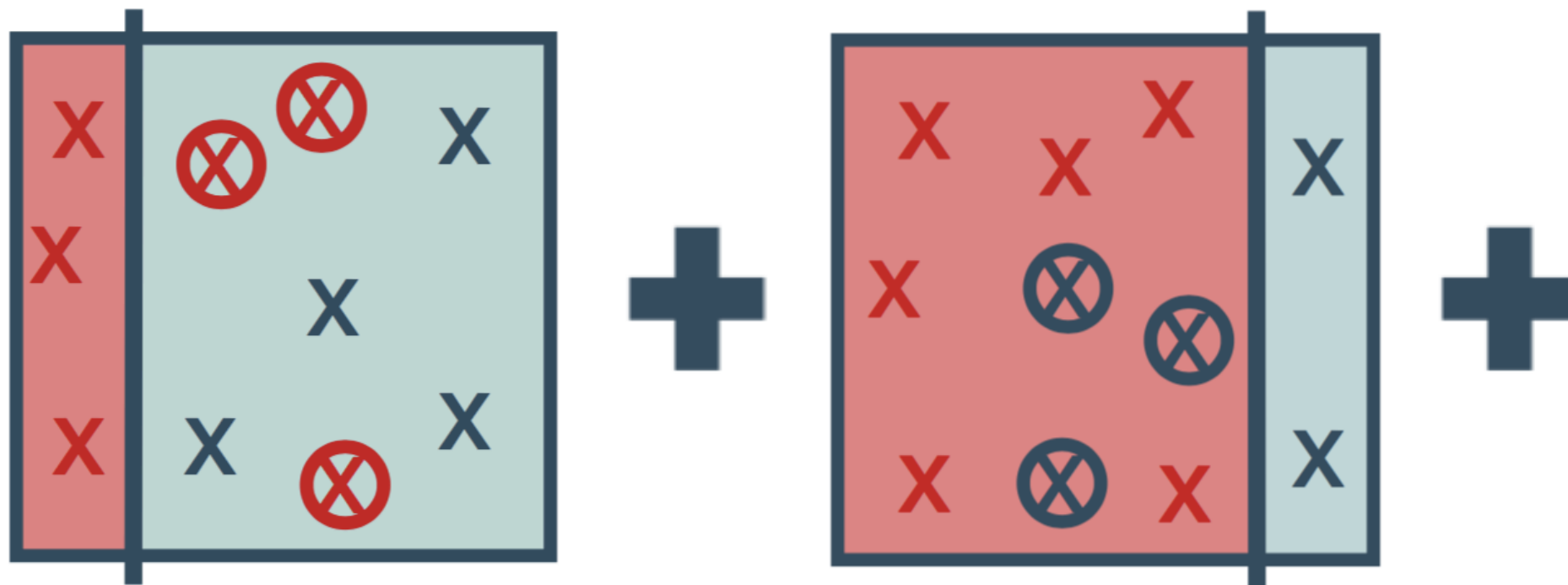


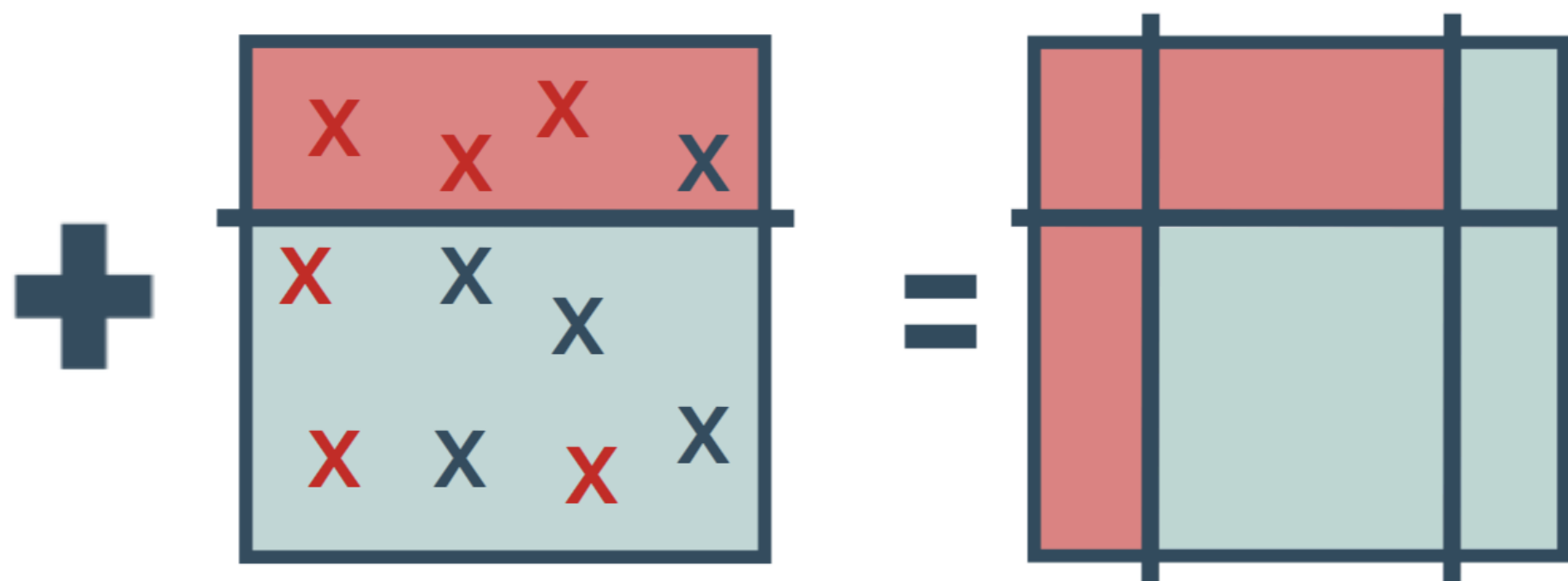
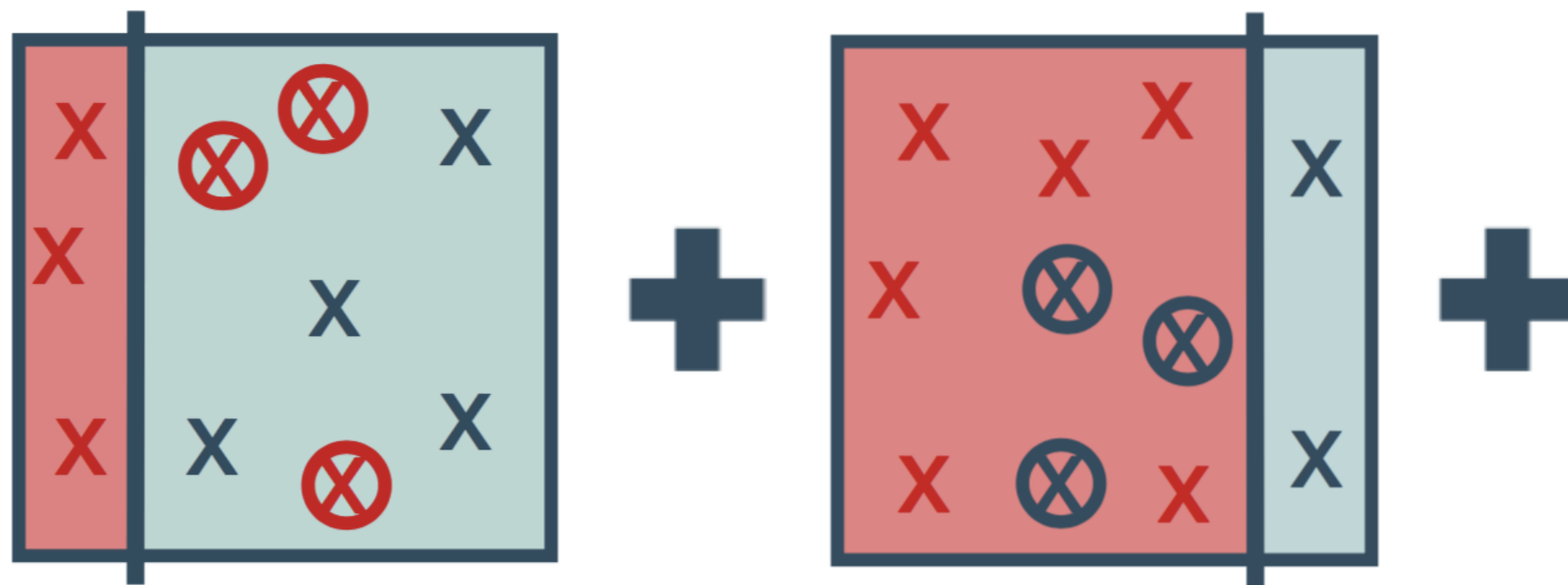












# Gradient Boosting

- функция потерь, которую нужно оптимизировать
- слабая модель для предсказаний
- аддитивная модель для добавления слабых моделей и минимизации функции потерь

1. Initialize  $f_0(x) = \arg \min_{\gamma} \sum_{i=1}^N L(y_i, \gamma)$ .

2. For  $m = 1$  to  $M$ :

(a) For  $i = 1, 2, \dots, N$  compute

$$r_{im} = - \left[ \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f_{m-1}}.$$

(b) Fit a regression tree to the targets  $r_{im}$  giving terminal regions  $R_{jm}$ ,  $j = 1, 2, \dots, J_m$ .

(c) For  $j = 1, 2, \dots, J_m$  compute

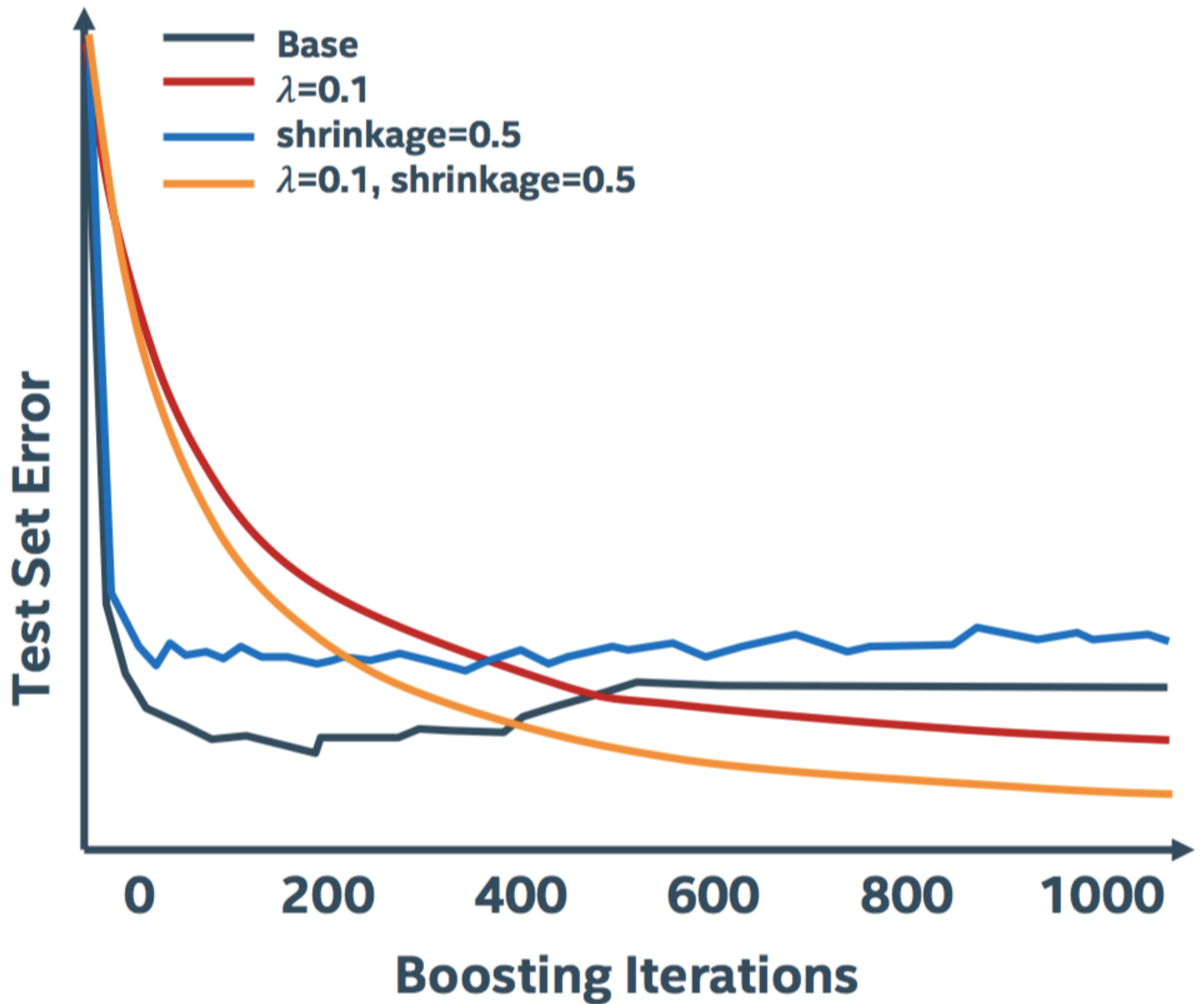
$$\gamma_{jm} = \arg \min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma).$$

(d) Update  $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$ .

3. Output  $\hat{f}(x) = f_M(x)$ .

# Регуляризация

- Shrinkage
- Стохастический градиентный бустинг
- Параметры деревьев



# XGBoost

- Системные фишки:
  - Параллелизация
  - Распределенные вычисления (Hadoop, Spark)
  - Оптимизация кеша
- Поддерживает модели:
  - Градиентный бустинг
  - Стохастический градиентный бустинг