

Лекция 2

Линейные методы регрессии. Часть 1.

Кантонистова Е.О.

ВШЭ, 2019

ЛИНЕЙНАЯ РЕГРЕССИЯ

Пусть x^1, \dots, x^d - признаки объекта x .

Линейная регрессия:

$$a(x) = w_0 + \sum_{j=1}^d w_j x^j = (w, x)$$

ЛИНЕЙНАЯ РЕГРЕССИЯ

Линейная регрессия:

$$a(x) = w_0 + \sum_{j=1}^d w_j x^j = (w, x)$$

Обучение линейной регрессии (минимизация среднеквадратичной ошибки):

$$Q(a, X) = \frac{1}{l} \sum_{i=1}^l (a(x_i) - y_i)^2 = \frac{1}{l} \sum_{i=1}^l ((w, x_i) - y_i)^2 \rightarrow \min_w$$

АНАЛИТИЧЕСКОЕ РЕШЕНИЕ ЗАДАЧИ МНК

Задача обучения линейной регрессии (в матричной форме):

$$\frac{1}{\ell} \|Xw - y\|^2 \rightarrow \min_w$$

Точное (аналитическое) решение:

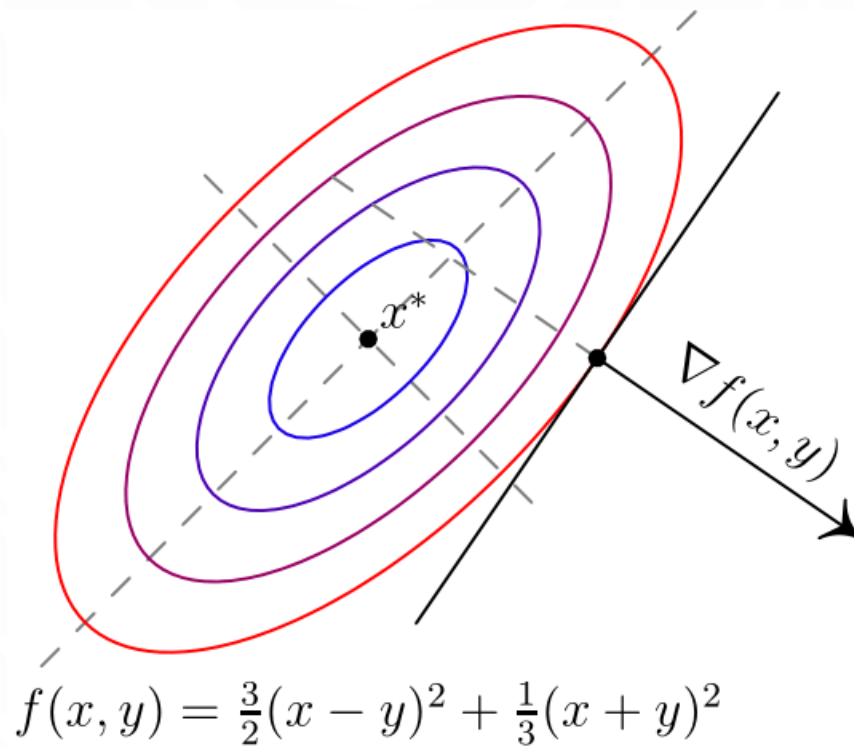
$$w = (X^T X)^{-1} X^T y$$

НЕДОСТАТКИ АНАЛИТИЧЕСКОЙ ФОРМУЛЫ

- Обращение матрицы – сложная операция ($O(N^3)$ от числа признаков)
- Матрица $X^T X$ может быть вырожденной или плохо обусловленной
- Если заменить среднеквадратичный функционал ошибки на другой, то скорее всего не найдем аналитическое решение

ТЕОРЕМА О ГРАДИЕНТЕ

Теорема. Градиент – это направление наискорейшего роста функции.



ГРАДИЕНТНЫЙ СПУСК

Теорема. Градиент – это направление наискорейшего роста функции.

Метод градиентного спуска:

- Инициализируем веса $w^{(0)}$.
- На каждом следующем шаге обновляем веса по формуле:

$$w^{(k)} = w^{(k-1)} - \eta_k \nabla Q(w^{(k-1)})$$

ГРАДИЕНТНЫЙ СПУСК

Теорема. Градиент – это направление наискорейшего роста функции.

Метод градиентного спуска:

- Инициализируем веса $w^{(0)}$.
- На каждом следующем шаге обновляем веса по формуле:

$$w^{(k)} = w^{(k-1)} - \eta_k \nabla Q(w^{(k-1)})$$

Скорость сходимости: $Q(w^{(k)}) - Q(w^*) = O(\frac{1}{k})$

ГРАДИЕНТНЫЙ СПУСК

$$w^{(k)} = w^{(k-1)} - \eta_k \nabla Q(w^{(k-1)})$$

Градиент функции Q :

$$\nabla Q(w) = \sum_{i=1}^l \nabla q_i(w)$$

Градиентный спуск:

$$w^{(k)} = w^{(k-1)} - \eta_k \sum_{i=1}^l \nabla q_i(w^{(k-1)})$$

ВАРИАНТЫ ИНИЦИАЛИЗАЦИИ ВЕСОВ

- $w_j = 0, j = 1, \dots, n$
- Небольшие случайные значения:

$$w_j := \text{random}(-\varepsilon, \varepsilon)$$

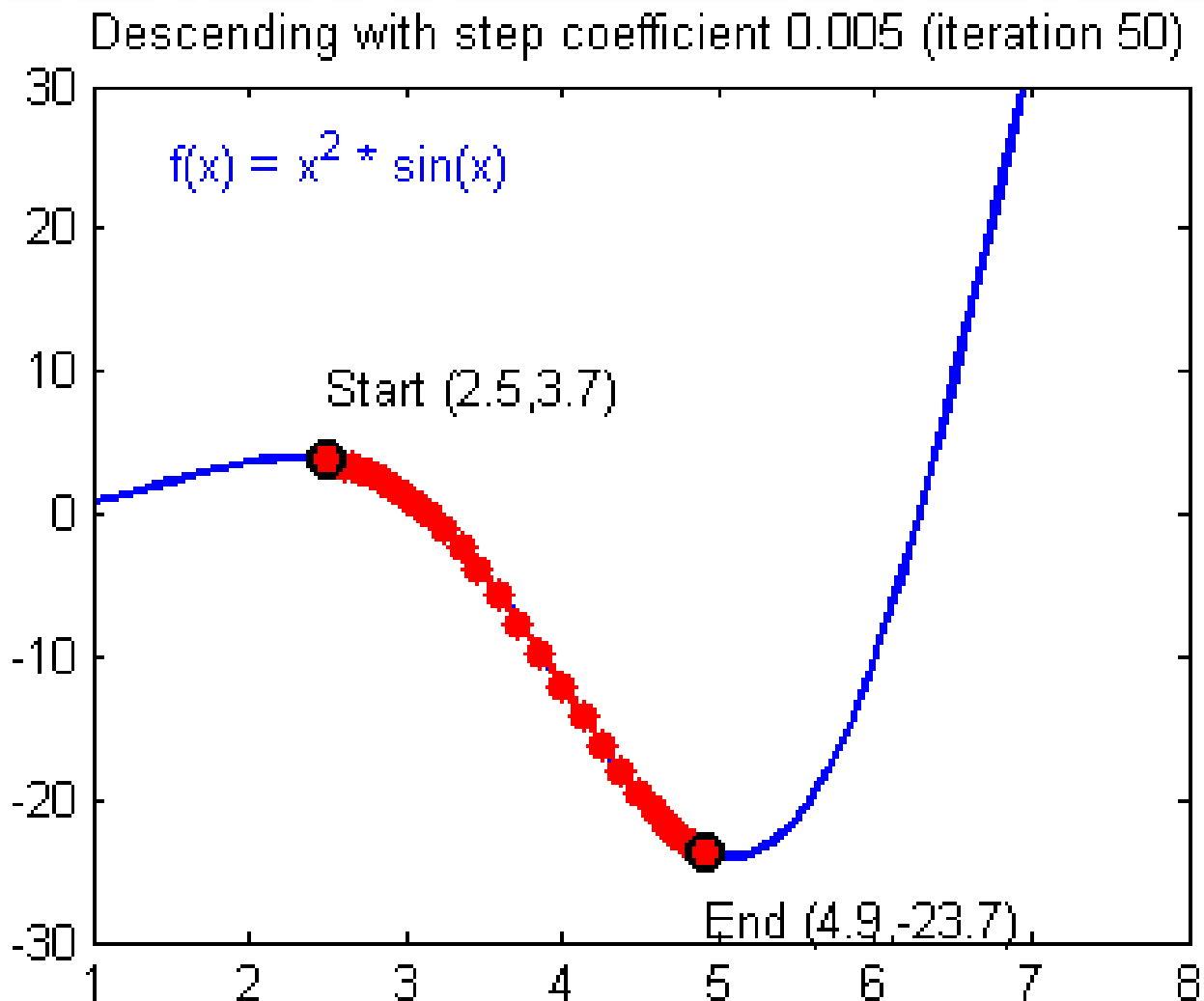
- Обучение по небольшой случайной подвыборке объектов
- Мультистарт: многократный запуск из разных случайных начальных приближений и выбор лучшего решения

КРИТЕРИИ ОСТАНОВА

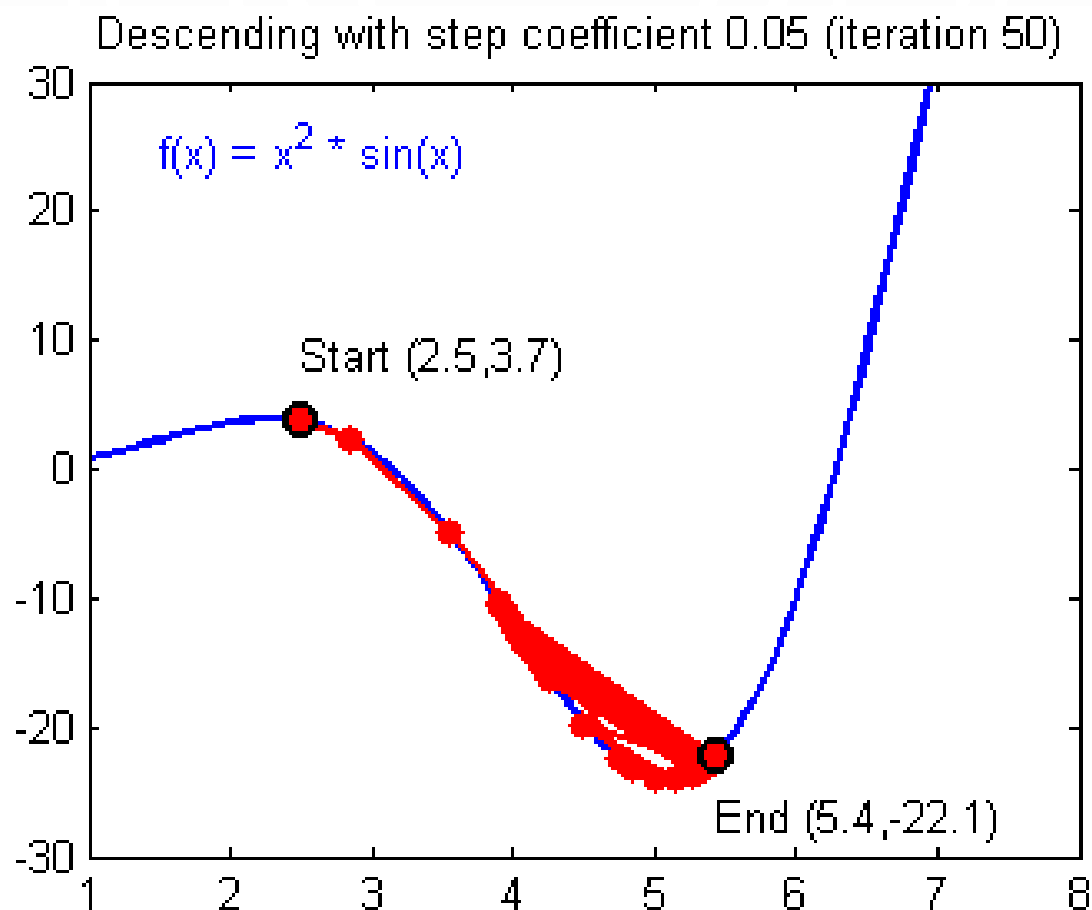
- $|\nabla Q(w^{(k-1)})| < \varepsilon$

- $\Delta w = |w^{(k)} - w^{(k-1)}| < \varepsilon$

ГРАДИЕНТНЫЙ СПУСК



ПРОБЛЕМА ВЫБОРА ГРАДИЕНТНОГО ШАГА



ГРАДИЕНТНЫЙ ШАГ

- $\eta_k = c$
- $\eta_k = \frac{1}{k}$
- $\eta_k = \lambda \left(\frac{s_0}{s_0+k} \right)^p$, λ, s_0, p - параметры

МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SGD

1) Stochastic gradient descent (SGD):

- на каждом шаге выбираем один случайный объект и сдвигаемся в сторону антиградиента по этому объекту:

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \eta_k \cdot \nabla q_{i_k}(\mathbf{w}^{(k-1)})$$

МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SGD

1) Stochastic gradient descent (SGD):

- на каждом шаге выбираем один случайный объект и сдвигаемся в сторону антиградиента по этому объекту:

$$w^{(k)} = w^{(k-1)} - \eta_k \cdot \nabla q_{i_k}(w^{(k-1)})$$

Скорость сходимости: $E[Q(w^{(k)}) - Q(w^*)] = \mathcal{O}(\frac{1}{\sqrt{k}})$

МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SGD

1) Stochastic gradient descent (SGD):

- на каждом шаге выбираем один случайный объект и сдвигаемся в сторону антиградиента по этому объекту:

$$w^{(k)} = w^{(k-1)} - \eta_k \cdot \nabla q_{i_k}(w^{(k-1)})$$

Скорость сходимости: $E[Q(w^{(k)}) - Q(w^*)] = O(\frac{1}{\sqrt{k}})$

+ Менее трудоемкий метод

- Медленнее сходится

МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SAG

2) Stochastic average gradient (SAG):

- Инициализируем веса w_j
- Инициализируем вспомогательные переменные $z^{(1)}, z^{(2)}, \dots$:

$$z^{(i)} = \nabla q_i(w)$$

МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SAG

2) Stochastic average gradient (SAG):

- Инициализируем веса w_j
- Инициализируем вспомогательные переменные $z^{(1)}, z^{(2)}, \dots$:

$$z^{(i)} = \nabla q_i(w)$$

- На каждом шаге выбираем один случайный объект и обновляем градиент по нему:

$$z_i^{(k)} = \begin{cases} \nabla q_i(w^{(k-1)}), i = i_k \\ z_i^{(k-1)}, \text{ иначе} \end{cases}$$

МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SAG

2) Stochastic average gradient (SAG):

- Инициализируем веса w_j
- Инициализируем вспомогательные переменные $z^{(1)}, z^{(2)}, \dots$:

$$z^{(i)} = \nabla q_i(w)$$

- На каждом шаге выбираем один случайный объект и обновляем градиент по нему:

$$z_i^{(k)} = \begin{cases} \nabla q_i(w^{(k-1)}), & i = i_k \\ z_i^{(k-1)}, & \text{иначе} \end{cases}$$

- Формула градиентного шага:

$$w^{(k)} = w^{(k-1)} - \eta_k \sum_{i=1}^l z_i^{(k)}$$

МЕТОДЫ ОЦЕНИВАНИЯ ГРАДИЕНТА: SAG

2) Stochastic average gradient (SAG):

- Формула градиентного шага:

$$\mathbf{w}^{(k)} = \mathbf{w}^{(k-1)} - \eta_k \sum_{i=1}^l \mathbf{z}_i^{(k)}$$

Скорость сходимости: $\mathbf{E}[Q(\mathbf{w}^{(k)}) - Q(\mathbf{w}^*)] = \mathcal{O}(\frac{1}{k})$

ПРОБЛЕМЫ ГРАДИЕНТНОГО СПУСКА

- Медленно сходится
- Застревает в локальных минимумах

ПРОБЛЕМА ЗАСТРЕВАНИЯ В LOSMIN



МЕТОД МОМЕНТОВ (MOMENTUM)

Вектор инерции (*усреднение градиента по предыдущим шагам*):

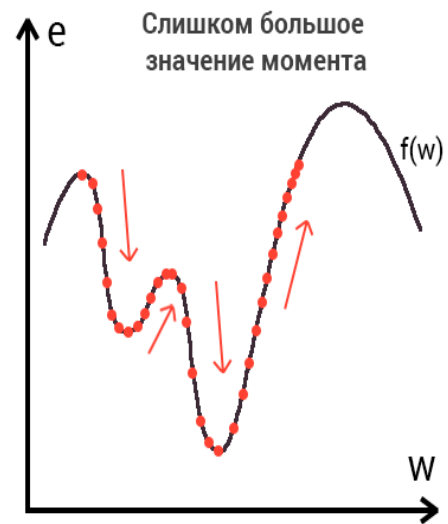
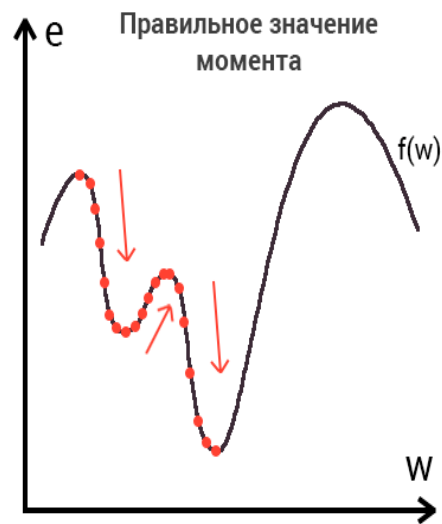
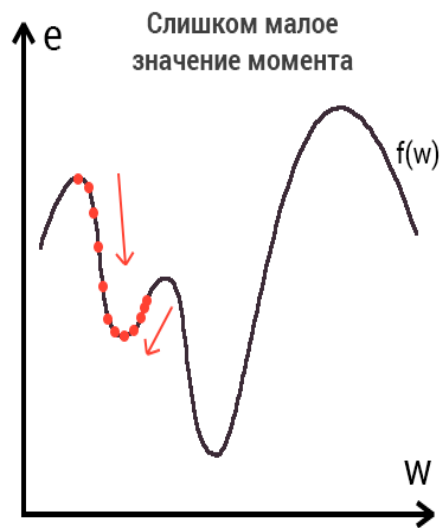
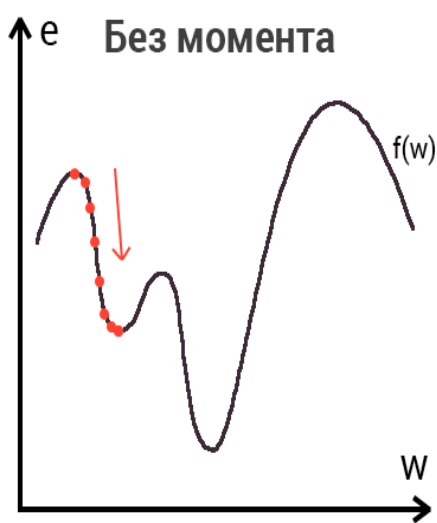
$$h_0 = 0;$$

$$h_k = \alpha h_{k-1} + \eta_k \nabla_w Q(w^{(k-1)})$$

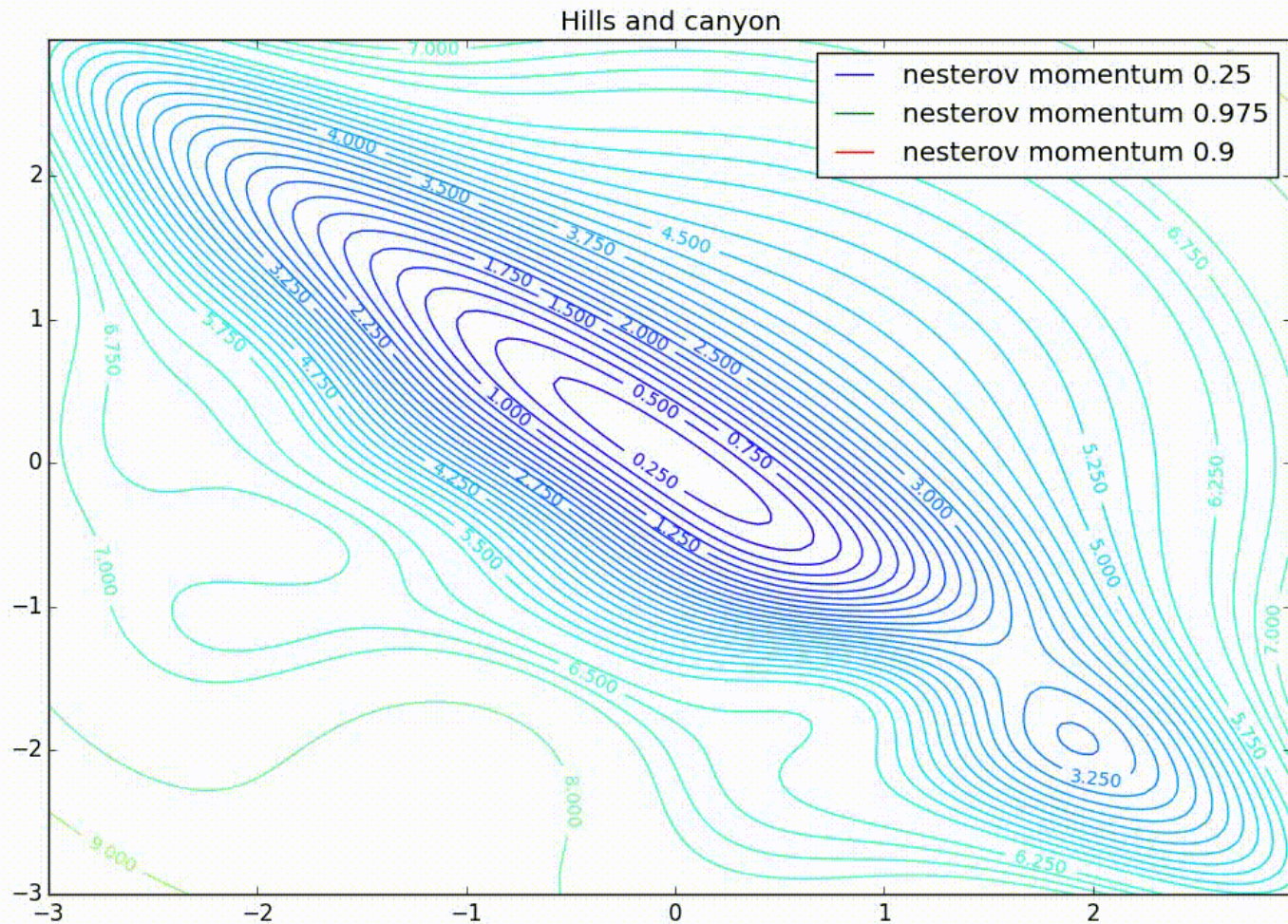
Формула метода моментов:

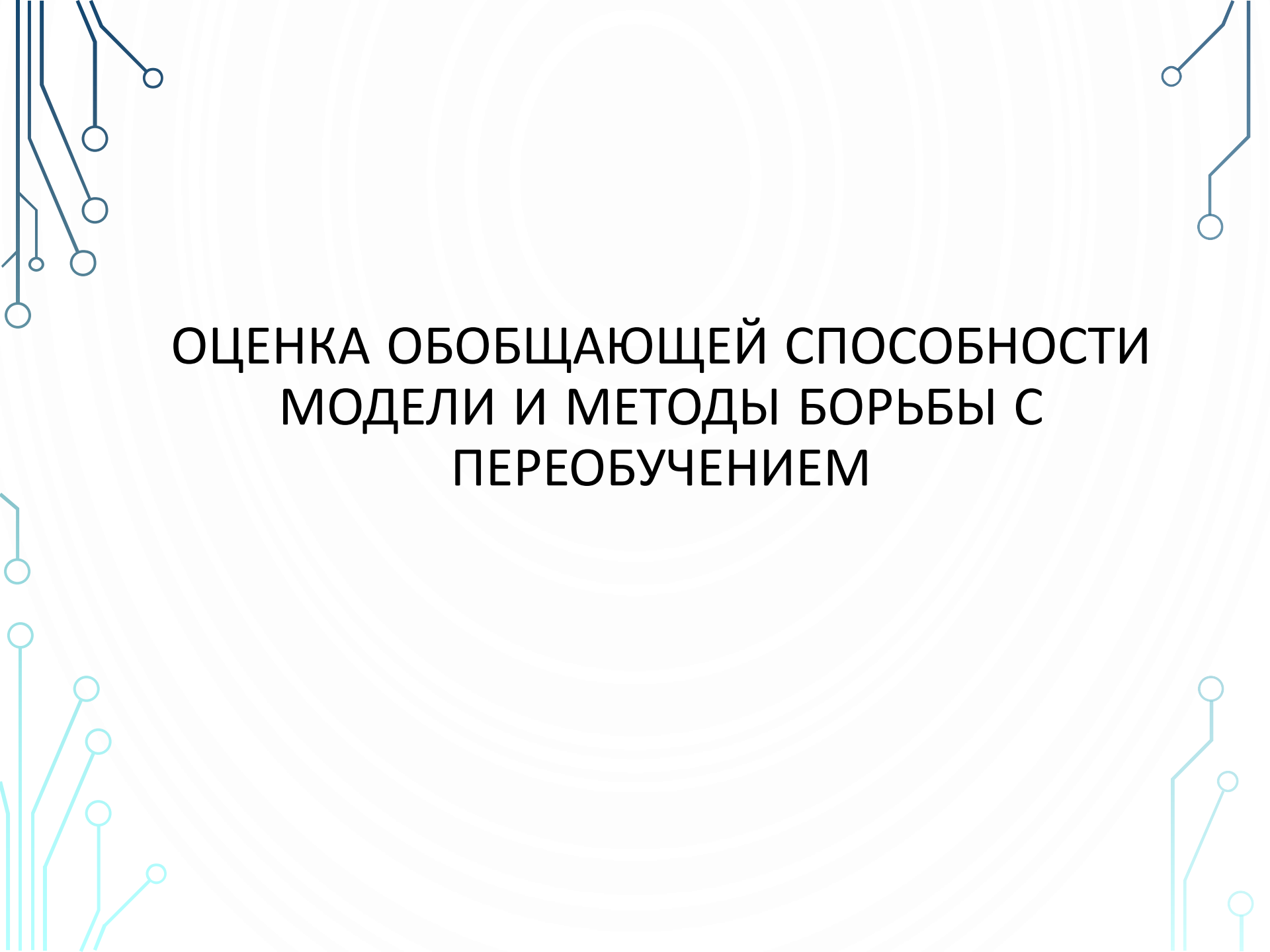
$$w^{(k)} = w^{(k-1)} - h_k$$

MOMENTUM



MOMENTUM



The background features a light gray grid of concentric circles. In the four corners, there are decorative circuit-like patterns. The top-left and top-right corners have dark blue lines with small circles at the ends. The bottom-left and bottom-right corners have light blue lines with small circles at the ends.

ОЦЕНКА ОБОБЩАЮЩЕЙ СПОСОБНОСТИ МОДЕЛИ И МЕТОДЫ БОРЬБЫ С ПЕРЕОБУЧЕНИЕМ

ОЦЕНКА ОБОБЩАЮЩЕЙ СПОСОБНОСТИ МОДЕЛИ

Переобучение (overfitting) – явление, при котором качество модели на новых данных сильно хуже, чем качество на тренировочных данных.

Fitting training data

Degree = 1

— True function
— Model
• Training data (MSE = 1.37)

Underfitting

Degree = 2

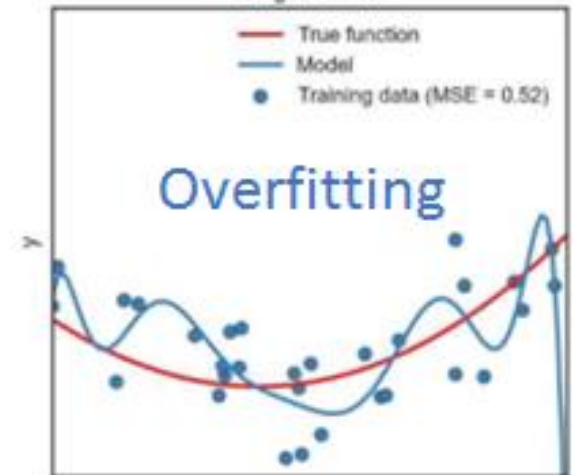
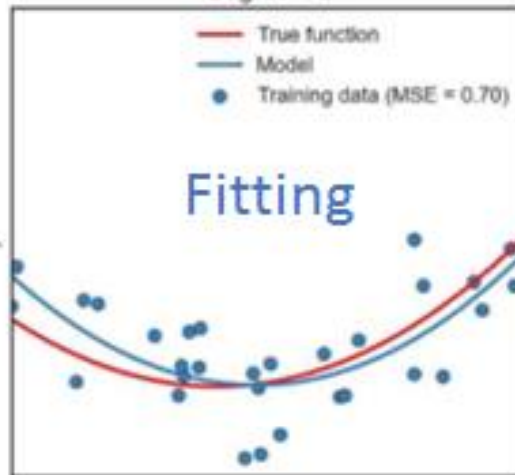
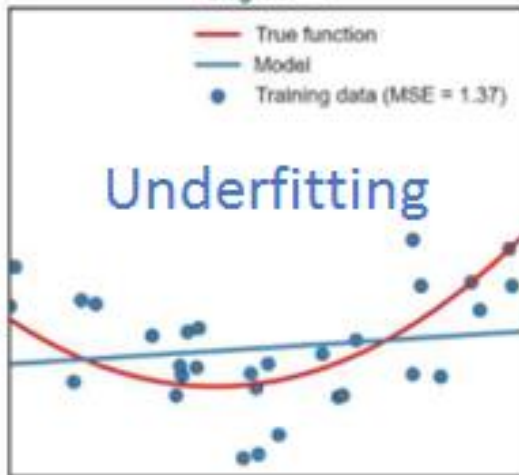
— True function
— Model
• Training data (MSE = 0.70)

Fitting

Degree = 10

— True function
— Model
• Training data (MSE = 0.52)

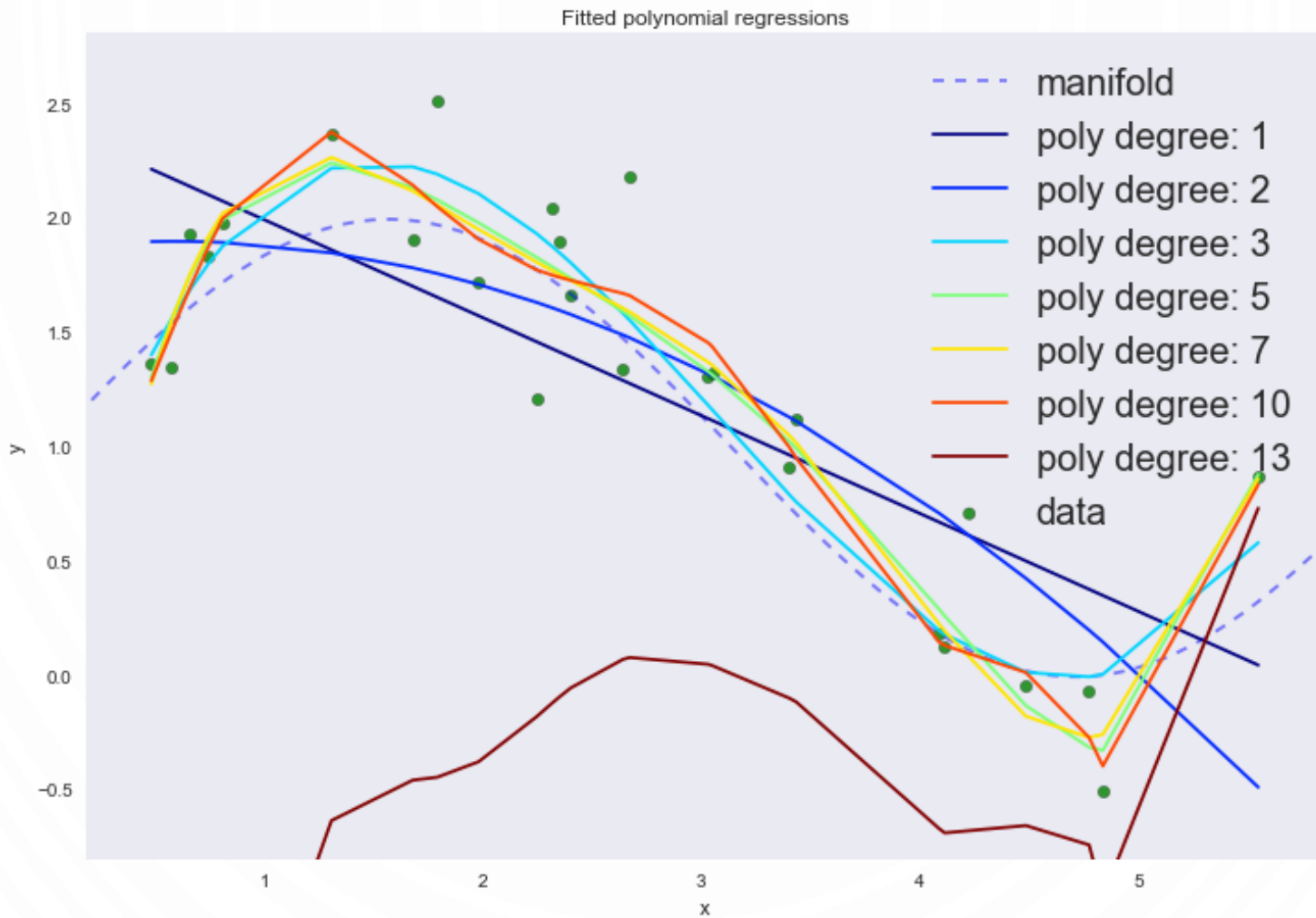
Overfitting



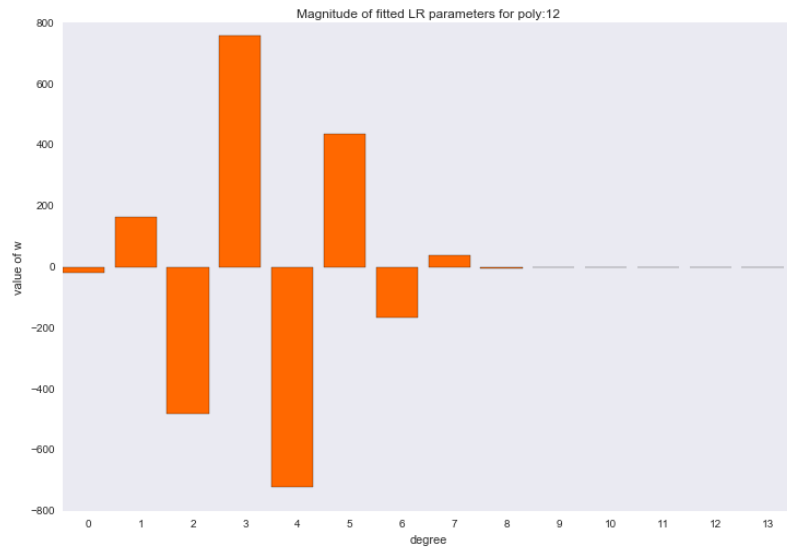
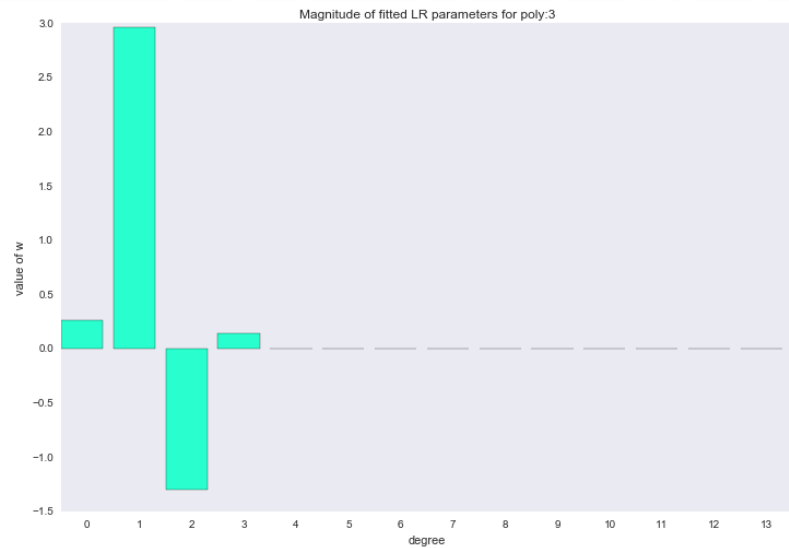
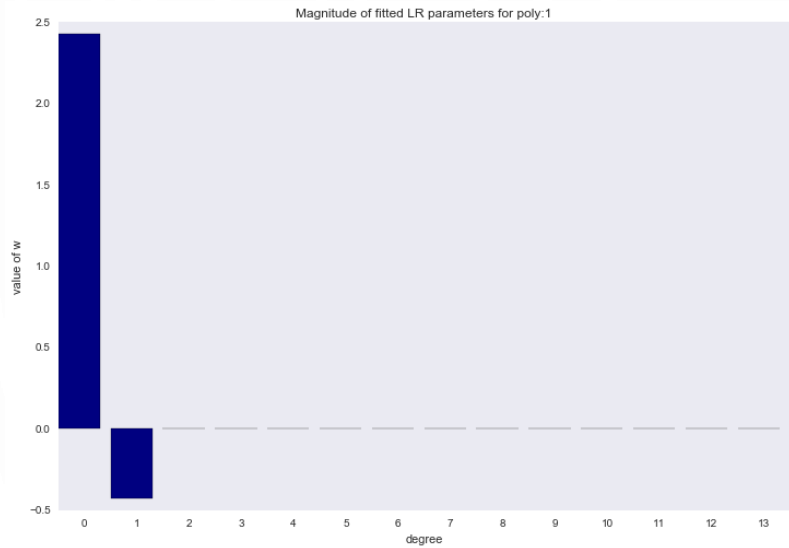
ПРИЗНАКИ ПЕРЕОБУЧЕННОЙ МОДЕЛИ

- Большая разница в качестве на тренировочных и тестовых данных (модель подгоняется под тренировочные данные и не может найти истинную зависимость)
- Большие значения параметров (весов) w_j модели
- Неустойчивость дискриминантной (разделяющей) функции (w, x) .

ПЕРЕОБУЧЕНИЕ: ПРИМЕР



ПЕРЕОБУЧЕНИЕ: ПРИМЕР



ОЦЕНИВАНИЕ КАЧЕСТВА МОДЕЛИ

- Отложенная выборка
- Кросс-валидация

ОТЛОЖЕННАЯ ВЫБОРКА

Делим тренировочную выборку на две части:

- По первой части обучаем модель (train)
- По оставшимся данным – оцениваем качество (test)

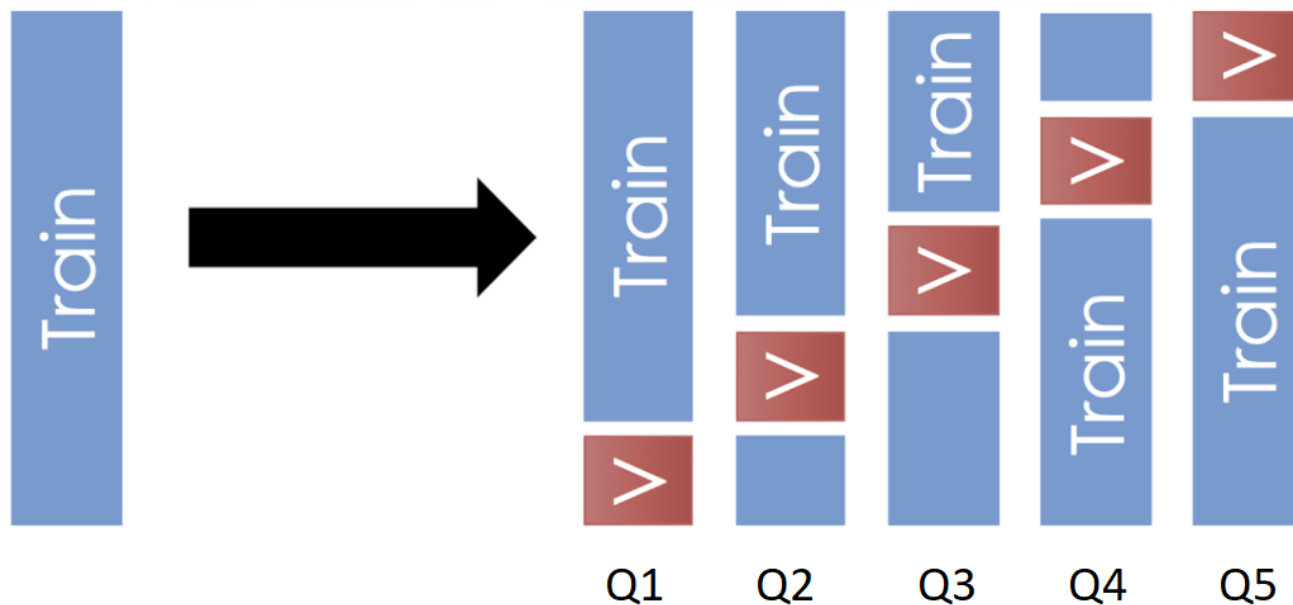


Недостаток:

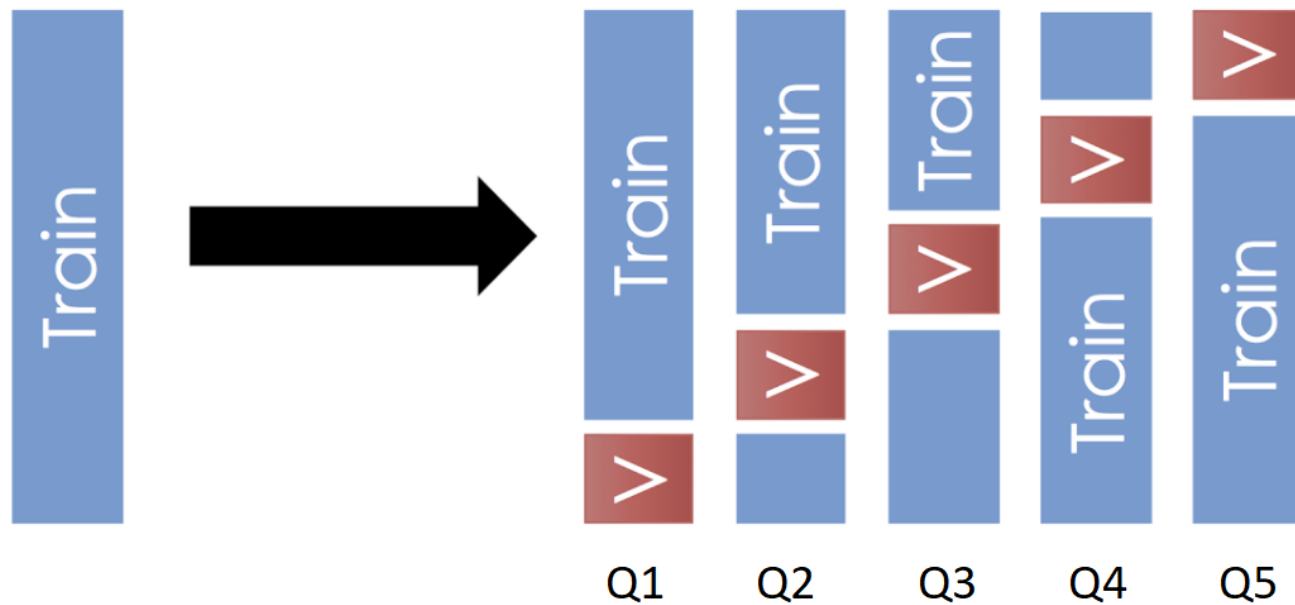
- Результат сильно зависит от разбиения на train и test

КРОСС-ВАЛИДАЦИЯ

- Разбиваем объекты на тренировку (train) и валидацию (validation) несколько раз (при разбиении k раз получаем k -fold кросс-валидацию)
- Для каждого разбиения вычисляем качество на валидационной части
- Усредняем полученные результаты



КРОСС-ВАЛИДАЦИЯ



$$CV = \frac{1}{k} \sum_{i=1}^k Q(a_i(x), X_i) = \frac{1}{k} \sum_{i=1}^k Q_i$$

ВИДЫ КРОСС-ВАЛИДАЦИИ

- **k-fold cross-validation** – разбиваем данные на k блоков, каждый из которых по очереди становится контрольным (валидационным)
- **Complete cross-validation** – перебираем ВСЕ разбиения
- **Leave-one-out cross-validation** – каждый блок состоит из одного объекта (число блоков = числу объектов)

ВЫБОР КОЛИЧЕСТВА БЛОКОВ В K-FOLD КРОСС-ВАЛИДАЦИИ



- Маленькое k – оценка может быть пессимистично занижена из-за маленького размера тренировочной части
- Большое k – оценка может быть неустойчивой из-за маленького размера валидационной части