Машинное обучение

Лекция 3

Линейная регрессия. Градиентный спуск.

Михаил Гущин

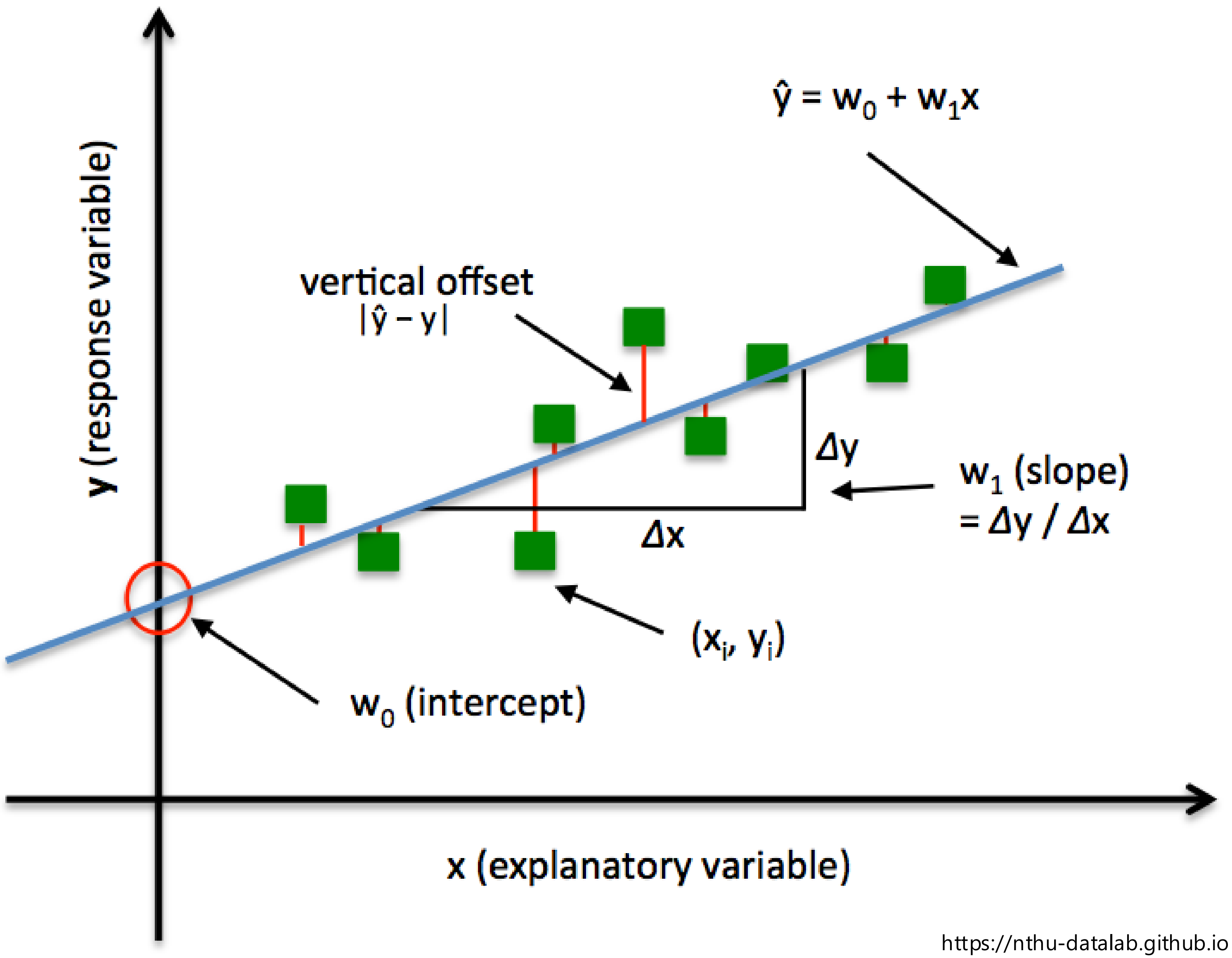
mhushchyn@hse.ru

НИУ ВШЭ, 2025

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Линейная регрессия. Аналитическое решение.

Линейная регрессия



https://nthu-datalab.github.io

Матричная форма

- Модель линейной регрессии:

$$\hat{y} = Xw$$

  - $X = \begin{pmatrix} \mathbf{1} & x_{11} & \cdots & x_{1d} \\ \vdots & & \ddots & \vdots \\ \mathbf{1} & x_{n1} & \cdots & x_{nd} \end{pmatrix}$ - матрица признаков объектов $(n, d+1)$;
  - $w = (w_0, w_1, \ldots, w_d)^T$ - вектор $(d+1)$ весов модели;
  - $\hat{y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)^T$ - вектор прогнозов модели для (n) объектов;

- Вектор квадратов ошибок прогнозов модели: $(\hat{y} - y)^2$

Аналитическое решение

$$L = (\hat{y} - y)^T(\hat{y} - y) = (Xw - y)^T(Xw - y)$$

Чтобы найти минимум L, надо:

$$\frac{\partial L}{\partial w} = 0$$

Тогда

$$\frac{\partial L}{\partial w} = 2X^T(Xw - y) = 2X^TXw - 2X^Ty = 0$$

Получаем оптимальные веса w линейной регрессии:

$$\boldsymbol{w = (X^TX)^{-1}X^Ty}$$

Шпаргалка по дифференцированию матриц: http://nabatchikov.com/blog/view/matrix_der

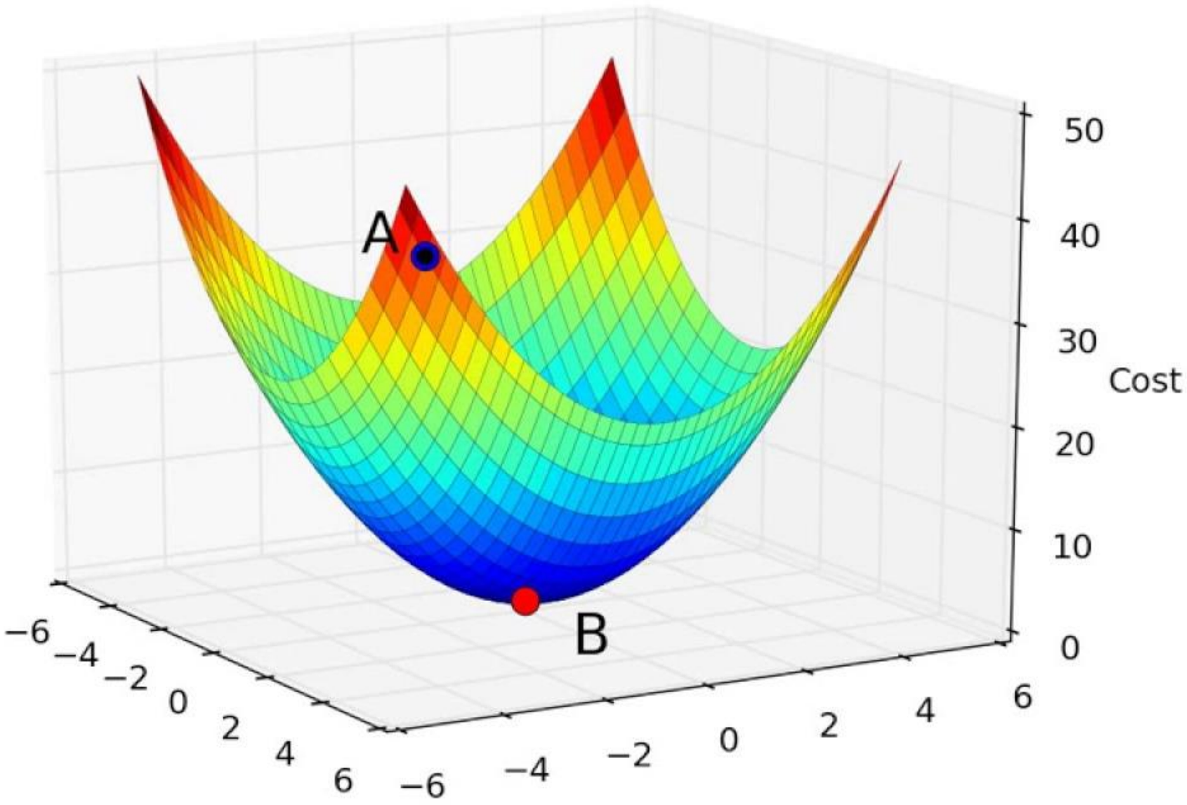
Линейная регрессия. Градиентный спуск.

Задача

- Есть функция $L(w)$
- Хотим найти ее минимум:

$$L \to \min_{w}$$

- Мы умеем считать ее производную $\frac{\partial L}{\partial w}$
- Но не умеем (или не хотим) решать уравнение $\frac{\partial L}{\partial w} = 0$



Градиент функции

- Градиент функции (∇L) – вектор первых частный производных функции:

$$\nabla L(w_0, w_1, \dots, w_d) = \left(\frac{\partial L}{\partial w_0}, \frac{\partial L}{\partial w_1}, \dots, \frac{\partial L}{\partial w_d}\right)$$

- В векторной форме мы будем писать:

$$\nabla L(w) = \frac{\partial L(w)}{\partial w}$$

Градиентный спуск

- Есть функция $L(w)$, минимум которой хотим найти
- Пусть $w^{(0)}$ - начальный вектор параметров. Например, $w^{(0)} = 0$
- Тогда **градиентный спуск** состоит в повторении:

$$w^{(k+1)} = w^{(k)} - \boldsymbol{\eta} \nabla L(w^{(k)})$$

  – $\boldsymbol{\eta}$ – длина шага градиентного спуска (**learning rate**) (мы сами его задаем)
  – k – номер итерации
  – $\nabla L(w^{(k)})$ – градиент функции потерь на итерации k

Пример

- Модель линейной регрессии:

$$\hat{y} = Xw$$

- Функция потерь MSE:

$$L = \frac{1}{n}(\hat{y} - y)^T(\hat{y} - y) = \frac{1}{n}(Xw - y)^T(Xw - y)$$

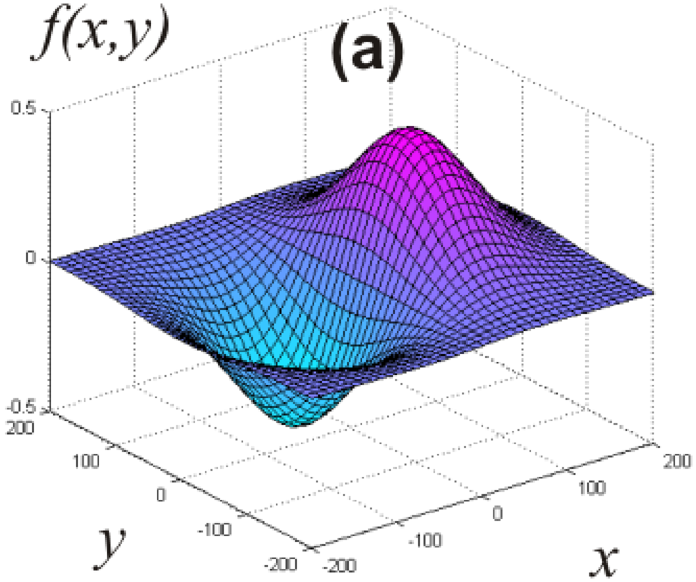
- Градиент:

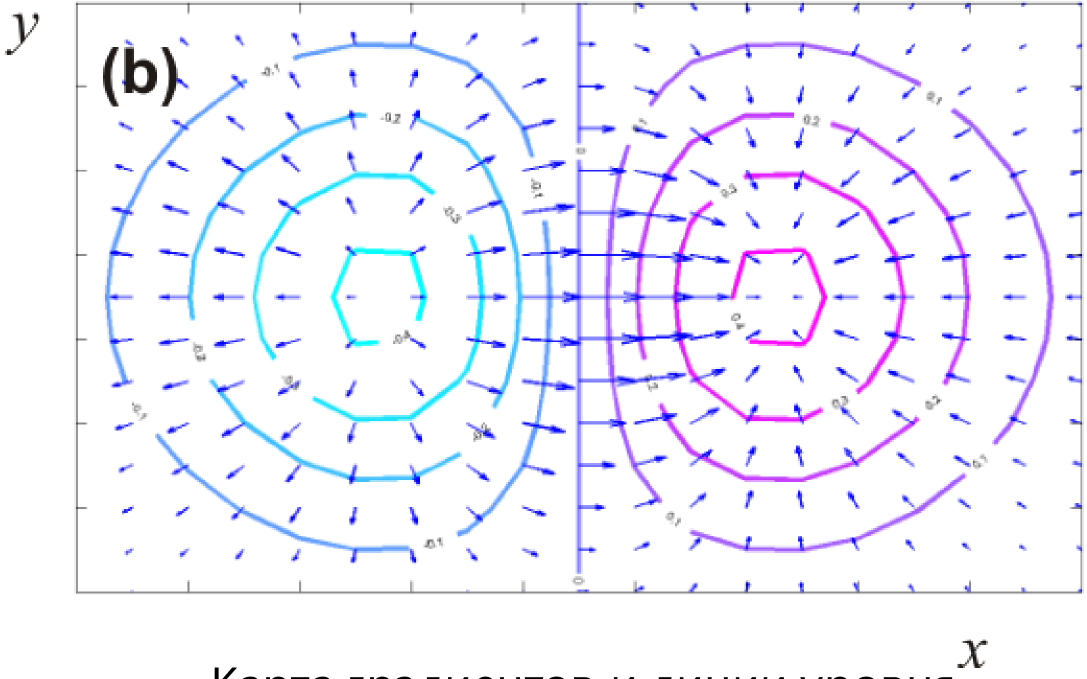
$$\nabla L(w) = \frac{2}{n}X^T(Xw - y)$$

- Градиентный спуск:

$$w^{(k+1)} = w^{(k)} - \boldsymbol{\eta}\nabla L\left(w^{(k)}\right) = w^{(k)} - \boldsymbol{\eta}\frac{2}{n}X^T\left(Xw^{(k)} - y\right)$$

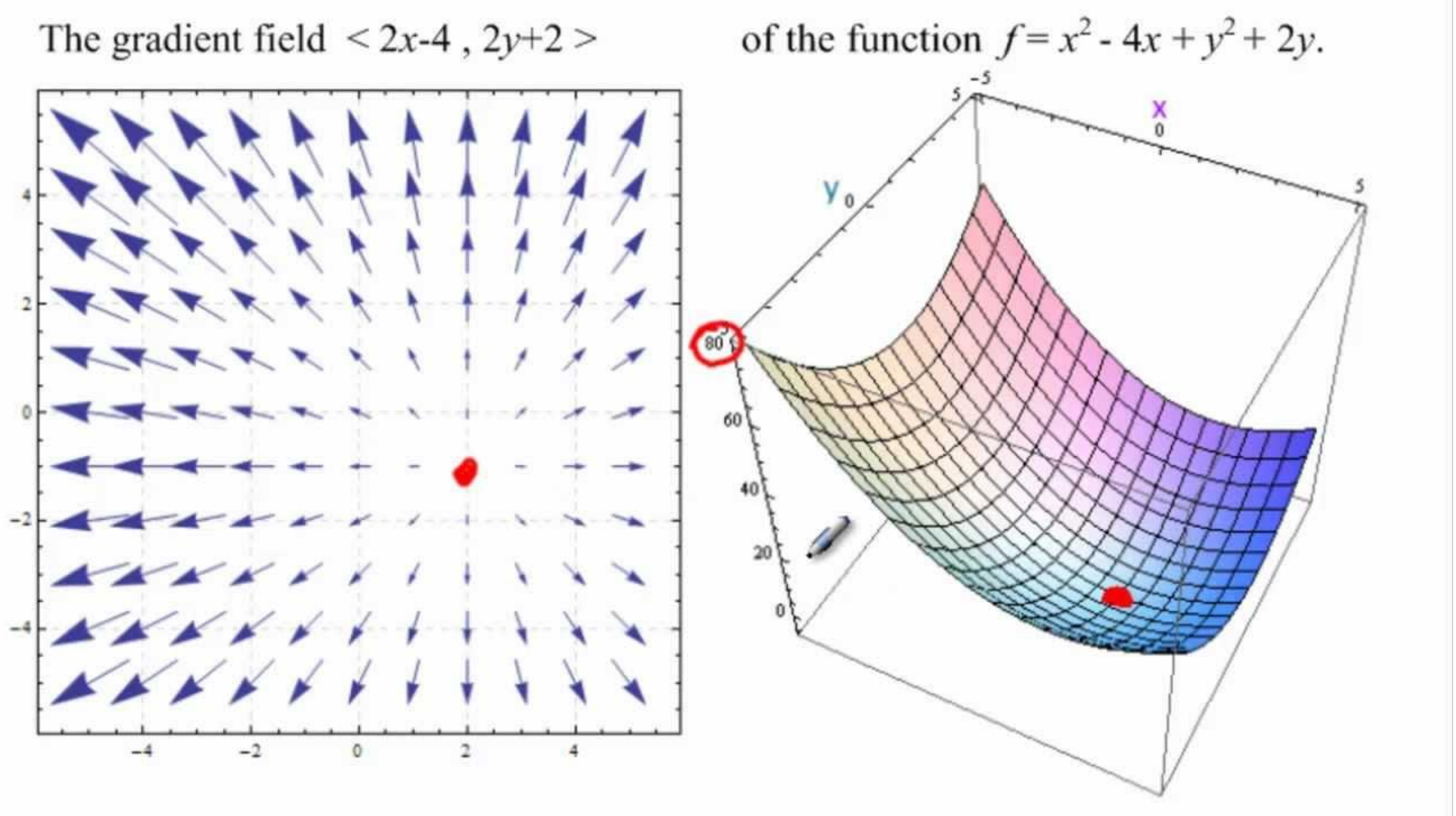
Свойства градиента





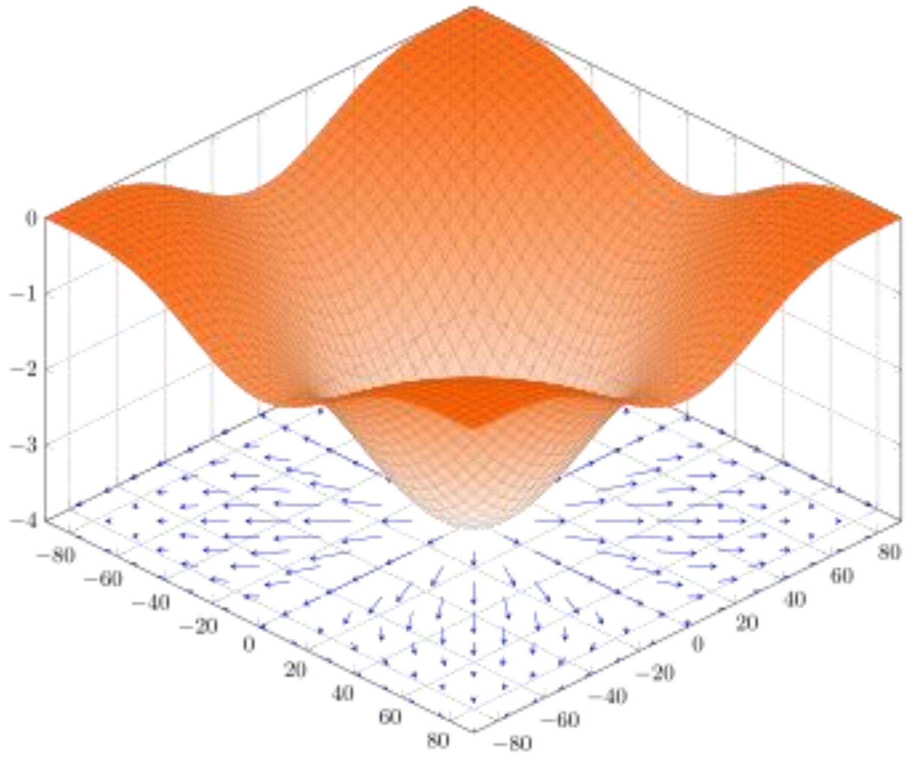
Карта градиентов и линии уровня функции

Свойства градиента



Свойства градиента

- Градиент функции в некоторой точке ортогонален линии уровня, проходящей через эту точку
- Градиент функции указывает направление наискорейшего возрастания функции в данной точке
- Направление антиградиента указывает направление наискорейшего убывания функции в данной точке

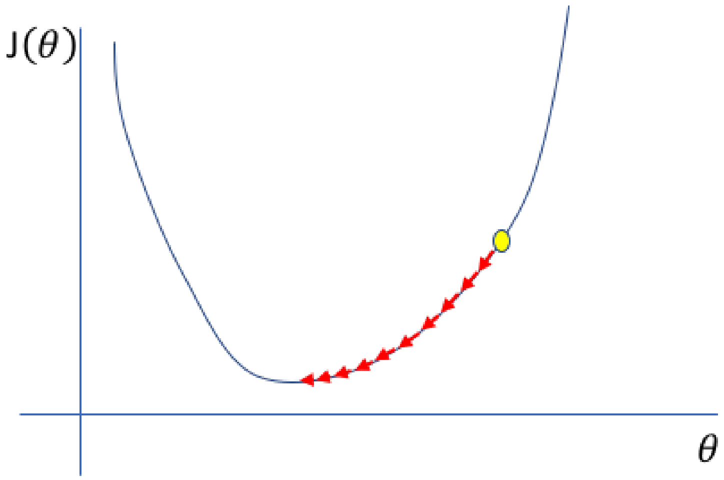


Градиентный спуск

- Как выбрать длину градиентного спуска?
- Сколько итераций делать?

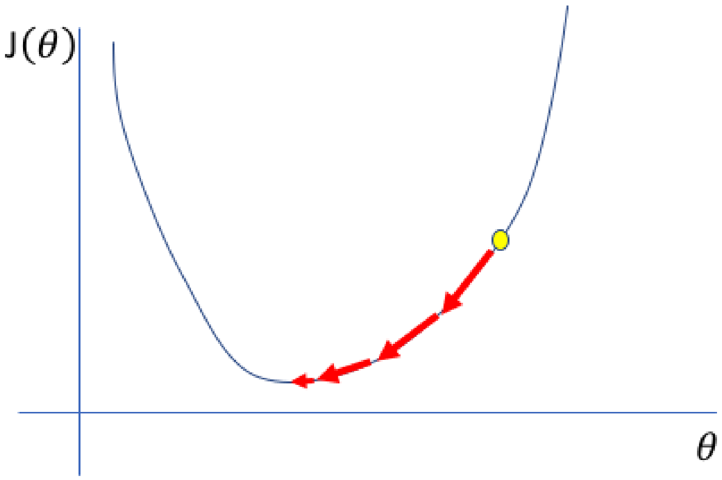
Выбор шага

Too low



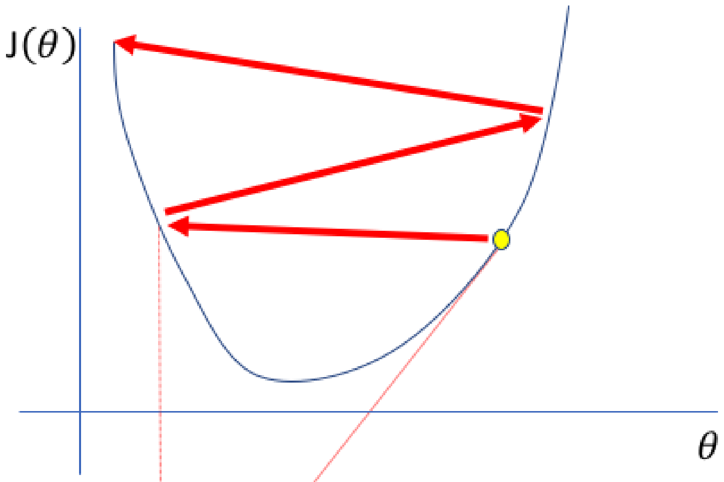
A small learning rate requires many updates before reaching the minimum point

Just right



The optimal learning rate swiftly reaches the minimum point

Too high



Too large of a learning rate causes drastic updates which lead to divergent behaviors

Выбор шага

- Константа: $\eta = const$
- Уменьшение с каждой итерацией k: $\eta_k = \frac{1}{k}$
- Другие варианты: $\eta_k = \lambda \left(\frac{s_0}{s_0+k}\right)^p$
  - λ, s_0, p – некоторые значения
  - как правило $s_0 = 1, p = 0.5$

Критерии остановки

- Близость градиента к нулю: $\nabla L \approx 0$

- Малое изменение вектора весов: $|w^{(k+1)} - w^{(k)}| \approx 0$

Стохастический градиентный спуск

- Модель линейной регрессии:

$$\hat{y} = Xw$$

- Функция потерь MSE:

$$L(w) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

- Мы хотим минимизировать L:

$$L \to \min_{w}$$

- Градиентный спуск:

$$w^{(k+1)} = w^{(k)} - \eta \nabla L(w^{(k)})$$

Стохастический градиентный спуск

- Полный градиентный спуск:

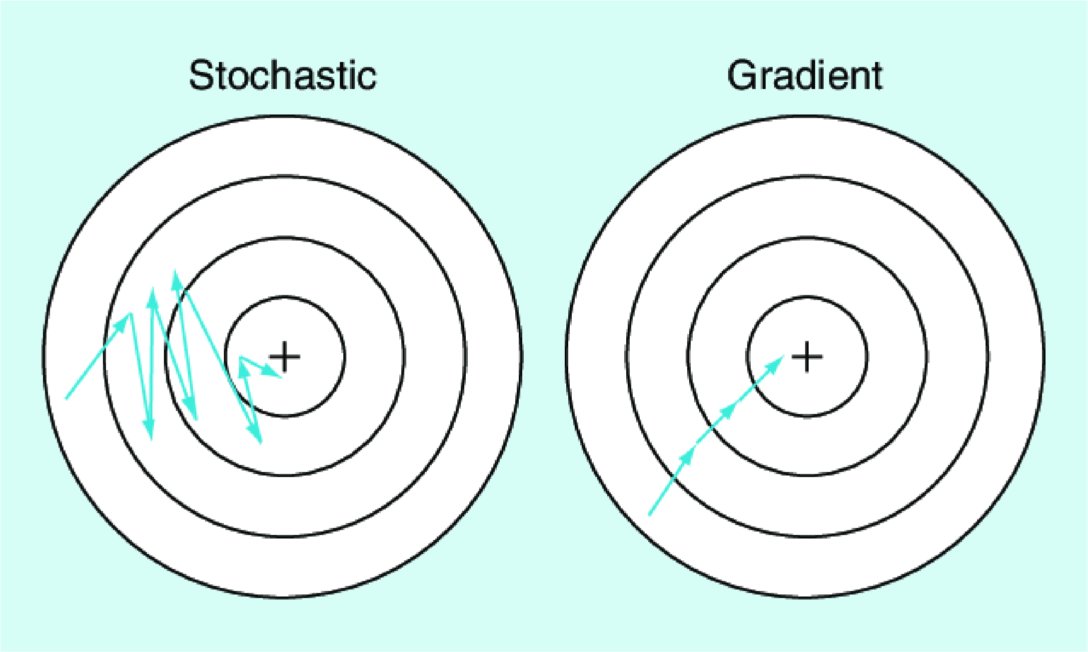
$$L(w) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

$$w^{(k+1)} = w^{(k)} - \eta\nabla L(w^{(k)})$$

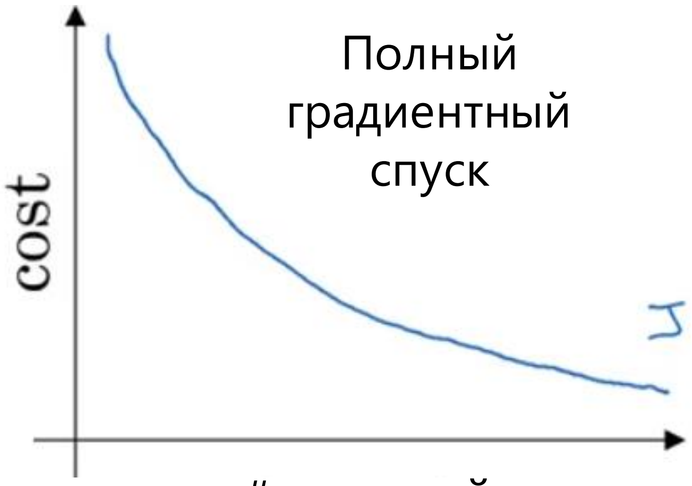
- **Стохастический** градиентный спуск:

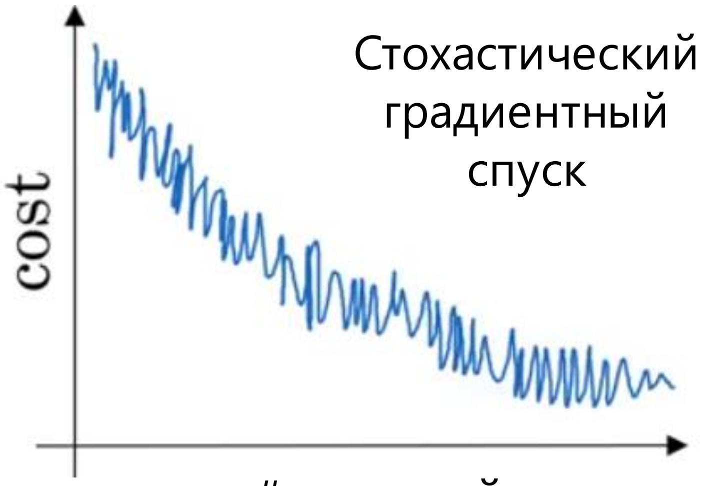
$$L_i(w) = (\hat{y}_i - y_i)^2$$

$$w^{(k+1)} = w^{(k)} - \eta\nabla L_i(w^{(k)})$$



Стохастический градиентный спуск





- Стохастический ГС требует меньше вычислительный операций
- В полном ГС обучение стабильнее
- Полный ГС требует меньше итераций, но больше вычислительных операций

Нормализация данных

Нормализация данных

- Модель линейной регрессии:

$$\hat{y}_i = w_0 + w_1 x_{i1} + w_2 x_{i2}$$

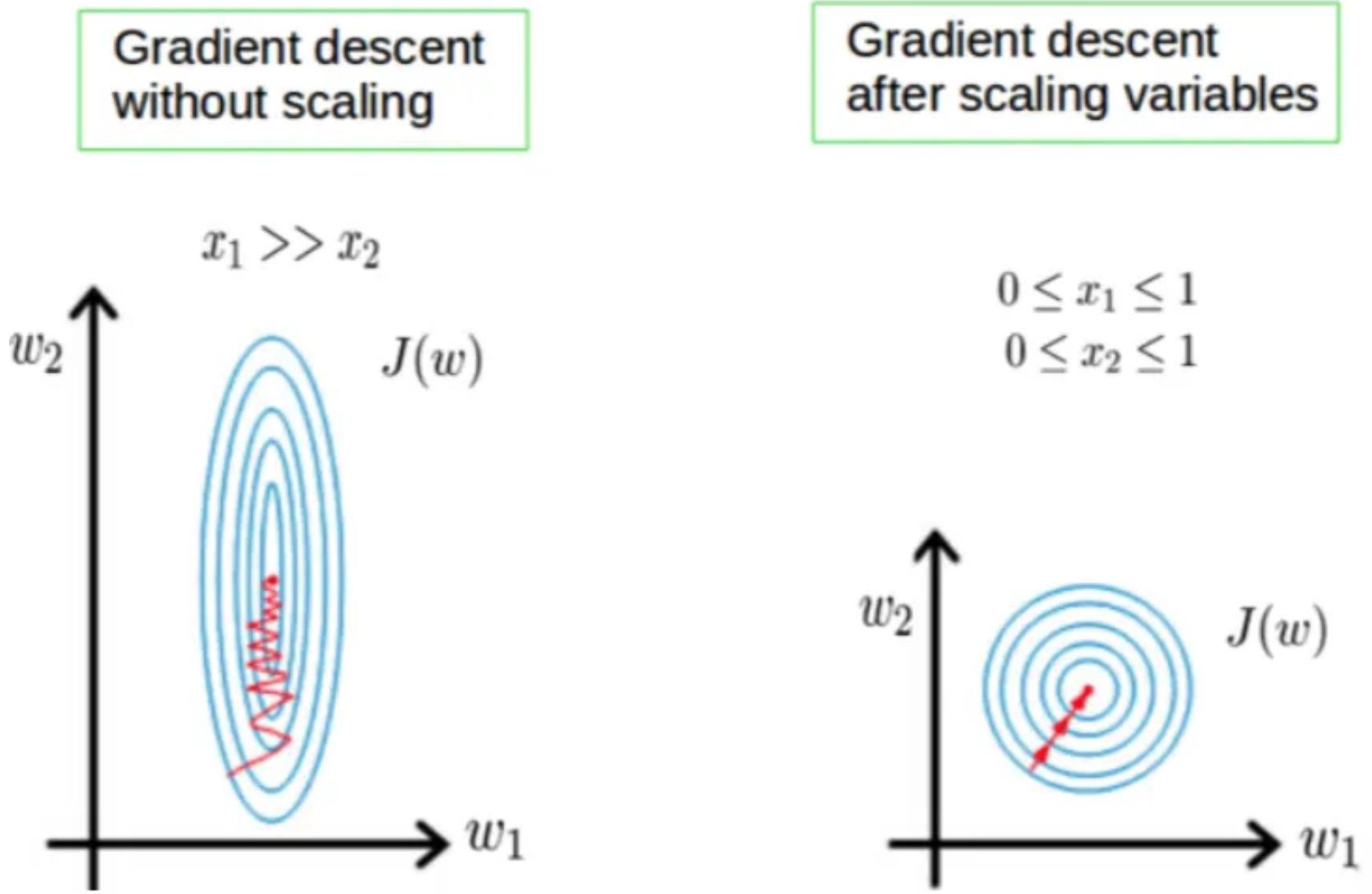
- Функция потерь MSE:

$$L(w) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \to \min_w$$

- Пусть признаки разные по масштабу:
  - например $|x_1| \approx 1000$, $|x_2| \approx 1$

- Тогда малые изменения dw_2 приводят к малым изменениям $L(w)$
- Малые изменения dw_1 приводят к большим изменениям $L(w)$

Нормализация данных



- Нормализация данных стабилизирует градиентный спуск
- Обучение происходит быстрее

Популярные способы нормализации

- Standard scaler:

$$x_i^{norm} = \frac{x_i - \mu}{\sigma}$$

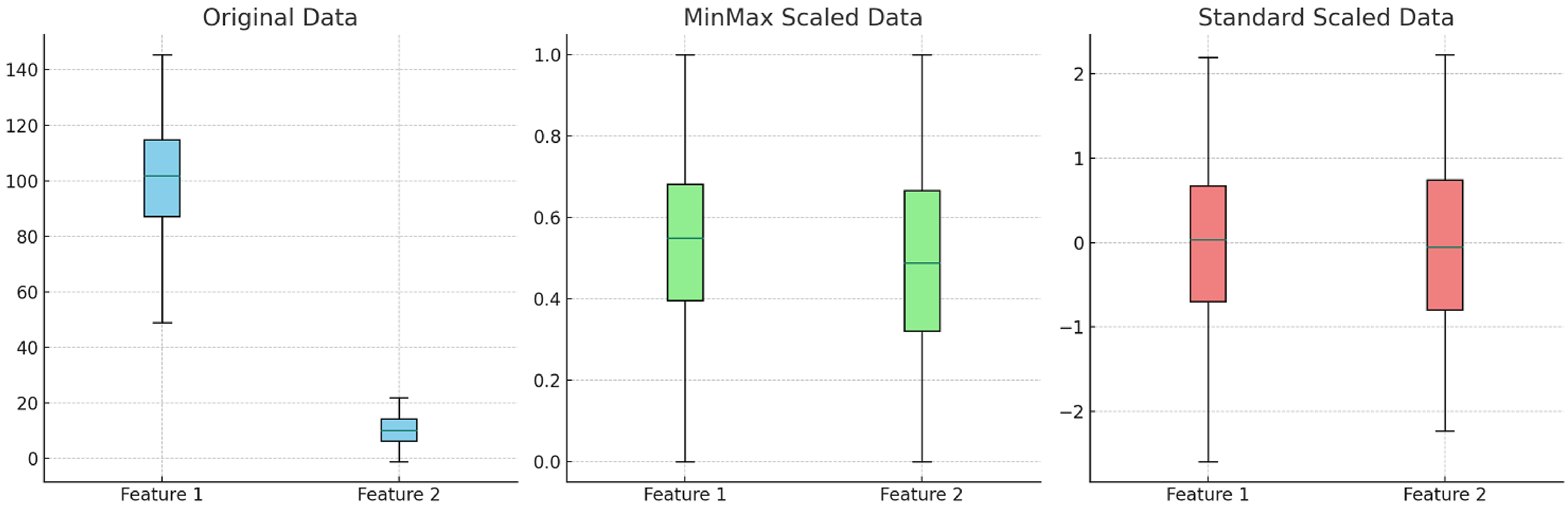
$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

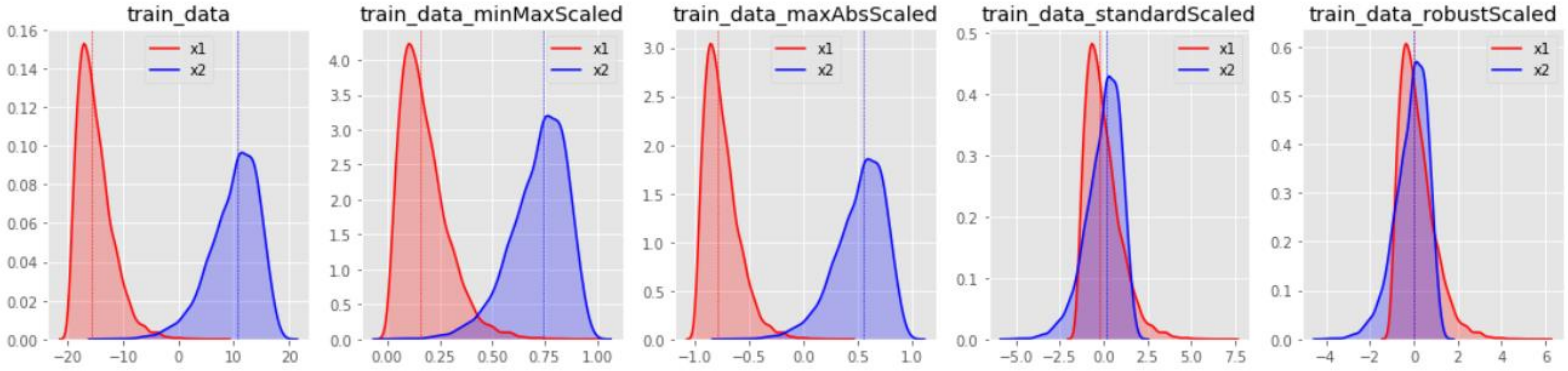
- Min-Mac scaler:

$$x_i^{norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

Популярные способы нормализации



Популярные способы нормализации



Переобучение

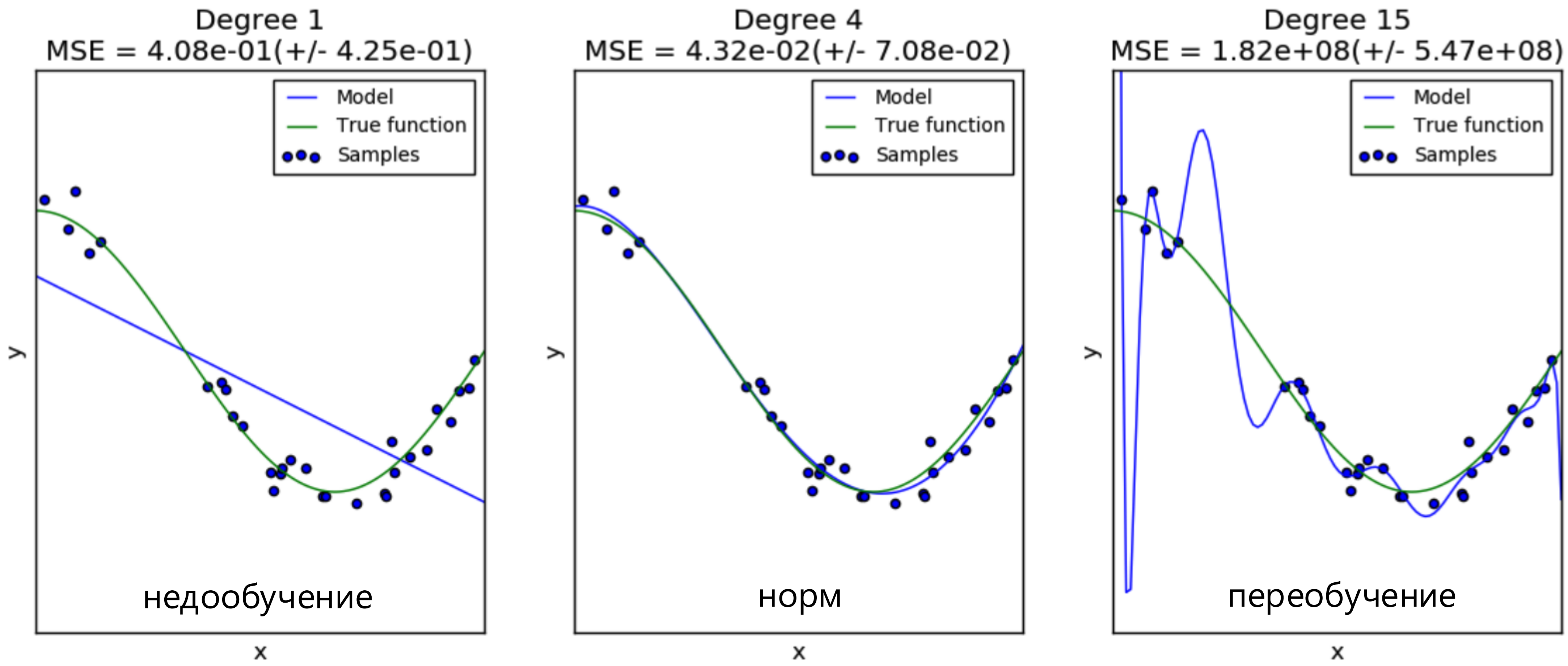
Задача

- Пусть дан набор из n точек: $\{x_i, y_i\}_{i=1}^{n}$, где $x_i \in \mathbb{R}^1$
- Для каждого x_i создадим дополнительные признаки:
  - $x_i, x_i^2, x_i^3, \dots, x_i^k$
- Рассмотрим модель полиномиальной линейной регрессии:

$$\hat{y}_i = w_0 + \sum_{j=1}^{\boldsymbol{k}} w_j x_i^j$$

- Максимальную степень полинома $\boldsymbol{k}$ будем менять от 1 до 15

Решение


Degree 1
MSE = 4.08e-01(+/- 4.25e-01)
Model
True function
Samples
y
x
недообучение
Degree 4
MSE = 4.32e-02(+/- 7.08e-02)
Model
True function
Samples
y
x
норм
Degree 15
MSE = 1.82e+08(+/- 5.47e+08)
Model
True function
Samples
y
x
переобучение

Регуляризация

Проблема переобучения

- Модель линейной регрессии:

$$\hat{y}_i = w_0 + \sum_{j=1}^{d} w_j x_{ij}$$

- Ошибка прогноза модели для объекта: $|\hat{y}_i - y_i|$
- Пусть значение некоторых весов очень большие по модулю, например $|w_k| > 10^3$
- Тогда малые изменения dx_{ik} приводят к очень большим изменениям $|\mathrm{d}\hat{y}_i| = |w_k dx_{ik}|$

Регуляризация

- Давайте добавим к функции потерь $L(w)$ **штраф** $R(w)$ на **величину весов** модели:

$$L_\alpha(w) = L(w) + \boldsymbol{\alpha} R(w)$$

  – $\boldsymbol{\alpha}$ – коэффициент регуляризации (подбираем сами)

- Регуляризация не позволяет весам модели принимать слишком большие значения

Виды регуляризации

- L_1 регуляризация (Lasso):

$$R_1(w) = \sum_{j=1}^{d} |w_j|$$

- L_2 регуляризация (Ridge):

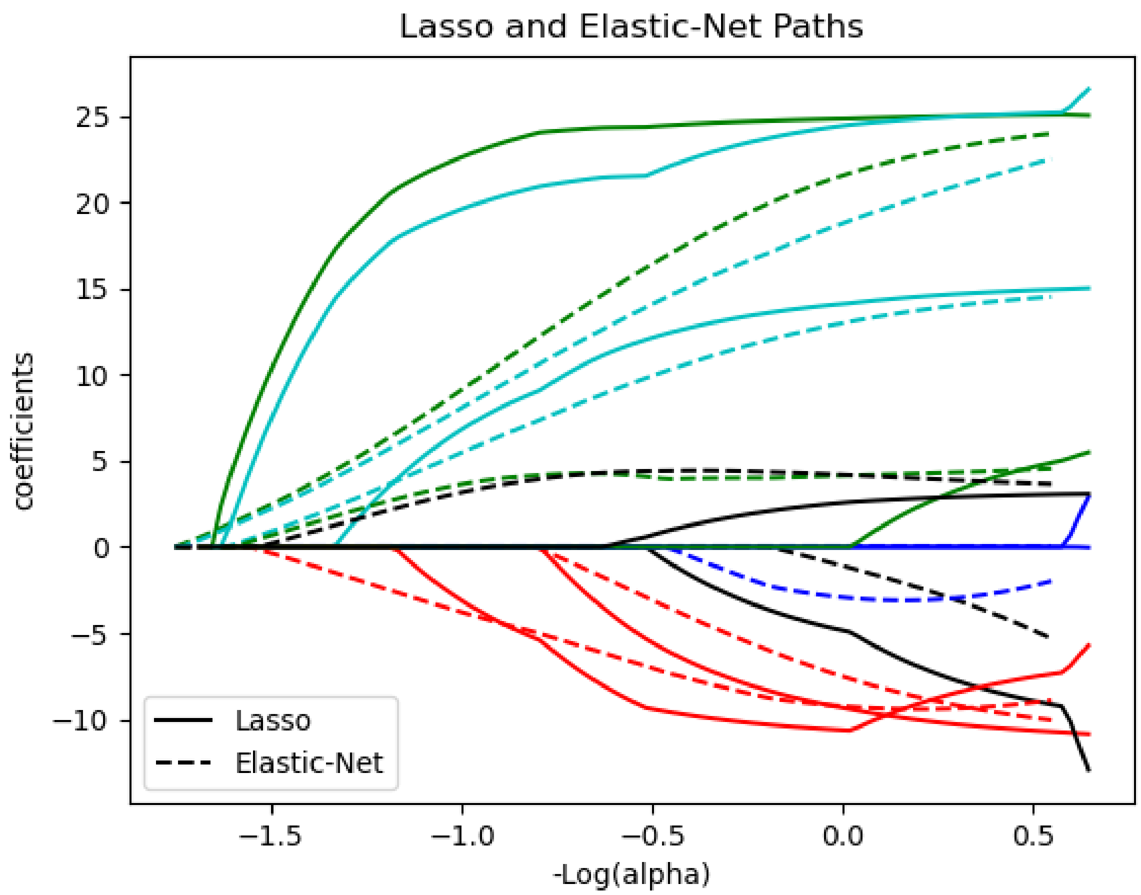
$$R_2(w) = \sum_{j=1}^{d} w_j^2$$

- $L_1 + \mathrm{L}_2$ регуляризация (Elastic Net):

$$L_\alpha(w) = L(w) + \alpha_1 R_1(w) + \alpha_2 R_2(w)$$

Свойства регуляризации

- L_2 регуляризация стремится уменьшить веса модели
- L_1 позволяет проводить **отбор признаков**
- $\boldsymbol{L_1}$ **обнуляет веса** для наименее информативных признаков



https://scikit-learn.org

Гиперпараметры и параметры

- Рассмотрим пример функции потерь с регуляризацией:

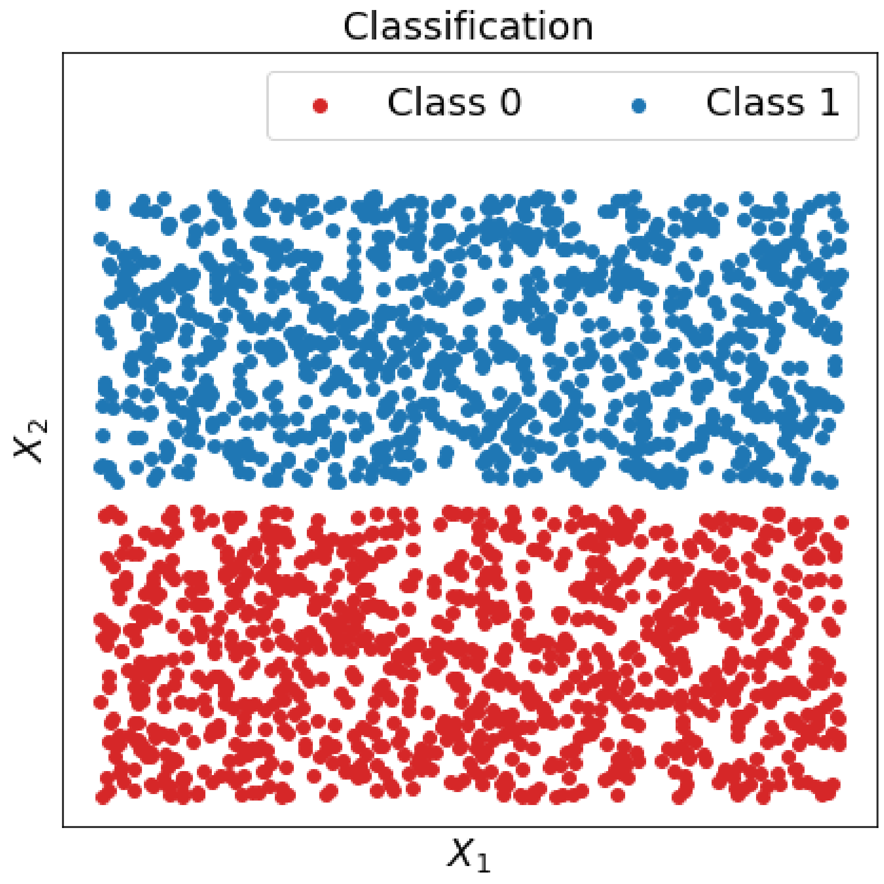
$$L_\alpha(w) = L(w) + \boldsymbol{\alpha} R(w)$$

- Здесь w – веса нашей модели. Их будем называть **параметрами** модели. Они определяются в процессе обучения.
- $\boldsymbol{\alpha}$ – коэффициент регуляризации. Его значение задаем мы сами. Такие параметры будем называть **гиперпараметрами**.

Важность признаков (Feature importance)

Интуиция

- Не все признаки одинаково полезны для решения задачи
- Некоторые из них более информативны, чем другие
- Например, X_1 неинформативна для классификации
- **Цель** – определить **важность каждого признака**



Линейные модели

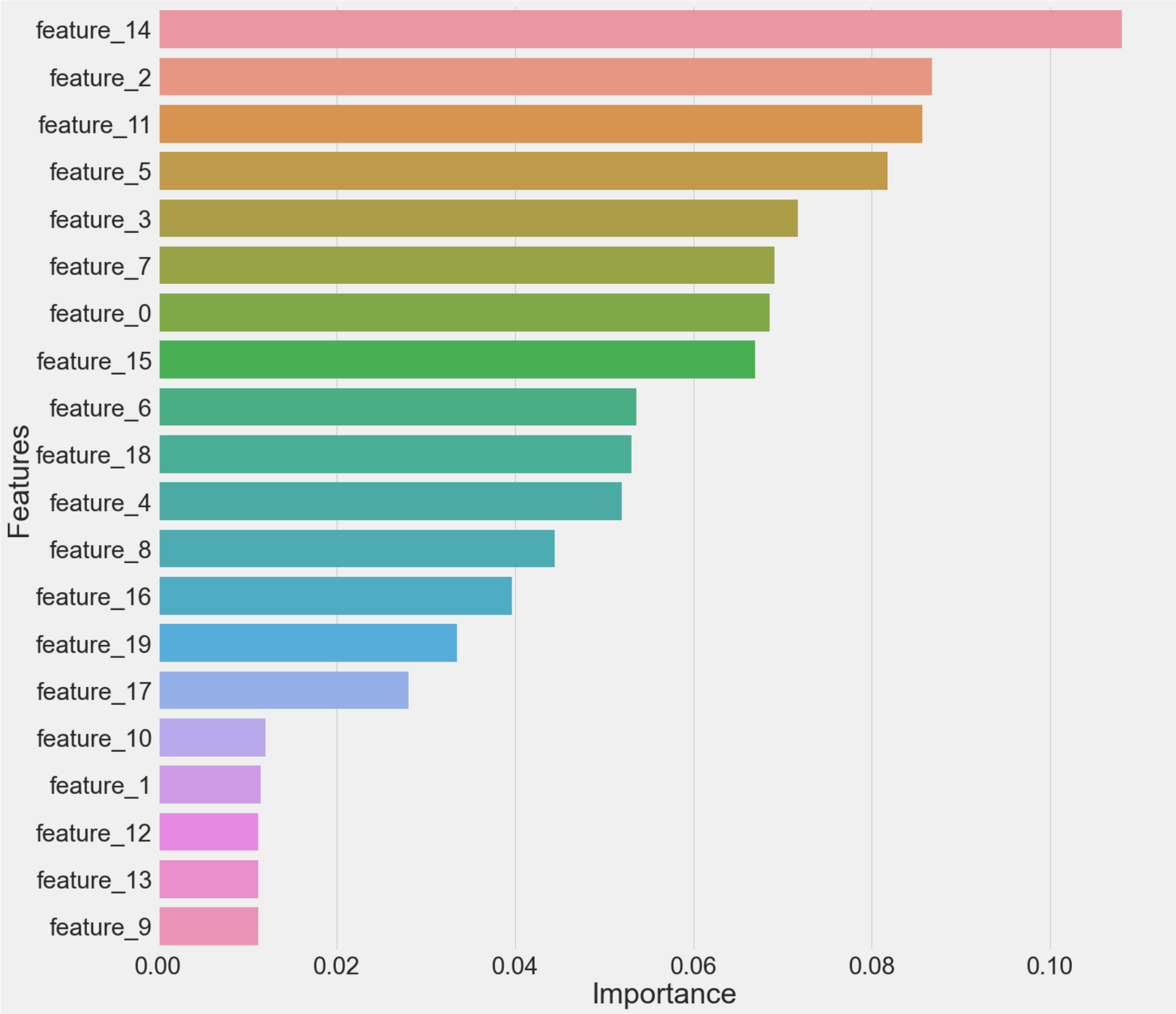
Рассмотрим линейную модель с регуляризацией (L_1 или L_2):

$$\hat{y} = w_0 + w_1 f_1 + w_2 f_2 + \cdots + w_k f_k$$

Если признаки нормированы (значения одного масштаба), то важность признака f_i равна:

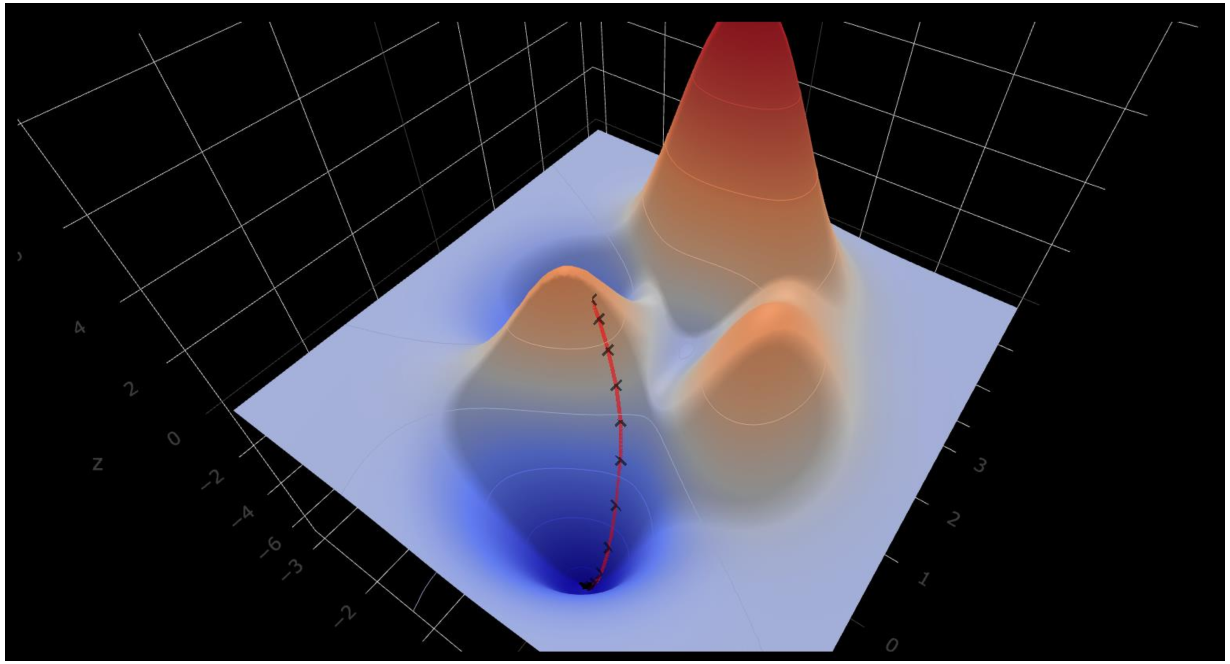
$$Imp(f_i) = |w_i|$$

Пример


feature_14
feature_2
feature_11
feature_5
feature_3
feature_7
feature_0
feature_15
feature_6
feature_18
feature_4
feature_8
feature_16
feature_19
feature_17
feature_10
feature_1
feature_12
feature_13
feature_9
Features
0.00
0.02
0.04
0.06
0.08
0.10
Importance

Демонстрации

- Линейная регрессия с градиентным спуском https://lukaszkujawa.github.io/gradient-descent.html
- Градиентный спуск на сложной функции: https://blog.skz.dev/gradient-descent



Заключение

Вопросы

- Что такое объект, целевая переменная, признак, модель, функция потерь, функционал ошибки и обучение?
- Что такое переобучение и недообучение? Как отличить переобучение от недообучения?
- Что такое кросс-валидация и для чего она используется? Чем применение кросс-валидации лучше, чем разбиение выборки на обучение и контроль?
- Чем гиперпараметры отличаются от параметров?
- Запишите формулы для линейной модели регрессии и для среднеквадратичной ошибки. Запишите среднеквадратичную ошибку в матричном виде.
- Что такое градиент? Какое его свойство используется при минимизации функций?
- Запишите алгоритм градиентного спуска. Приведите примеры критериев остановка. Как длина шага влияет на процесс оптимизации?
- Для чего нужно нормировать данные при обучении линейных моделей? Какие способы нормировки вы знаете?
- Что такое регуляризация? Для чего ее используют в линейных моделях? Запишите L1- и L2-регуляризаторы. Почему L1-регуляризация отбирает признаки?