Машинное обучение

Лекция 2

Линейная регрессия

Михаил Гущин

mhushchyn@hse.ru

НИУ ВШЭ, 2022

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

На прошлой лекции

- Задачи машинного обучения с учителем: данные $X \in \mathbb{R}^{n \times d}, y \in \mathbb{R}^n$
  - Регрессия: $y \in \mathbb{R}^n$
  - Классификация:
    - Бинарная: $y \in \{0, 1\}^n$
    - Многоклассовая: $y \in \{0, 1, 2, \dots, c\}^n$
  - Рекомендательные системы
- Задачи обучения без учителя: данные $X \in \mathbb{R}^{n \times d}$
  - Кластеризация: $X \in \mathbb{R}^{n \times d} \to Z \in \{0, 1, 2, \dots, c\}^n$
  - Понижение размерности: $X \in \mathbb{R}^{n \times d} \to Z \in \mathbb{R}^{n \times k}, k < d$

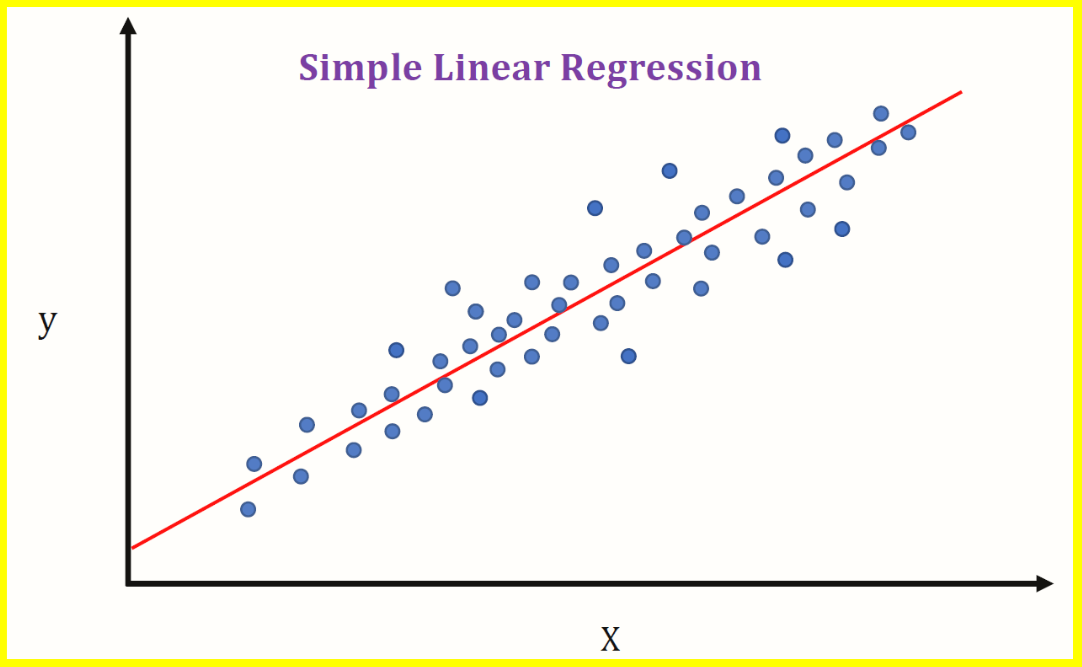
План

- Линейная регрессия
- Градиентный спуск
- Метрики качества
- Переобучение
- Регуляризация

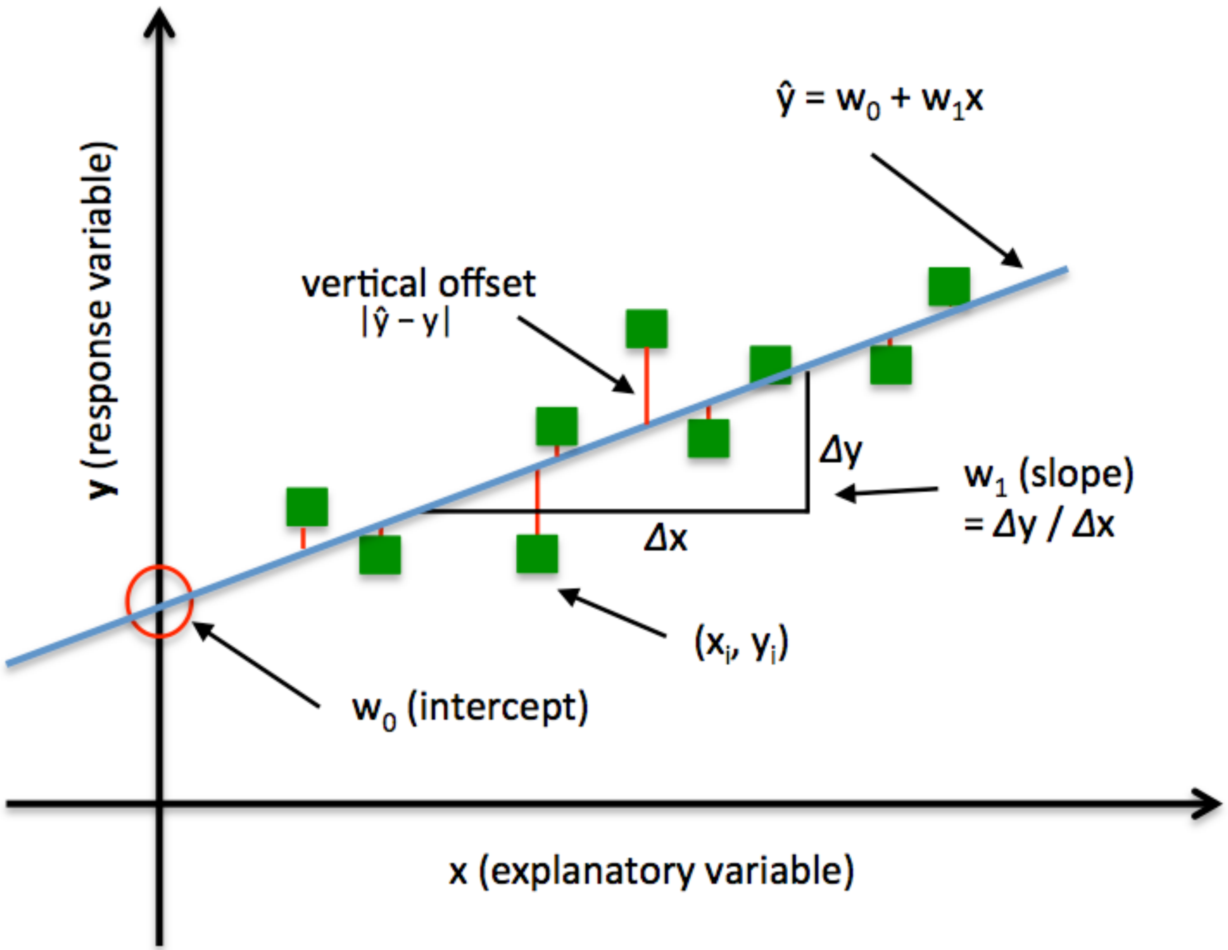
Линейная регрессия

Задача регрессии

- Есть объекты (X)
- Нужно предсказать некоторую величину (y)
- Функция, которая описывает зависимость y от X - **модель регрессии**



Линейная регрессия



https://nthu-datalab.github.io

Векторная форма

- Пусть дан набор из n точек: $\{x_i, y_i\}_{i=1}^n$, где
  - $x_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$ - вектор из d признаков объекта;
  - y_i - скалярная величина, которую хотим предсказать для объекта.
- Модель линейной регрессии:

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

  - w_j - веса модели;
  - $\hat{y}_i$ - прогноз для объекта;
- Ошибка прогноза модели для объекта: $|\hat{y}_i - y_i|$

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;

  – $w = (w_0, w_1, \dots, w_d)^T$ - вектор $(d + 1)$ весов модели;

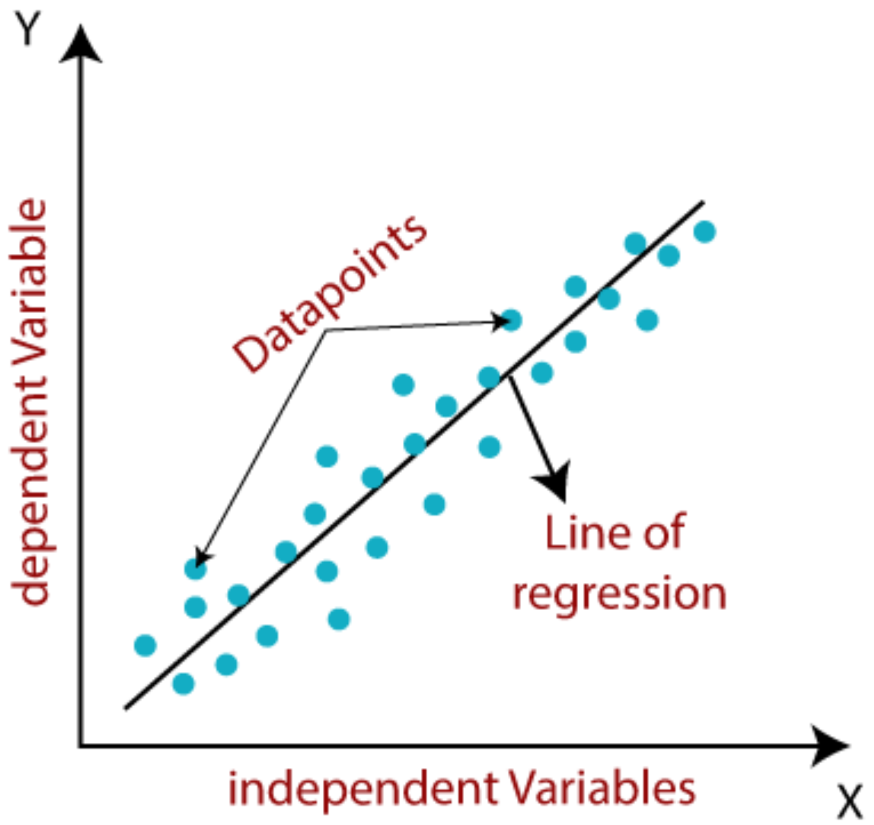
  – $\hat{y} = (\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n)^T$ - вектор прогнозов модели для (n) объектов;

- Вектор ошибок прогнозов модели: $|\hat{y} - y|$

Задача

- Хотим, чтобы средняя ошибка прогнозов $|\hat{y} - y|$ была минимальной

- **Как найти** оптимальные веса $\boldsymbol{w}$ модели?



Решение

- **Функция потерь (Loss function)** (скалярная и векторная формы):

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

- Значение L – среднеквадратичная ошибка (**Mean Squared Error** (**MSE**))
- Мы хотим минимизировать L:

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

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:

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

Повтор

- Модель линейной регрессии:

$$\hat{y} = Xw$$

- Функция потерь MSE:

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

- Мы хотим минимизировать L:

$$L \to \min_w$$

- Аналитическое решение:

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

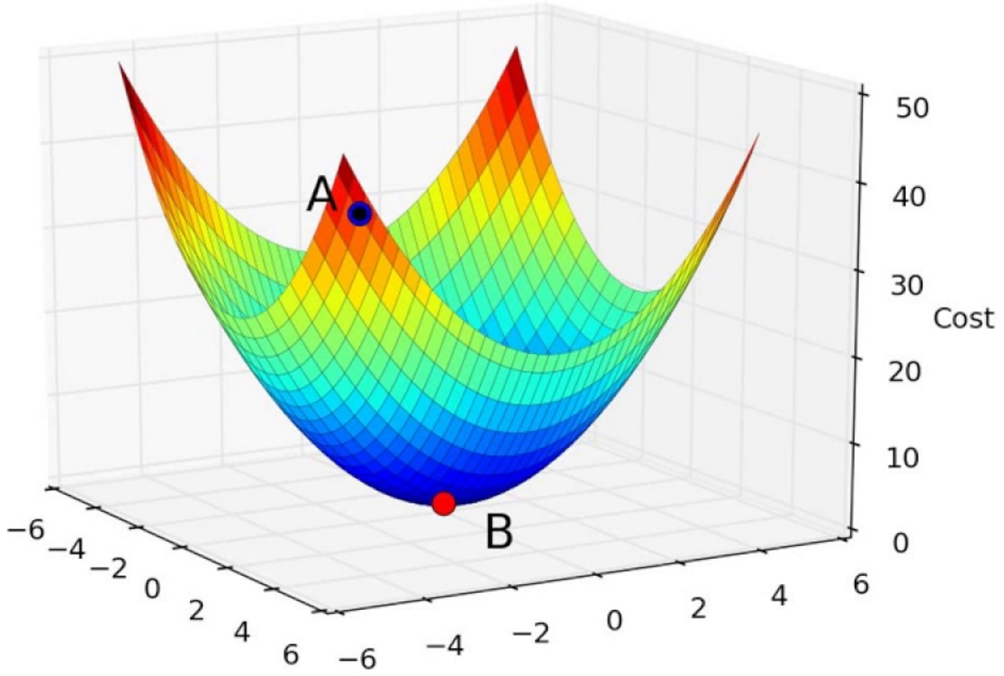
Градиентный спуск

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



Градиент функции

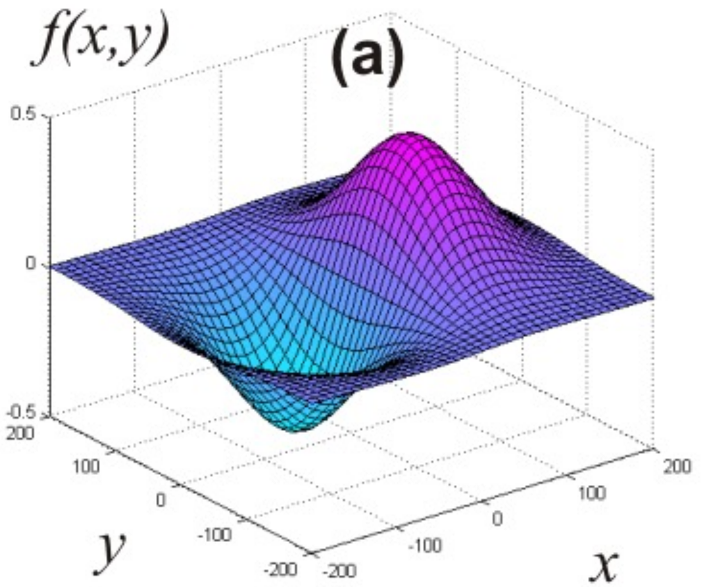
- Градиент функции (∇L) – вектор первых частный производных функции:

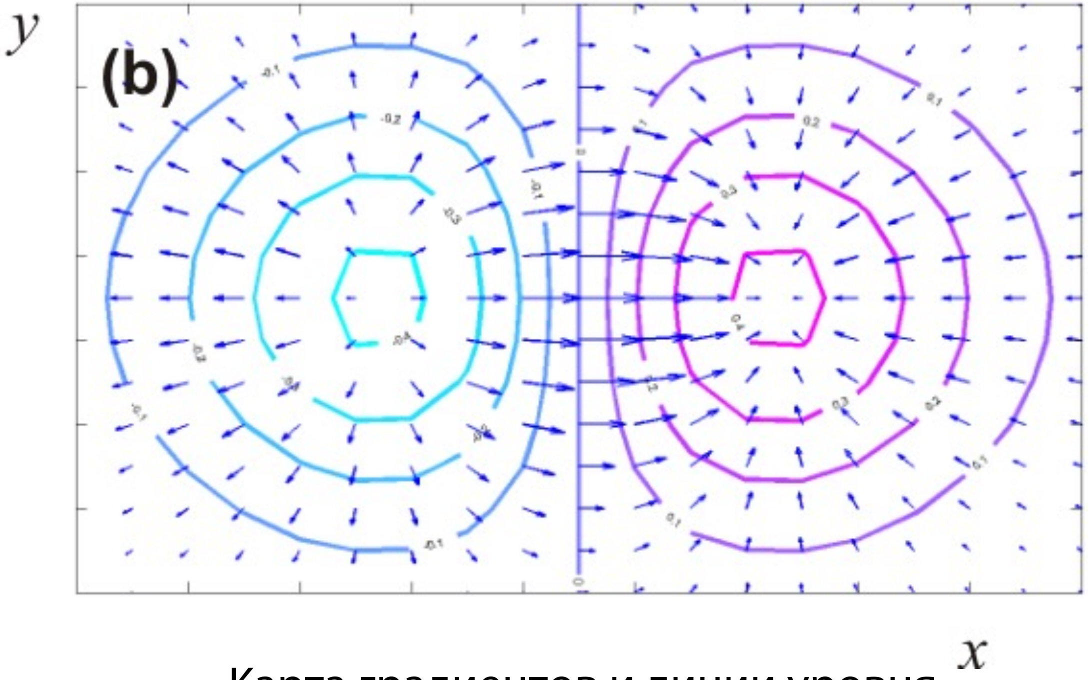
$$\nabla L(w_0, w_1, \dots, w_d) = \left(\frac{\partial L}{\partial w_0}, \frac{\partial L}{\partial w_1}, \dots, \frac{\partial L}{\partial w_d}\right)$$

- В векторной форме мы будем писать:

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

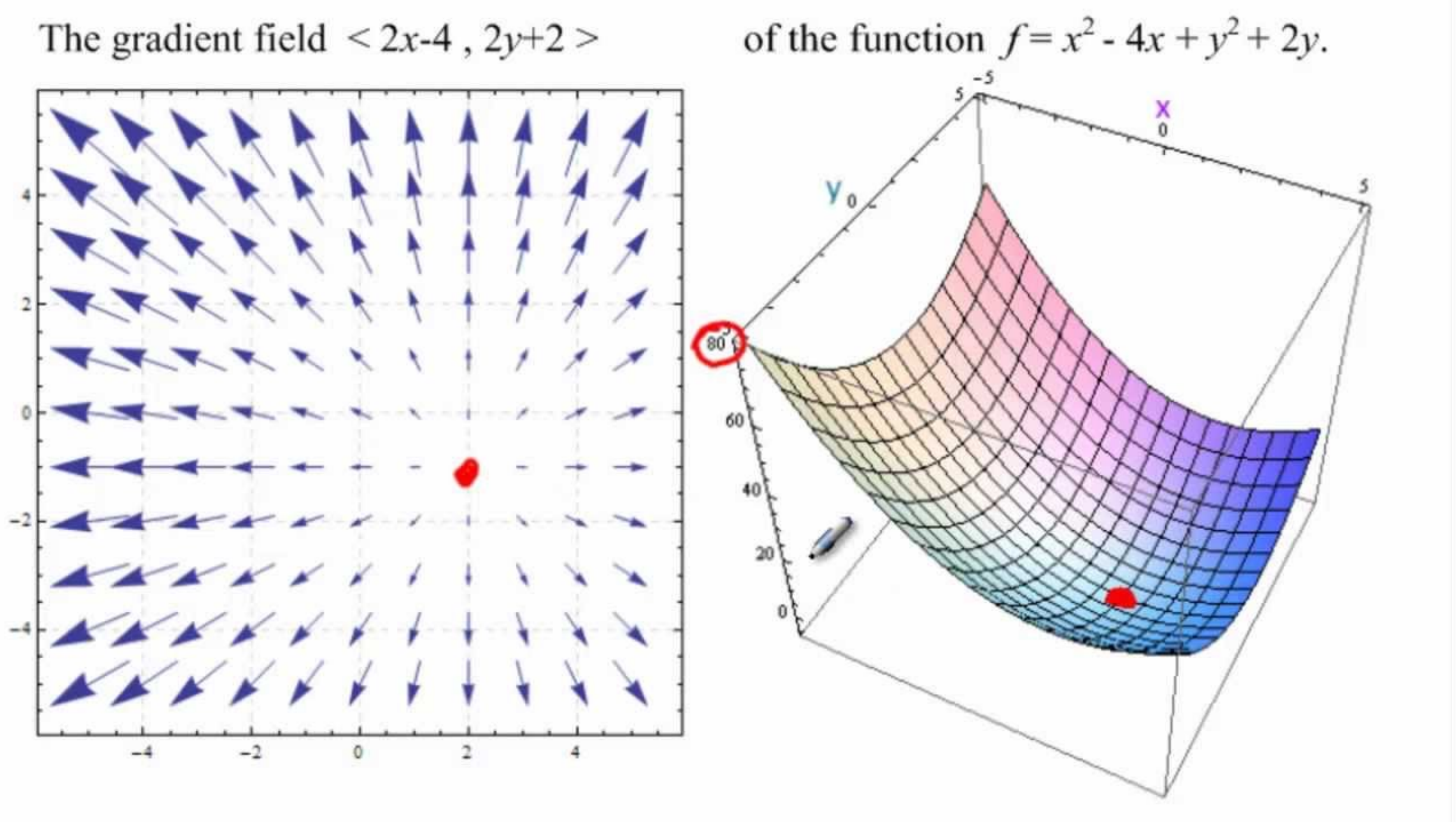
Свойства градиента





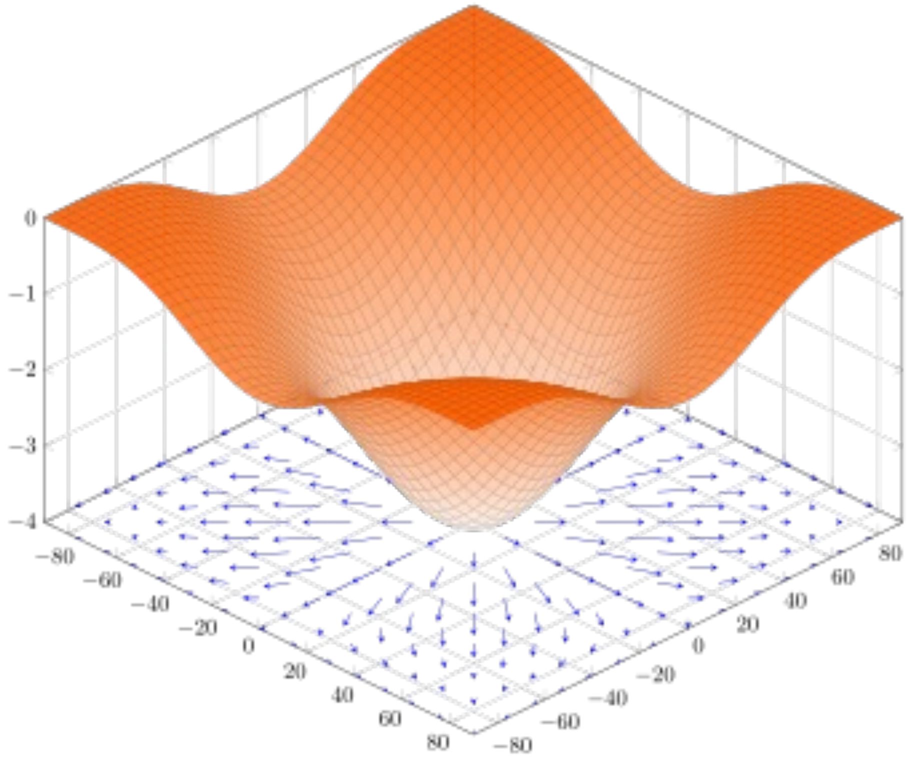
Карта градиентов и линии уровня функции

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

- Есть функция $L(w)$, минимум которой хотим найти
- Пусть w_0 - начальный вектор параметров
- Тогда **градиентный спуск** состоит в повторении:

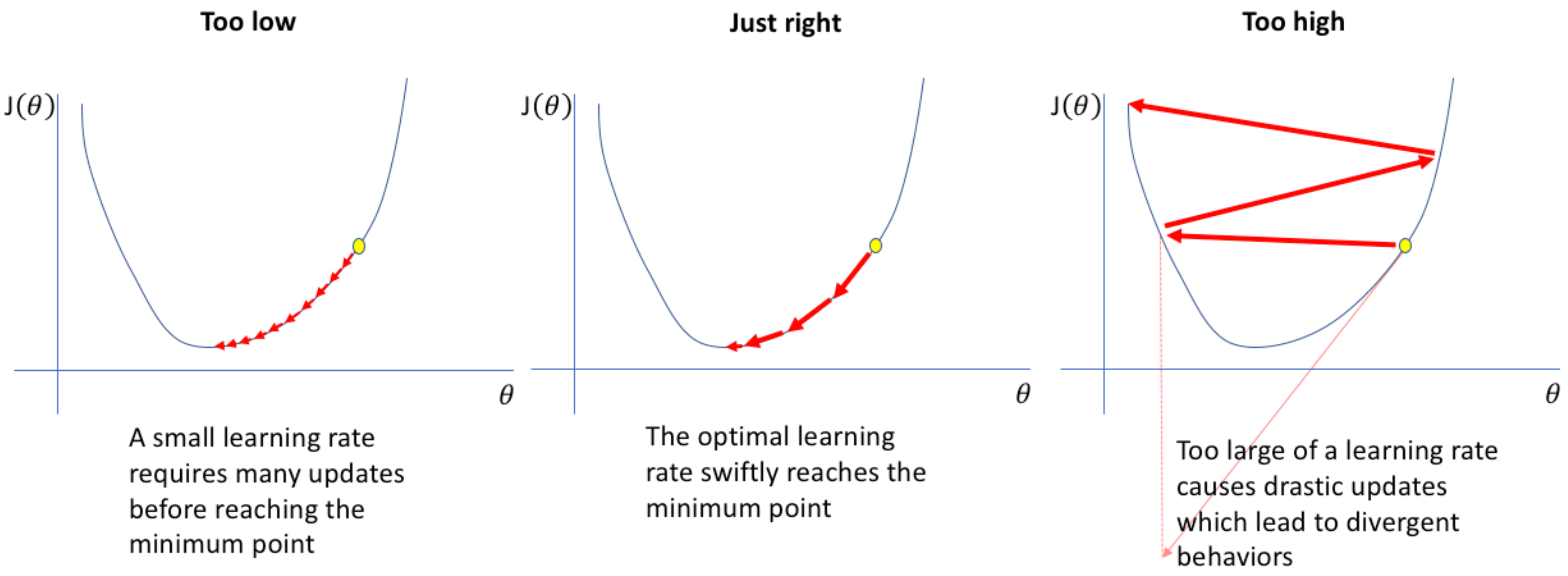
$$w^{(k+1)} = w^{(k)} - \boldsymbol{\eta} \nabla L(w^{(k)})$$

  – $\boldsymbol{\eta}$ – длина шага градиентного спуска (**learning rate**) (мы сами его задаем)
  – k – номер итерации

Градиентный спуск

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

Выбор шага


Too low
Just right
Too high
J(θ)
θ
J(θ)
θ
J(θ)
θ
A small learning rate requires many updates before reaching the minimum point
The optimal learning rate swiftly reaches the minimum point
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: $|w^{(k+1)} - w^{(k)}| \approx 0$

Повтор

- 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:

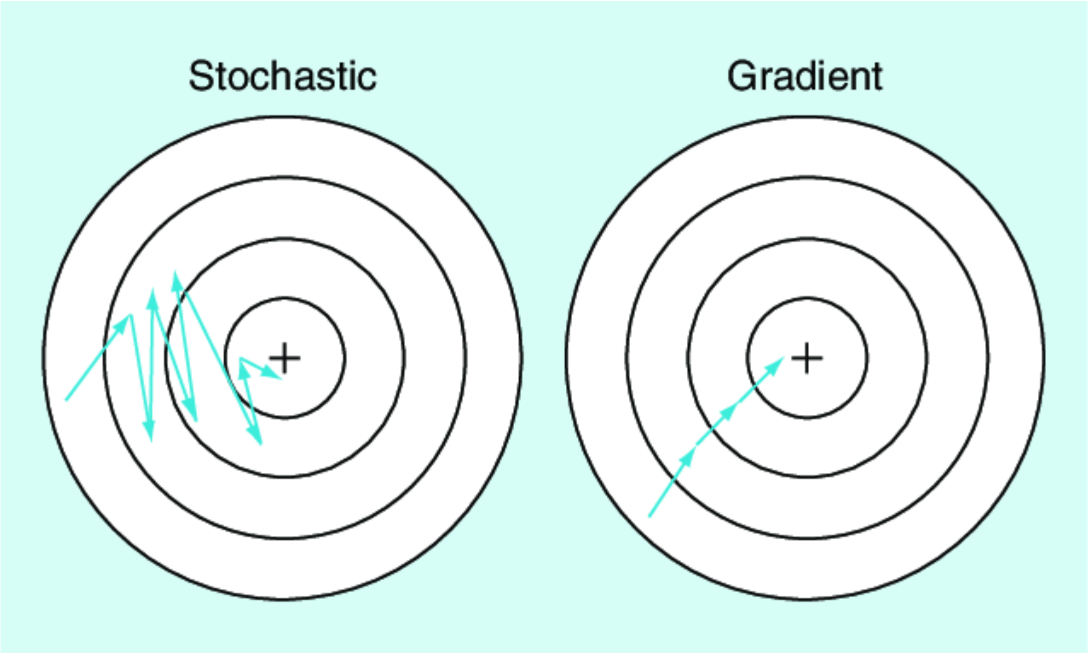
$$L(w) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

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

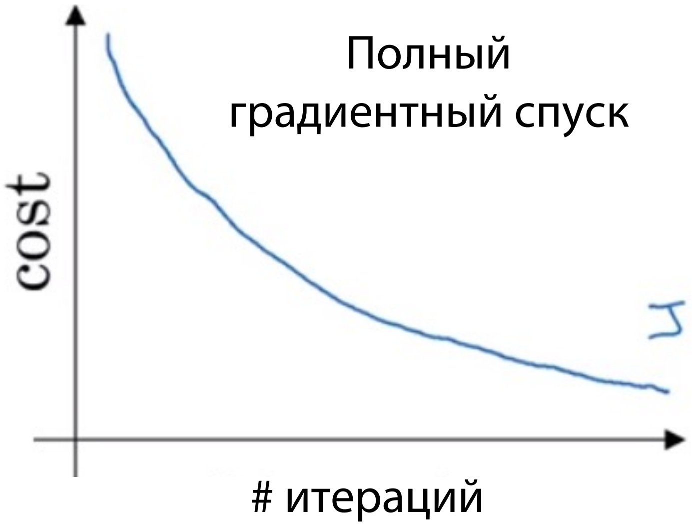
- **Стохастический** градиентный спуск:

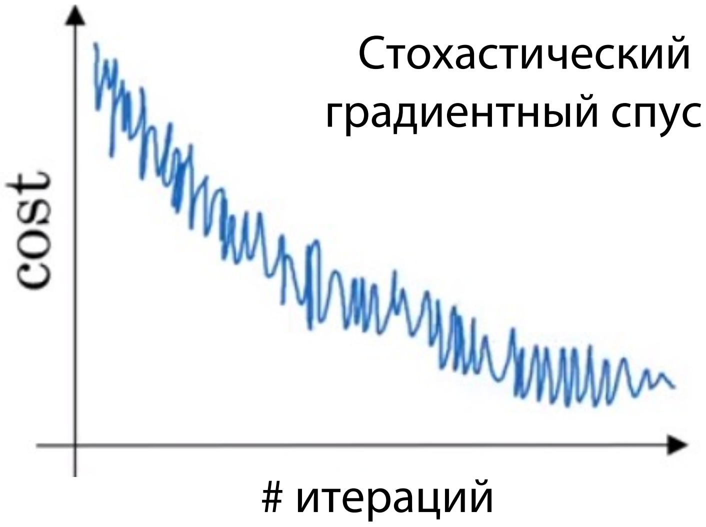
$$L_{\boldsymbol{i}}(w) = (\boldsymbol{\hat{y}_i} - \boldsymbol{y_i})^2$$

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



Стохастический градиентный спуск





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

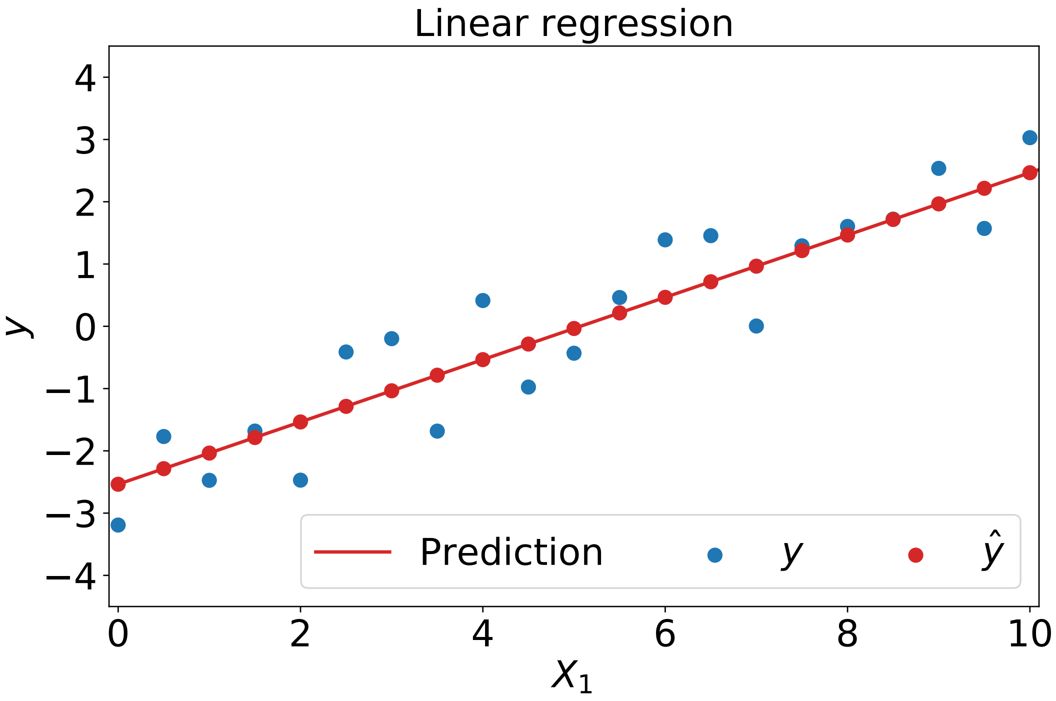
Метрики качества

Задача

Пусть даны X, y и линейна модель:

$$\hat{y} = Xw$$

Цель – измерить **качество модели**, определить насколько блзки прогнозы $\hat{y}$ к реальным значениям y.



Популярные метрики качества

- Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$$

- Mean Absolute Error (MAE):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|$$

- Трудно определить хорошую модель: $RMSE = 1$ выражает разное качество моделей при $\bar{y} = 100$ and $\bar{y} = 1$

Другие метрики качества #1

- Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{100}{N} \sum_{i=1}^{N} \left| \frac{\hat{y}_i - y_i}{y_i} \right|$$

- Измеряем относительную ошибку модели
- Легко интерпретировать
- Чувствительна к масштабу y

Другие метрики качества #2

- Relative Squared Error (RSE):

$$RSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}}$$

- Relative Absolute Error (RAE):

$$RAE = \frac{\sum_{i=1}^{N}|y_i - \hat{y}_i|}{\sum_{i=1}^{N}|y_i - \bar{y}|}$$

- Робастны (мене чувствительны) к масштабу y

Other quality metrics #3

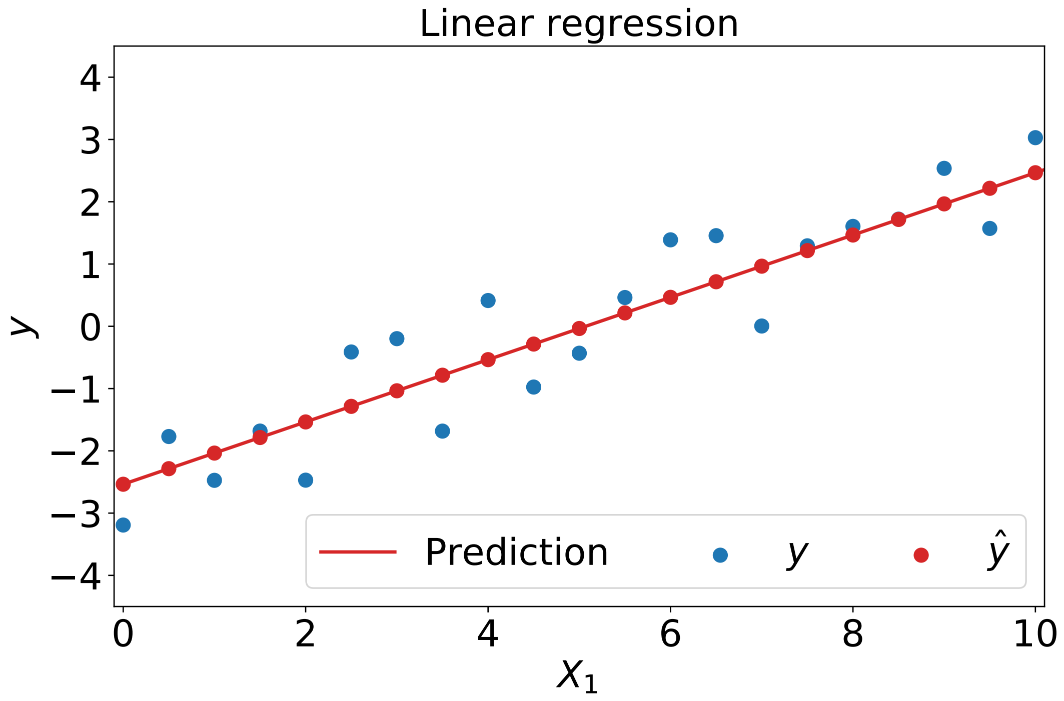
- Root Mean Squared Logarithmic Error (RMSLE):

$$RMSLE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log(y_i + 1) \ - \log(\hat{y}_i + 1))^2}$$

- Отличный выбор, когда y_i меняется на несколько порядков: $y_i \in [0, 10^6]$

Пример

Metric	No outliers
RMSE	0.67
MAE	0.59
MAPE, %	1035
RSE	0.39
RAE	0.40

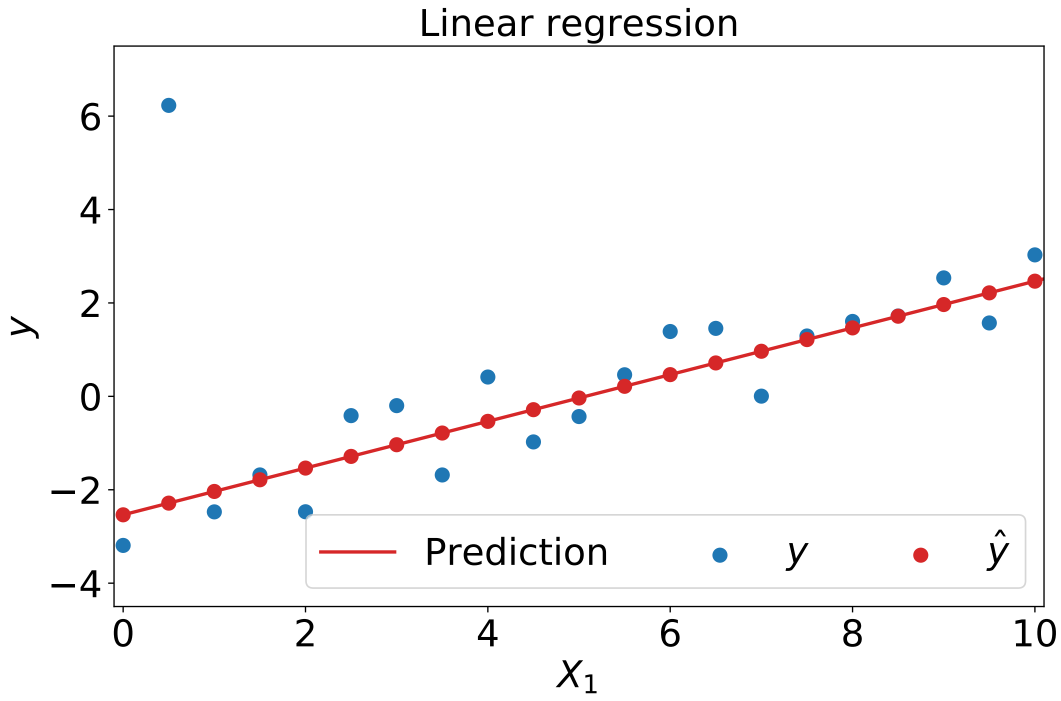


MAPE ведет себя плохо, потому что y и y_i близки к 0

Demonstration

Metric	No outliers	With outlier
RMSE	0.67	1.93
MAE	0.59	0.96
MAPE, %	1035	1040
RSE	0.39	0.92
RAE	0.40	0.58

- Выбросы могут сместить метрики
- MAE и RAE более робастны



Переобучение

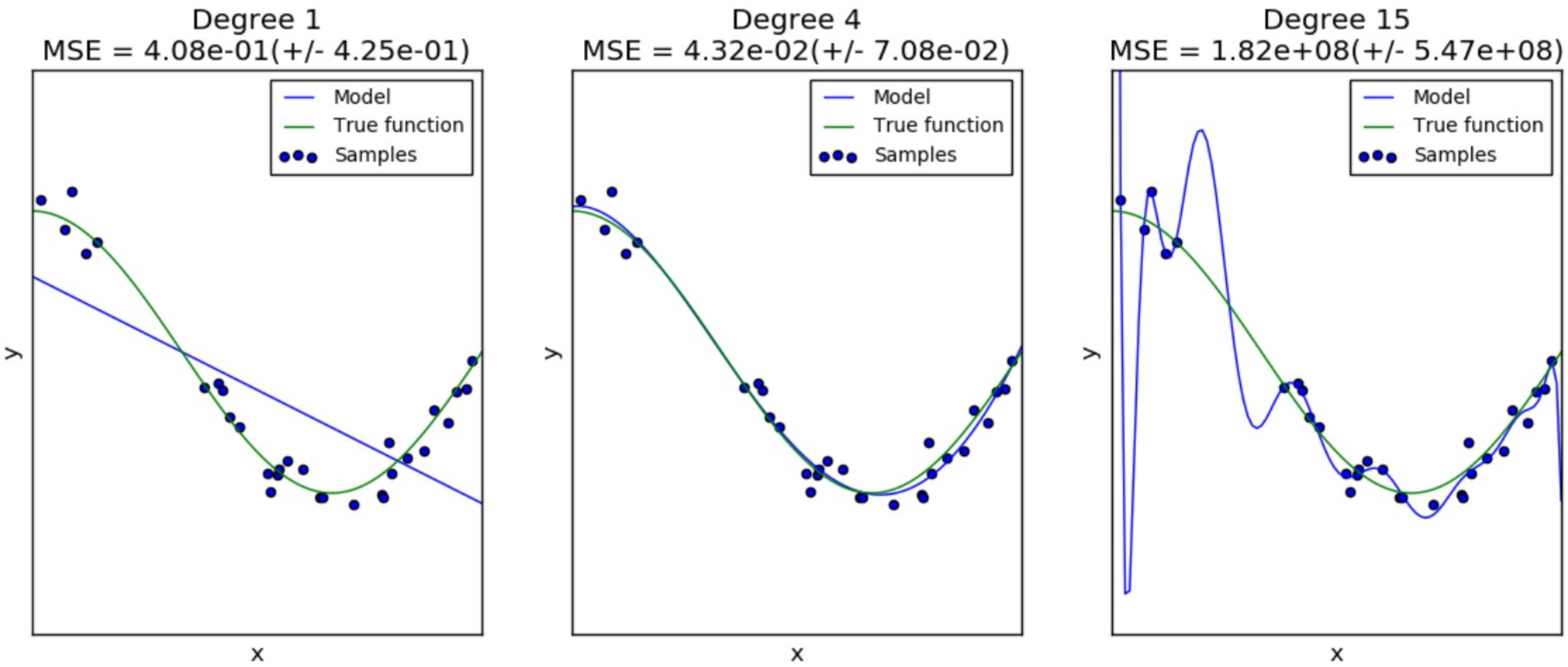
Задача

- 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:
  - $x_{i1}, x_{i1}^2, x_{i1}^3, \dots, x_{i1}^k$
- Рассмотрим модель полиномиальной линейной регрессии:

$$\hat{y}_i = w_0 + \sum_{j=1}^{\boldsymbol{k}} w_j x_{i1}^j$$

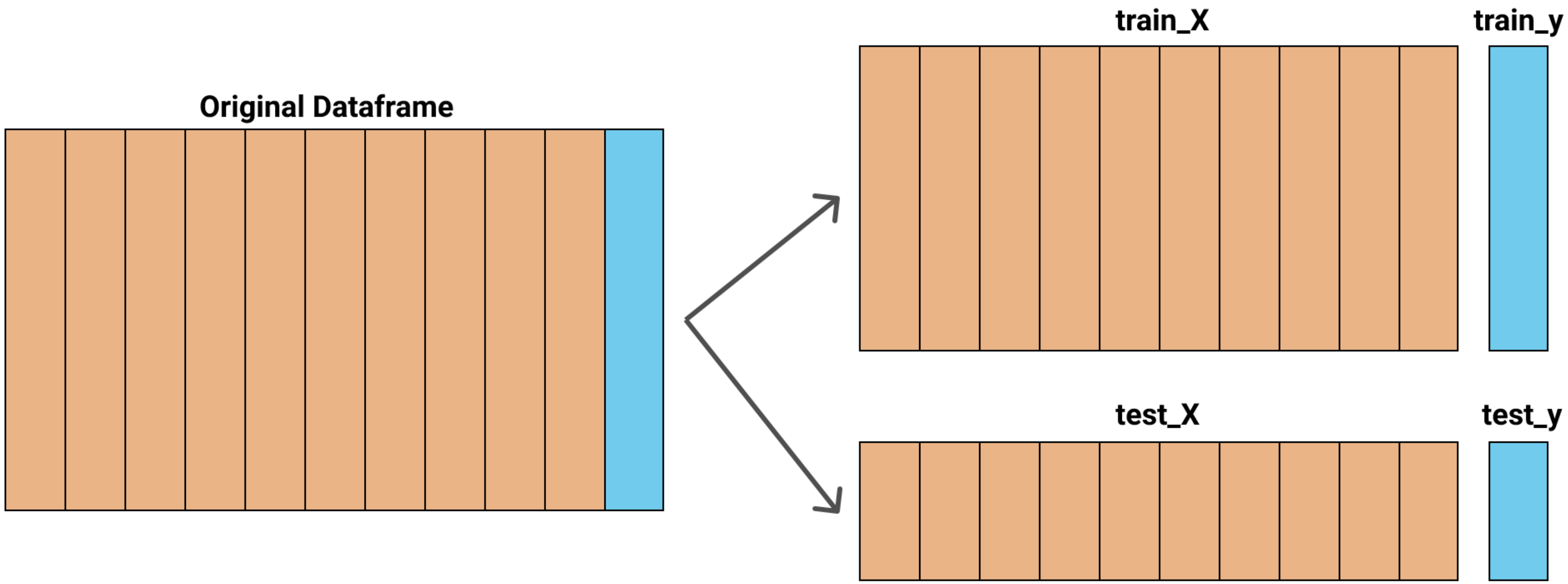
- Максимальную степень полинома $\boldsymbol{k}$ будем менять от 1 до 15

Решение

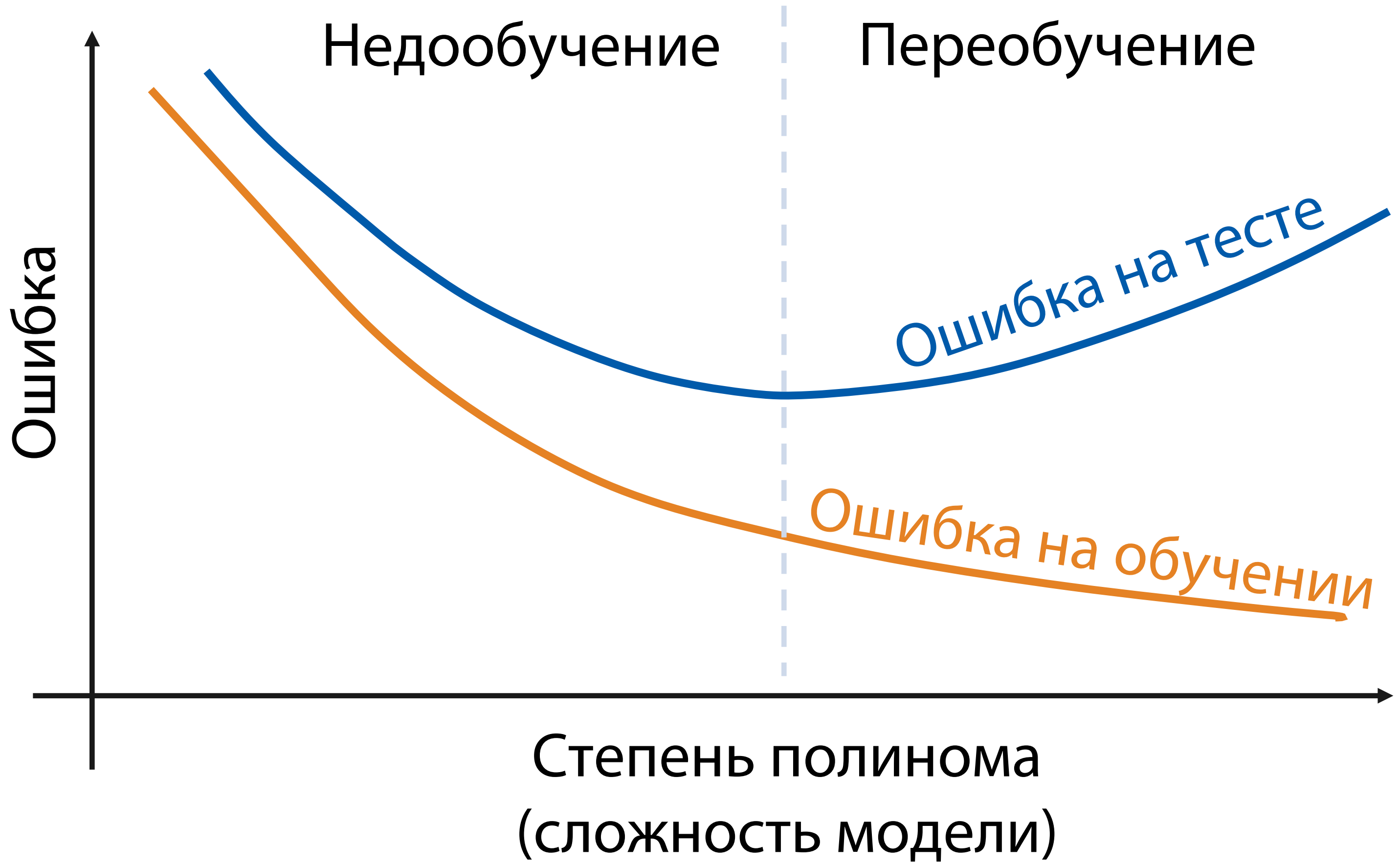

Degree 1
MSE = 4.08e-01(+/- 4.25e-01)
Model
True function
Samples
y
x
Degree 4
MSE = 4.32e-02(+/- 7.08e-02)
Model
True function
Samples
y
x
Degree 15
MSE = 1.82e+08(+/- 5.47e+08)
Model
True function
Samples
y
x

Обучение и тест

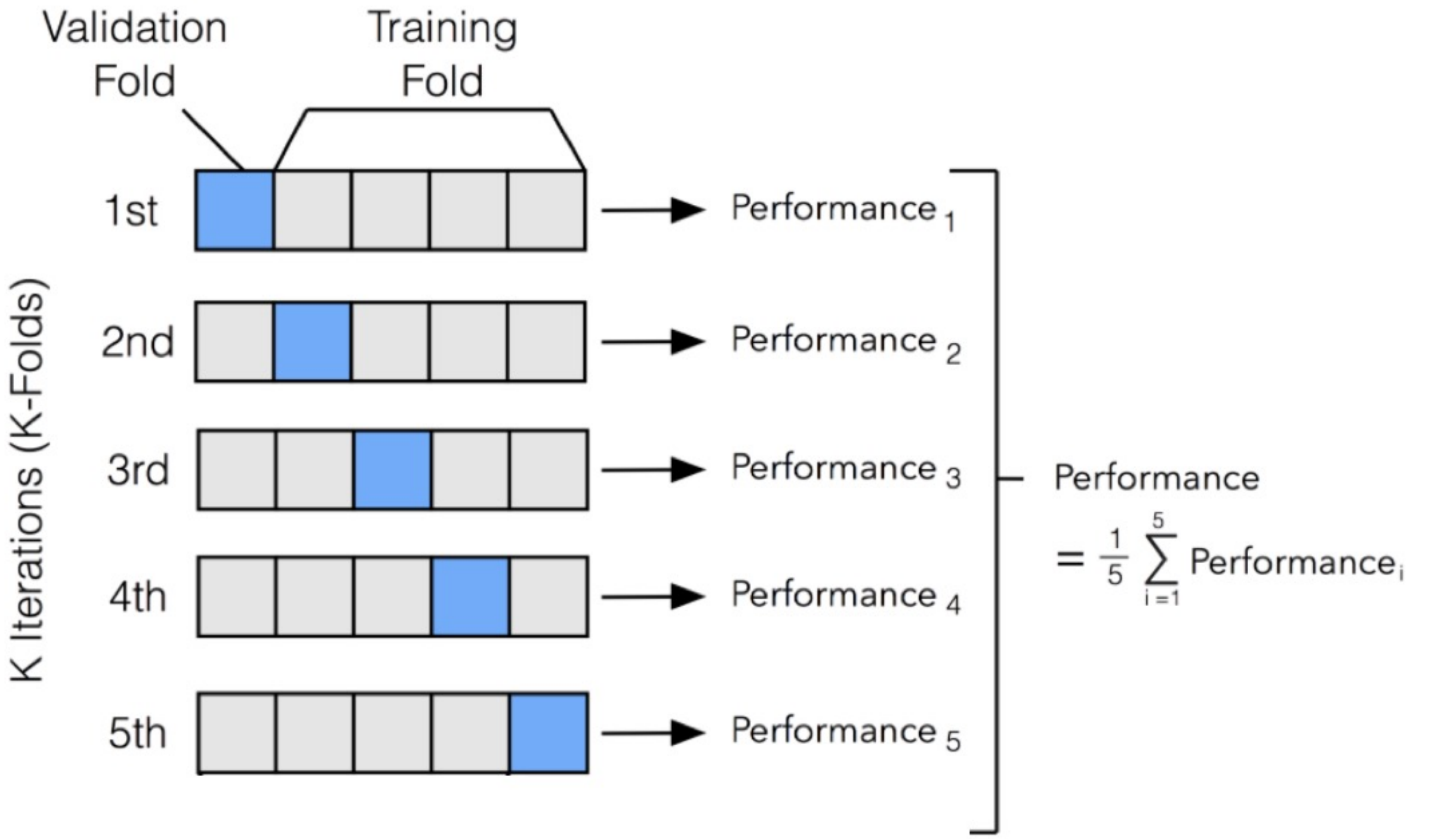
- **Обучающая выборка (train):** для обучения модели
- **Тестовая (отложенная) выборка (test):** для измерения качества модели



Переобучение



K-Fold кросс-валидация

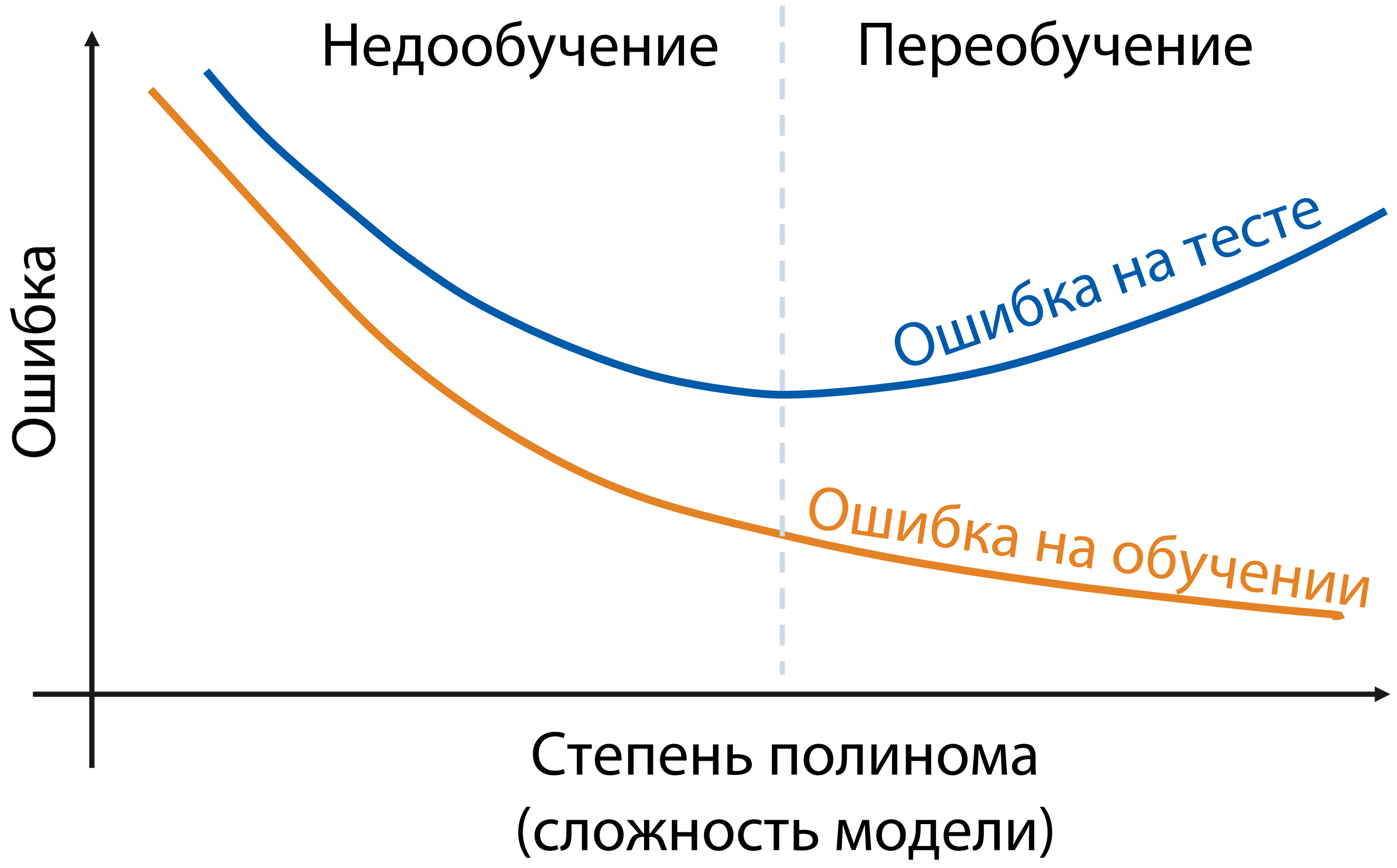


Кросс-валидация (cross-validation)

- Используется для измерения качества моделей в машинном обучении
- Отложенная выборка (train / test):
  - Делим всю выборку на две подвыборки в пропорции 70:30
  - Большая часть данных не используется для обучения (хуже качество модели)
- K-Fold кросс-валидация
  - K берем порядка 10
  - Больше данных участвует в обучении отдельной модели
  - Проверяем качество на всех данных
  - Более точная оценка качества

Регуляризация

Проблема переобучения



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):

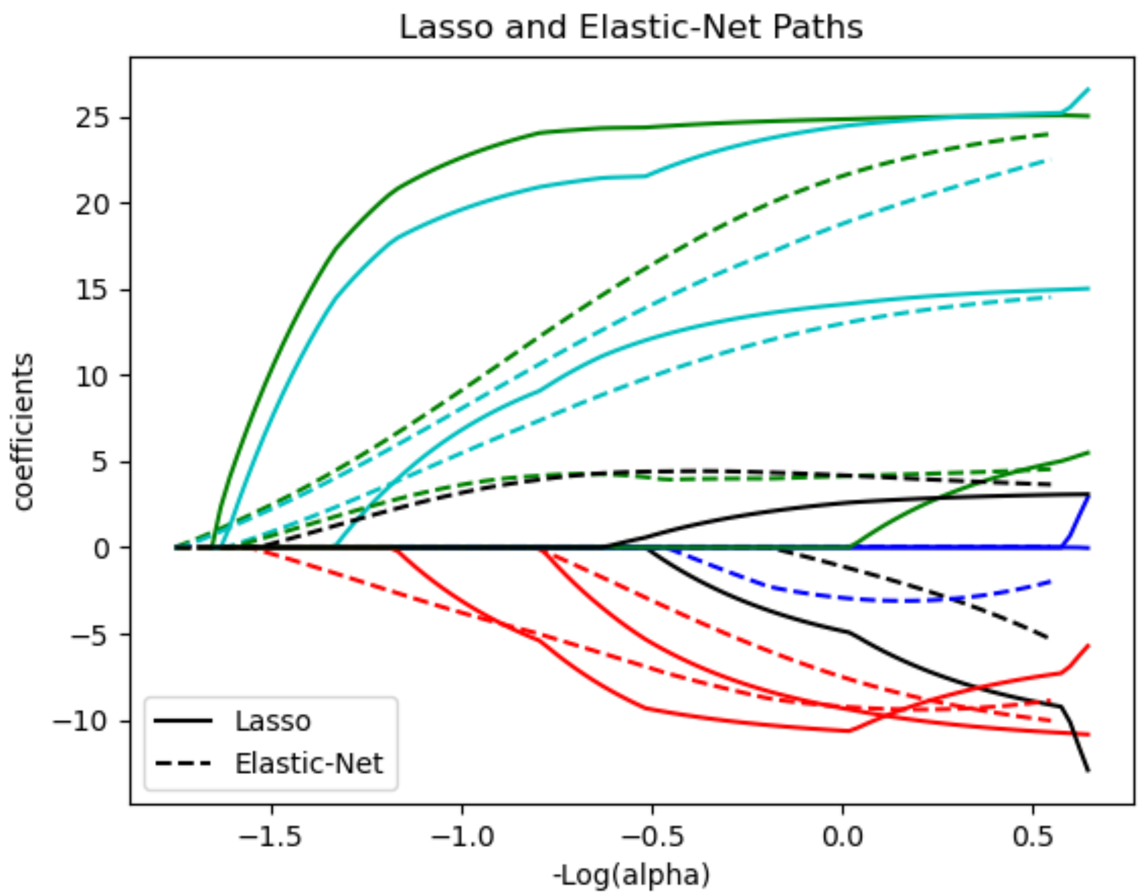
$$R(w) = \sum_{j=1}^{d} |w_j|$$

- L_2 регуляризация (Ridge):

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

Свойства регуляризации

- L_2 регуляризация стремится уменьшить веса модели
- L_1 позволяет проводить **отбор признаков**
- L_1 **обнуляет веса** для наименее информативных признаков



https://scikit-learn.org

Объяснение

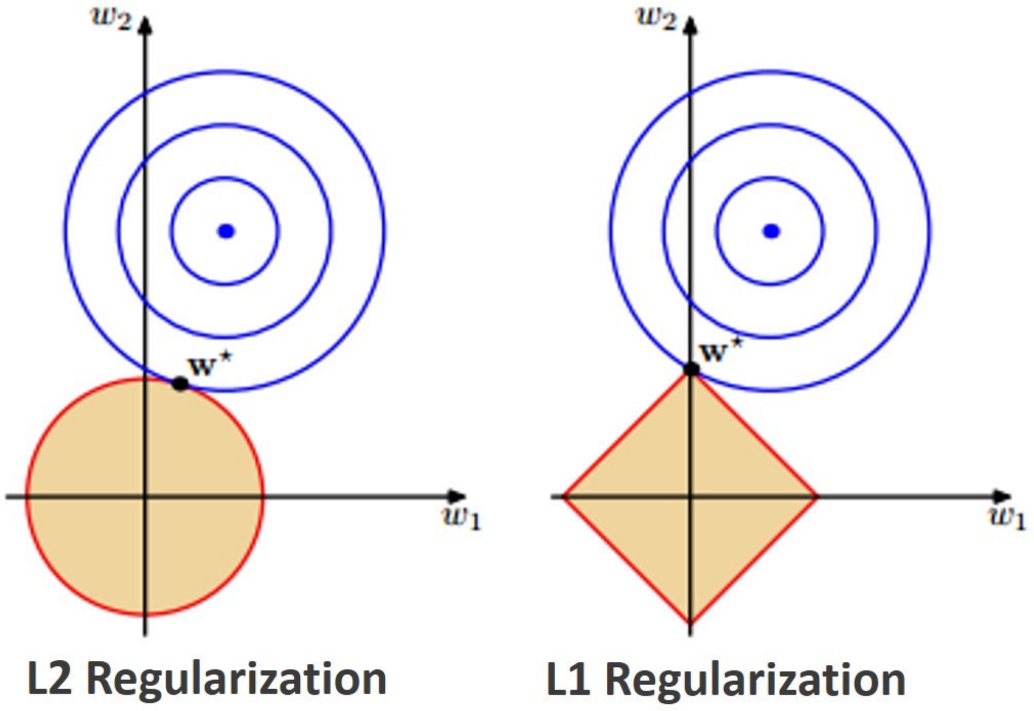
- Минимизация функции потерь

$$L(w) + \alpha R(w) \to \min_w$$

- Эквивалентна задаче условной минимизации:

$$\begin{cases} L(w) \to \min\limits_w \\ R(w) \le C \end{cases}$$

- Оптимум такой задачи чаще оказывается в 0 для L_1 регуляризации



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**.

Заключение

Резюме

- Модель линейной регрессии:

$$\hat{y} = Xw$$

- Функция потерь MSE с регуляризацией:

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

- Мы хотим минимизировать L:

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

- Градиентный спуск:

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

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.
- В чем состоят преимущества и недостатки использования метрик Mean squared error (MSE) и Mean absolute error (MAE) в задаче регрессии? Запишите формулу метрики Mean absolute percentage error (MAPE).
- 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?