

- Постановка задачи обучения с учителем
- Признаки
- Типы задач
- Переобучение
- Линейная регрессия
- Градиентный спуск
- L1 и L2 регуляризация

Входные данные: Набор обучающих объектов

$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, где:

- x_i — вектор признаков i -го объекта.
- y_i — метка (целевое значение) для i -го объекта.

Задача: Найти функцию $f(x)$ такую, что для новых объектов x она предсказывала метки y как можно точнее:

$$y = f(x)$$

$$f : X \rightarrow Y$$

1. Количественные признаки $x \in \mathbb{R}$

Пример: количество детей, температура.

2. Категориальные признаки

$$x \in \{C_1, C_2, \dots, C_k\}, \quad \text{где } k — \text{ количество категорий.}$$

Пример: цвет автомобиля, тип постройки.

3. Упорядоченные признаки

$$x \in \{O_1 < O_2 < \dots < O_k\}, \quad \text{где } O_i — \text{ упорядоченные категории.}$$

Пример: уровень образования (начальное, среднее, высшее), размер одежды (S, M, L).

4. Бинарные признаки $x \in \{0, 1\}$

Пример: результат теста (положительный/отрицательный).

Текстовые данные

- Длина текста может быть произвольной.
- Слова могут иметь разные значения в зависимости от контекста.
- Тексты часто содержат шум (опечатки, сокращения).

Изображения

- Высокая размерность данных.
- Пространственные зависимости между соседними пикселями.
- Различные форматы и разрешения.

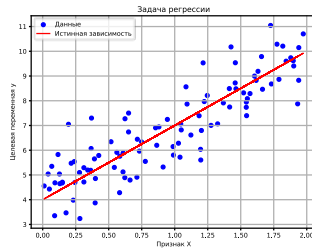
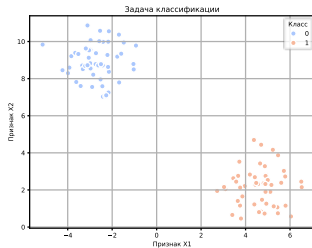
Временные ряды

- Последовательная зависимость между точками данных.
- Возможность наличия сезонности и цикличности.
- Необходимость учета времени как параметра.

Тип задачи	Множество Y
Регрессия	$Y \subseteq \mathbb{R}$
Бинарная классификация	$Y \subseteq \{0, 1\}$
Многоклассовая классификация	$Y = \{C_1, C_2, \dots, C_k\}$
Упорядоченная классификация	$Y = \{O_1 < O_2 < \dots < O_k\}$
Многомерная регрессия	$Y \subseteq \mathbb{R}^n$

К какому типу могут относиться следующие задачи обучения с учителем?

- Оценка качества товара
- Распознавание цифр
- Предсказание стоимости дома
- Письмо это спам/не спам
- Предсказание численных характеристик системы



Классификация и регрессия

Дано

- $x_1, \dots, x_n$ - обучающая выборка
- $y_1 \dots y_n$ - множество ответов
При этом (x, y) из некоторого распределения D
- $a(x, w) : X \times W \rightarrow Y$ - параметрическая модель

Найти

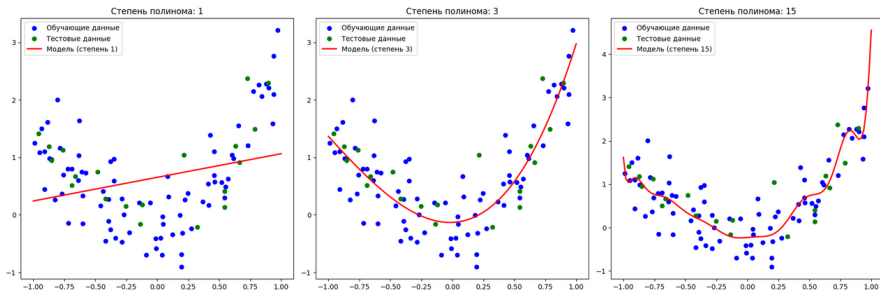
$$w : a(x, w) \rightarrow y(x)$$

Критерий минимизации эмперического риска

$$\sum_{i=1}^n L(w, x_i) \rightarrow \min_w$$

- $L(w, x)$ - функция потерь

- Определение задачи
- Подготовка данных
- Выбор модели
- Обучение модели
- Оценка модели
- Улучшение модели
- Развертывание модели



Степень полинома: 1, Ошибка на обучающей выборке: 0.69, Ошибка на тестовой выборке: 0.51

Степень полинома: 3, Ошибка на обучающей выборке: 0.20, Ошибка на тестовой выборке: 0.16

Степень полинома: 15, Ошибка на обучающей выборке: 0.18, Ошибка на тестовой выборке: 0.20

Переобучение

Переобучение (overfitting) — ситуация, когда модель слишком хорошо обучается на обучающих данных, запоминая шум и детали, которые не имеют отношения к реальной зависимости.

Недообучение (underttting) — ситуация, когда модель слишком плохо обучается на обучающих данных, в силу малого числа параметров.

Как заметить переобучение?

- Высокая точность на обучающей выборке и низкая на тестовой.
- Модель становится чрезмерно сложной для задачи.

Как бороться с переобучением?

- Уменьшение сложности модели
- Регуляризация
- Увеличение объема данных
- Кросс-валидация

Дано

Пусть имеется набор данных $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, где:

- $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{id})^\top \in \mathbb{R}^d$ – вектор признаков объекта x_i
- $y_i \in \mathbb{R}$ – целевая переменная для объекта i

Линейная модель

$$y = f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b,$$

где:

- $\mathbf{w} = (w_1, w_2, \dots, w_d)^\top \in \mathbb{R}^d$ – вектор весов
- $b \in \mathbb{R}$ – смещение (bias)

- $X \in \mathbb{R}^{n \times d}$ – матрица размерности $n \times d$
- где каждая строка $X[i, :]$ соответствует вектору признаков объекта i
- $\mathbf{y} \in \mathbb{R}^n$ – вектор целевых значений длины n

Тогда предсказания модели можно записать как:

$$\hat{\mathbf{y}} = X\mathbf{w} + b\mathbf{1},$$

Для удобства часто добавляют дополнительный столбец единиц к матрице X , чтобы объединить веса $\mathbf{w}$ и смещение b . Тогда модель можно записать как:

$$\hat{\mathbf{y}} = X'\mathbf{w}',$$

- $X' \in \mathbb{R}^{n \times (d+1)}$ – расширенная матрица признаков, полученная добавлением столбца единиц к X
- $\mathbf{w}' = (\mathbf{w}^\top, b)^\top \in \mathbb{R}^{d+1}$ – расширенный вектор параметров.

Основная задача обучения линейной модели — найти **оптимальные параметры $\mathbf{w}$ и b** , которые **минимизируют функцию потерь**. Наиболее распространенной функцией потерь в задачи регрессии является **среднеквадратичная ошибка (MSE)**:

$$L(\mathbf{w}, b) = \frac{1}{n} \sum_{i=1}^n (y_i - (\mathbf{w}^\top \mathbf{x}_i + b))^2.$$

В матричной форме эта функция записывается как:

$$L(\mathbf{w}', X', \mathbf{y}) = \frac{1}{n} \|\mathbf{y} - X' \mathbf{w}'\|^2,$$

где $\|\cdot\|$ обозначает евклидову норму.

Оптимальные параметры находятся путем минимизации этой функции:

$$\mathbf{w}'_{\text{opt}} = \arg \min_{\mathbf{w}'} L(\mathbf{w}', X', \mathbf{y}).$$

Распишем функцию потерь более подробно:

$$L(\mathbf{w}') = \frac{1}{n} (\mathbf{y} - X'\mathbf{w}')^\top (\mathbf{y} - X'\mathbf{w}').$$

Развернем скалярное произведение:

$$L(\mathbf{w}') = \frac{1}{n} [\mathbf{y}^\top \mathbf{y} - 2\mathbf{y}^\top X'\mathbf{w}' + \mathbf{w}'^\top X'^\top X'\mathbf{w}'] .$$

$$L(\mathbf{w}') \propto -2\mathbf{y}^\top X'\mathbf{w}' + \mathbf{w}'^\top X'^\top X'\mathbf{w}' .$$

Чтобы найти минимум функции потерь, возьмем производную по $\mathbf{w}'$ и приравняем её к нулю:

$$\nabla_{\mathbf{w}'} L = -2X'^\top \mathbf{y} + 2X'^\top X'\mathbf{w}' = 0.$$

Упростим:

$$X'^\top X'\mathbf{w}' = X'^\top \mathbf{y}.$$

Если матрица $X'^\top X'$ обратима (то есть она невырождена), то решение имеет вид:

$$\mathbf{w}'_{\text{opt}} = (X'^\top X')^{-1} X'^\top \mathbf{y}.$$

Градиентный спуск — это **итерационный** метод оптимизации, который обновляет параметры модели $\mathbf{w}$ следующим образом:

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} L,$$

где:

- $\eta > 0$ — скорость обучения (learning rate),
- $\nabla_{\mathbf{w}} L$ — градиент функции потерь по $\mathbf{w}$

Задача 2

- Какова вычислительная сложность у аналитического решения?
- Какова вычислительная сложность у решения через градиентный спуск?

У нас есть данные о цене (y) в зависимости от площади дома (x):

x (площадь, м ²)	цена, млн руб Y
1	2
2	4
3	6
4	8

Мы хотим найти параметры линейной модели:

$$y = wx + b,$$

используя градиентный спуск для минимизации среднеквадратичной ошибки (MSE):

$$L(w, b) = \frac{1}{n} \sum_{i=1}^n (y_i - (wx_i + b))^2.$$

$$\frac{\partial L}{\partial w} = -\frac{2}{n} \sum_{i=1}^n x_i (y_i - (wx_i + b)).$$

$$\frac{\partial L}{\partial b} = -\frac{2}{n} \sum_{i=1}^n (y_i - (wx_i + b)).$$

$$w_{\text{new}} = w_{\text{old}} - \eta \cdot \frac{\partial L}{\partial w},$$

$$b_{\text{new}} = b_{\text{old}} - \eta \cdot \frac{\partial L}{\partial b},$$

Пусть $w_0 = 0$, $b_0 = 0$, $\eta = 0.01$

Итерация 1

$$w_0 = 0, b_0 = 0.$$

$$\hat{y}_i = w_0 x_i + b_0 = 0 \text{ для всех } i.$$

$$e_i = y_i - \hat{y}_i = y_i - 0 = y_i.$$

$$\frac{\partial L}{\partial w} = -\frac{2}{4} \sum_{i=1}^4 x_i e_i = -\frac{2}{4} (1 \cdot 2 + 2 \cdot 4 + 3 \cdot 6 + 4 \cdot 8) = -\frac{2}{4} \cdot 60 = -30,$$

$$\frac{\partial L}{\partial b} = -\frac{2}{4} \sum_{i=1}^4 e_i = -\frac{2}{4} (2 + 4 + 6 + 8) = -\frac{2}{4} \cdot 20 = -10.$$

$$w_1 = w_0 - \eta \cdot \frac{\partial L}{\partial w} = 0 - 0.01 \cdot (-30) = 0.3,$$

$$b_1 = b_0 - \eta \cdot \frac{\partial L}{\partial b} = 0 - 0.01 \cdot (-10) = 0.1.$$

Итерация 2

$$w_1 = 0.3, b_1 = 0.1$$

$$\hat{y}_i = w_1 x_i + b_1 = 0.3x_i + 0.1$$

$$e_1 = 2 - (0.3 \cdot 1 + 0.1) = 1.6, \quad e_2 = 4 - (0.3 \cdot 2 + 0.1) = 3.3,$$

$$e_3 = 6 - (0.3 \cdot 3 + 0.1) = 5.0, \quad e_4 = 8 - (0.3 \cdot 4 + 0.1) = 6.7.$$

$$\frac{\partial L}{\partial w} = -\frac{2}{4} \sum_{i=1}^4 x_i e_i = -\frac{2}{4} (1 \cdot 1.6 + 2 \cdot 3.3 + 3 \cdot 5.0 + 4 \cdot 6.7),$$

$$\frac{\partial L}{\partial b} = -\frac{2}{4} \sum_{i=1}^4 e_i = -\frac{2}{4} (1.6 + 3.3 + 5.0 + 6.7) = -\frac{2}{4} \cdot 16.6 = -8.3.$$

$$w_2 = w_1 - \eta \cdot \frac{\partial L}{\partial w} = 0.3 - 0.01 \cdot (-24.3) = 0.543,$$

$$b_2 = b_1 - \eta \cdot \frac{\partial L}{\partial b} = 0.1 - 0.01 \cdot (-8.3) = 0.183.$$

В классическом градиентном спуске градиент вычисляется по **всем данным**.

Этот подход требует обработки всех данных на каждой итерации, что может быть медленным для больших наборов данных.

Mini-batch Gradient Descent

$$\nabla L_B(\mathbf{w}) = \frac{1}{b} \sum_{i \in B} \nabla l(\mathbf{w}; \mathbf{x}_i, y_i),$$

где B — случайный мини-батч размера b .

Обновление параметров происходит аналогично:

$$\mathbf{w}_{\text{new}} = \mathbf{w}_{\text{old}} - \eta \cdot \nabla L_B(\mathbf{w}_{\text{old}}).$$

Задача 3

Какая асимптотическая сложность у данного алгоритма?

Исходная функция потерь

$$L(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^n (y_i - (\mathbf{w}^\top \mathbf{x}_i + b))^2,$$

L2-регуляризация (Ridge)

$$L_{\text{Ridge}}(\mathbf{w}) = L(\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2$$

- $\|\mathbf{w}\|_2^2 = \sum_{j=1}^d w_j^2$ – квадрат евклидовой нормы вектора $\mathbf{w}$,
- $\lambda > 0$ – коэффициент регуляризации.

L1-регуляризация (Lasso)

$$L_{\text{Lasso}}(\mathbf{w}) = L(\mathbf{w}) + \lambda \|\mathbf{w}\|_1$$

- $\|\mathbf{w}\|_1 = \sum_{j=1}^d |w_j|$ – L_1 -норма вектора $\mathbf{w}$,
- $\lambda > 0$ – коэффициент регуляризации.

L2-регуляризация (Ridge)

1. **Уменьшает** переобучение, сокращая величину весов.
2. Применение: Хорошо работает, когда **все признаки** важны, но их значимость нужно уменьшить.

L1-регуляризация (Lasso)

1. **Обнуляет** некоторые веса, что позволяет автоматически отбирать наиболее важные признаки.
2. Применение: Используется, когда **важны только некоторые** признаки, а остальные можно игнорировать.

Вопросы?