

## Лекция 2. Восстановление регрессии

### Основы интеллектуального анализа данных

Полузёров Т. Д.

БГУ ФПМИ

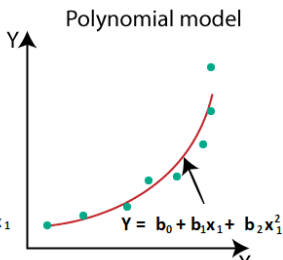
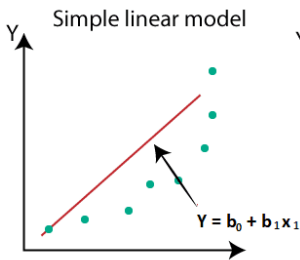
- 1 Линейная регрессия
  - МНК
  - Мультиколлинеарность
  - Регуляризация
- 2 Нелинейные обобщения
  - Преобразования признаков
  - Нелинейная модель
  - Другие функция потерь
- 3 Градиентные методы оптимизации

# Постановка задачи регрессии

Пусть имеется выборка  $(X, y)_{i=1}^{\ell}$ , где  $X = (x_i)_{i=1}^{\ell} \subseteq \mathbb{X} = \mathbb{R}^{\ell \times n}$  - матрица признаков,  $y = (y_i)_{i=1}^{\ell} \subseteq \mathbb{Y} = \mathbb{R}^{\ell}$  - вектор целевых значений.

Между  $\mathbb{Y}$  и  $\mathbb{X}$  существует некоторая неизвестная зависимость  $y^* : \mathbb{X} \rightarrow \mathbb{Y}$

Задача регрессии состоит в том, чтобы по имеющимся данным  $(X, y)$  с помощью некоторой функции  $a(x, \theta), \theta \in \Theta$  приблизить  $y^*$  на всем множестве  $\mathbb{X}$ .



# Метод наименьших квадратов

Для решения такого рода задач применяется **метод наименьших квадратов (МНК)**:

$$Q(\theta, X) = \sum_{i=1}^{\ell} (a(x_i, \theta) - y_i)^2 \rightarrow \min_{\theta}$$

где  $a(x, \theta)$  - некоторая **модель регрессии** (параметрическое семейство функций).  $\theta = (\theta_1, \dots, \theta_p)^T$

Результат оптимизации - набор конкретных значений параметров для выбранного семейства:

$$\theta^* = \arg \min_{\theta} Q(\theta, X)$$

# Решение оптимизационной задачи МНК

В случае дифференцируемости  $a(x, \theta)$  по  $\theta$ , решение находится из системы из  $p$  уравнений (необходимое условие минимума):

$$\frac{\partial Q}{\partial \theta} = 2 \sum_{i=1}^{\ell} (a(x_i, \theta) - y_i) \frac{\partial a}{\partial \theta} = 0$$

Решая систему, получим обученную модель  $a^*(x) := a(x, \theta^*)$  описывающую зависимость  $y$  от  $x$  наилучшим образом (в среднеквадратичном смысле).

# Линейная регрессия

Частный случай, когда  $a(x, \theta)$  линейна по своим параметрам -  
**линейная регрессия**

$$a(x, \omega) = \omega_0 + \sum_{j=1}^n \omega_j x_j = \omega_0 + \langle \omega, x \rangle$$

Определяется вектором коэффициентов  $\omega = (\omega_1, \dots, \omega_n) \in \mathbb{R}^n$  и свободным членом  $\omega_0 \in \mathbb{R}$

Для упрощения формул добавим к признаковому описанию объектов признак равный единице

$$x := (1, x_1, \dots, x_n), \quad \omega := (\omega_0, \omega_1, \dots, \omega_n)$$

Тогда модель линейной регрессии:

$$a(x) = \langle \omega, x \rangle$$

## Применение МНК к линейной модели

Удобно работать в матричной форме:

$$Q(\omega) = \|X\omega - y\|^2$$

Необходимое условие минимума:

$$\frac{\partial Q}{\partial \theta} = 2X^T(X\omega - y) = 0$$

$$X^T X \omega = X^T y$$

Аналитическое решение:

$$\omega^* = (X^T X)^{-1} X^T y$$

# Проблема линейной зависимости признаков

Аналитическое МНК решение задачи линейной регрессии:

$$\omega^* = (X^T X)^{-1} X^T y$$

Если среди признаков (столбцов  $X$ ) есть **линейно зависимые**, то определитель матрицы  $X^T X$  равен нулю и её обращение  $(X^T X)^{-1}$  невозможно! Следовательно, **решения нет**.

Если матрица имеет полный ранг, но столбцы **почти линейно зависимы** (сильная корреляция), то говорят что матрица плохо обусловлена.



# Мультиколлинеарность

Почти линейную зависимость среди признаков называют **проблемой мультиколлинеарности**. Она ведет к:

- **большой разброс** по абсолютной величине и знаку у коэффициентов  $\omega^*$
- **неустойчивое обучение** - добавление или удаление нескольких объектов из  $X$  влечет значительно разные оптимальные  $\omega^*$
- **решение неустойчиво** - малое изменение входных данных влечет сильное изменение значения функции регрессии

# Методы борьбы с мультиколлинеарностью

Для борьбы с мультиколлинеарностью можно:

- удалять скоррелированные столбцы
- вводить ограничения на параметры
- добавить штраф в функционале качества, зависящий от значений параметров (регуляризация)

# Квадратичная регуляризация - гребневая регрессия

**Метод гребневой регрессии** (Ridge regression) состоит в добавлении слагаемого, штрафующего за большие веса:

$$Q_{\alpha}(\theta) = \|X\omega - y\|^2 + \alpha\|\omega\|^2$$

компоненту  $\alpha\|\omega\|^2$  называют квадратичным регуляризатором, а параметр  $\alpha$  - параметром регуляризации

В этом случае решение имеет вид:

$$\omega_{\alpha}^* = (X^T X + \alpha E)^{-1} X^T y$$

где  $E$  - единичная матрица

# Лассо - отбор признаков

Другая идея состоит в добавлении ограничения на сумму абсолютных значений весов. Называется **метод Лассо** (LASSO, Least Absolute Shrinkage and Selection Operator):

$$\begin{cases} Q(\theta) = \|X\omega - y\|^2 \rightarrow \min_{\omega} \\ \sum_{j=0}^n |\omega_j| \leq \beta \end{cases}$$

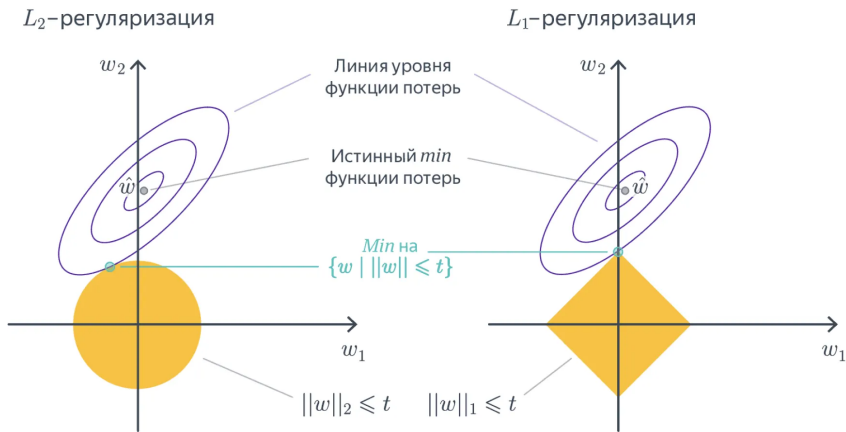
параметр  $\beta$  - селективность.

Альтернативная формулировка метода:

$$Q(\theta) = \|X\omega - y\|^2 + \beta \|\omega\|_1 \rightarrow \min_{\omega}$$

Особенность метода состоит в умении отбирать признаки. С уменьшением параметра  $\beta$  становится "выгоднее" **занулять некоторые веса**.

# Эффект от регуляризации



## Нелинейные преобразования признаков

Можно перейти от исходного признакового описания  $x = (x_1, \dots, x_n)$  к новым признакам  $g(x) = (g_1(x), \dots, g_m(x))$ .

Например, вместо сложной модели (нелинейной по признакам)  
 $a(x, \omega) = \omega_1 \ln(x_1) + \omega_2 \exp(x_2) + \omega_3 \frac{x_1}{x_2}$

можно перейти к новому признаковому описанию  $x' = g(x)$ ,  
 $g(x) = (\ln(x_1), \exp(x_2), \frac{x_1}{x_2})$ . И тогда

$a(x', \omega) = \omega_1 x'_1 + \omega_2 x'_2 + \omega_3 x'_3 = \langle w, x' \rangle$  - линейная модель и по признакам, и по весам.

# Нелинейная модель регрессии

Случай нелинейной модели  $a(x, \omega)$  по своим параметрам  $\omega$ .  
Другими словами, модель нельзя представить скалярным произведением:  $a(x, \omega) \neq \langle \omega, g(x) \rangle$ , где  $g(x)$  - некоторое преобразование признаков,  $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ .

Примеры:

- $a(x, \omega) = x^\omega$  - нелинейная
- $a(x, \omega) = \sin(x_1 \omega_1) + \cos(x_2 \omega_2)$  - нелинейная
- $a(x, \omega) = \omega_1 \ln(x_1) + \omega_2 \exp(x_2)$  - линейная

# Обобщенные линейные модели

Огромный класс моделей образуют Обобщенные линейные модели (**GLM**, Generalized linear Models).

Идея в том, что модель регрессии линейна, но задана нелинейная функция связи  $h(\cdot)$  между результатом модели и целевой переменной.

$$a(x, \omega) = h(\langle \omega, x \rangle)$$



## Другие функция потерь

Квадратичная функции потерь  $\mathcal{L}(a, y) = (a(x) - y)^2$  соответствует методу наименьших квадратов.

- $\mathcal{L}(a, y) = |a(x) - y|$  - линейная
- $\mathcal{L}(a, y) = [a(x) \neq y]$  - 0-1 функция
- $\mathcal{L}(a, y) = \ln(1 + e^{-a(x)y})$  - логистическая
- $\mathcal{L}(a, y) = \max(0, 1 - a(x)y)$  - Hinge loss

Необходимо подбирать функцию потерь которая лучшим образом описывающую бизнес требования задачи.

## Решение задачи оптимизации

В случае не квадратичной функции потерь или нелинейной модели регрессии решение ищут численными методами.

$$Q(a, X) = \sum_{i=1}^{\ell} \mathcal{L}(a(x_i, \omega), y_i) \rightarrow \min$$

Вектор частных производных (градиент) по параметрам в общем виде :

$$\frac{\partial Q}{\partial \omega_j} = \sum_{i=1}^{\ell} \frac{\partial \mathcal{L}}{\partial a} \cdot \frac{\partial a}{\partial \omega_j}$$

## Численное решение

Наиболее простой и подходящий класс методов –  
**градиентные методы оптимизации.**

Общая схема:

$$Q(\omega) = \sum_{i=1}^{\ell} \mathcal{L}_i(\omega) \rightarrow \min_{\omega}$$

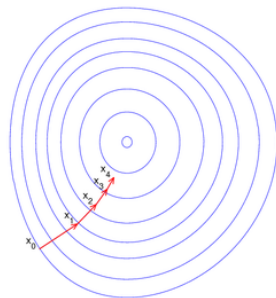
$$\omega^{(t+1)} = \omega^{(t)} - \alpha \cdot \nabla_{\omega} Q(\omega^{(t)})$$

где  $\alpha \in \mathbb{R}$

- некоторый параметр (размер шага)

Градиент квадратичного функционала:

$$\nabla_{\omega} Q = 2X^T(X\omega - y)$$



# Метод градиентного спуска

---

## Algorithm 1 Метод градиентного спуска

---

**Input:**  $\alpha$  - градиентный шаг (темп обучения)

**Output:**  $\omega^*$  - оптимум функционала  $Q(\omega)$

**begin**

    Инициализировать  $\omega^{(0)}$

**while** не выполнен критерий остановки **do**

        вычислить градиент в точке

$$\nabla Q(\omega)|_{\omega=\omega^{(t)}} = \left( \frac{\partial Q(\omega)}{\partial \omega} \right)_{i=1}^n$$

        сделать шаг в сторону антиградиента

$$\omega^{(t+1)} = \omega^{(t)} - \alpha \cdot \nabla_{\omega} Q(\omega^{(t)})$$

**end**

**end**

---

## Идея ускорения алгоритма

Градиент  $\nabla Q(\omega)$  представим в виде суммы градиентов:

$$\nabla Q(\omega) = \frac{1}{\ell} \sum_{i=1}^{\ell} \nabla \mathcal{L}(\omega)$$

Идея состоит в том, чтобы вычислять не точное значение градиента по всей выборке  $X$ , а оценить по некоторой подвыборке  $X' \subset X, |X'| = k \ll \ell$  небольшого размера.

$$\nabla Q(\omega) \approx \frac{1}{k} \sum_{i=1}^k \nabla \mathcal{L}(\omega)$$

## Метод стохастического градиента

---

### Algorithm 2 Метод стохастического градиента

---

**Input:**  $k$  - размер подвыборки,  $\alpha$  - градиентный шаг

**Output:**  $\omega^*$  - оптимум функционала  $Q(\omega)$

**begin**

    Инициализировать  $\omega^{(0)}$

**while** не выполнен критерий остановки **do**

        выбрать набор  $X'$ ,  $|X'| = k$

        вычислить градиент в точке по подвыборке  $X'$

$$\nabla Q(\omega)|_{\omega=\omega^{(t)}} = \left( \frac{\partial Q(\omega)}{\partial \omega} \right)_{i=1}^n$$

        сделать шаг в сторону антиградиента

$$\omega^{(t+1)} = \omega^{(t)} - \alpha \cdot \nabla_{\omega} Q(\omega^{(t)})$$

**end**

**end**