Лекция 2. Восстановление регрессии

Основы интеллектуального анализа данных

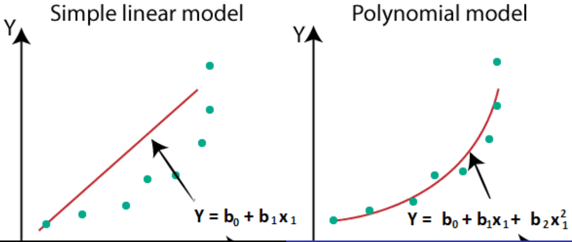
Полузёров Т. Д.

БГУ ФПМИ

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 $y^* : \mathbb{X} \to \mathbb{Y}$

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):

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

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

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 θ, 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**

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

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

Для упрощения формул добавим к признаковому описанию объектов признак равный единице

$$x := (1, x_1, ..., x_n), \;\; \omega := (\omega_0, \omega_1, ..., \omega_n)$$

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 ω^*
- **неустойчивое обучение** - добавление или удаление нескольких объектов из X влечет значительно разные оптимальные ω^*
- **решение неустойчиво** - малое изменение входных данных влечет сильное изменение значения функции регрессии

Методы борьбы с мультиколлинеарностью

Для борьбы с мультиколлинеарностью можно:

- удалять скоррелированные столбы
- вводить ограничения на параметры
- добавить штраф в фунционале 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 α - 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):

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

параметр β - селективность.

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

Влияние

МНК в случае линейной модели

- Модель регрессии - линейная, $a(x) = \langle \omega, x \rangle$
- Функция потерь - квадратичная, $\mathcal{L}(a, x_i) = (a(x_i) - y_i)^2$
- Метод обучения - минимизация среднего риска, $Q(\omega) = \sum_{i=1}^{\ell}(a(x_i, \omega) - y_i)^2 \to \min\limits_{\omega}$

В матричном виде:

$$Q(\omega) = \frac{1}{\ell}\|X\omega - y\|^2 \to \min_{\omega}$$

Точное аналитическое решение

Задача имеет точное аналитическое решение:

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

Недостатки аналитического решения:

- Обращение матрицы $(X^T X)^{-1}$: в случае плохо обусловленной матрицы веса неустойчивы и очень большие по модулю. Для вырожденной матрица - обращение невозможно.
- Вычислительная сложность - $O(n^2\ell + n^3)$

Численное решение

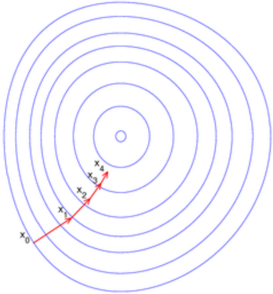
Наиболее простой и подходящий класс методов – **градиентные методы оптимизации**.

Общая схема:

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

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

где $a \in \mathbb{R}$
- некоторый параметр (размер шага)



Градиент квадратичного функционала:

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

Метод градиентного спуска

Algorithm 1 Метод градиентного спуска

Input: α - градиентный шаг (темп обучения)
Output: ω^* - оптимум функцмонала $Q(\omega)$
begin
  Инициализировать $\omega^{(0)}$
  while *не выполнен критерий остановки* **do**
    вычислить градиент в точке
    $\nabla Q(\omega)\big|_{\omega=\omega^{(t)}} = \left(\frac{\partial Q(\omega)}{\partial \omega}\right)_{i=1}^{n}$
    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 но некоторой подвыборке $X^{'} \subset X, |X^{'}| = k \ll \ell$ небольшого размера.

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

Метод стохастического градиента

Algorithm 2 Метод стохастического градиента

Input: k - размер подвыборки, α - градиентный шаг
Output: ω^* - оптимум функцмонала $Q(\omega)$
begin
Инициализировать $\omega^{(0)}$
while *не выполнен критерий остановки* **do**
выбрать набор $X^{'}, |X^{'}| = k$

вычислить градиент в точке по подвыборке $X^{'}$
$\nabla Q(\omega)\big|_{\omega=\omega^{(t)}} = \left(\frac{\partial Q(\omega)}{\partial \omega}\right)_{i=1}^{n}$

сделать шаг в сторону антиградиента
$\omega^{(t+1)} = \omega^{(t)} - \alpha \cdot \nabla_{\omega} Q(\omega^{(t)})$
end
end

Другие популярные градиентные методы

- SAG
- Метод инерции (momentum)
- AdaGrad, RMSprop
- Adam