Лекция 2. Восстановление регрессии

Основы интеллектуального анализа данных

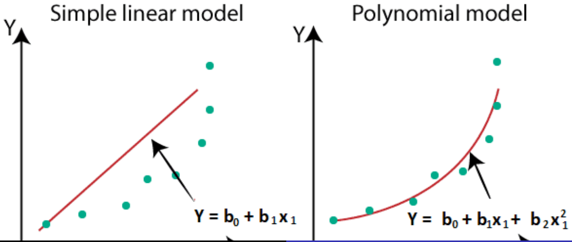
Полузёров Т. Д.

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 $f(x, \theta), \theta \in \Theta$ приблизить y^* **на всем множестве** $\mathbb{X}$.



Метод наименьших квадратов

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

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

В случае дифференцируемости f, решение находится из уравнения:

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

Линейная регрессия

Частный случай, когда $f(x,\theta)$ 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
$x := (1, x_1, ..., x_n), \omega = (\omega_0, \omega_1, ..., \omega_n)$

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

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

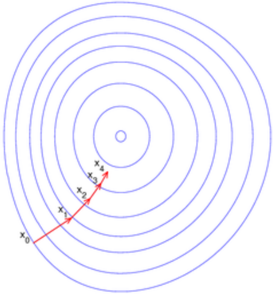
Наиболее простой и подходящий класс методов – **градиентные методы оптимизации**.

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

Другие популярные градиентные методы

Другие популярные градиентные методы