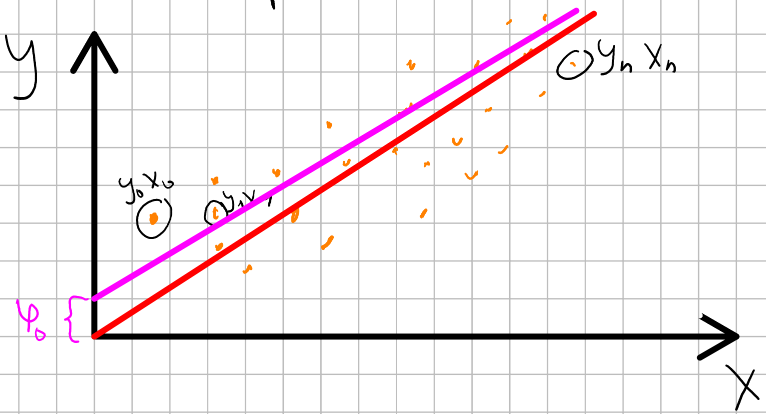
Python в инженерных расчётах.

Лекция #10. Линейная регрессия

Машинное обучение:

- Линейная регрессия
- Кластеризация данных
- Дерево принятия решений
- Нейронная сеть

} Sklearn библиотека



$$\vec{y} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix} \qquad \vec{x} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix}$$

$\boxed{y = \varphi \cdot x}$ — одномерная модель

$y = \varphi_0 + \varphi_1 \cdot x$

$$X = \begin{pmatrix} 1 & x_0 \\ 1 & x_1 \\ 1 & x_2 \\ 1 & \vdots \\ 1 & x_n \end{pmatrix}$$

Многомерная модель:

$$\vec{y} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix} \qquad X = \begin{pmatrix} x_{00} & x_{10} & \dots & x_{k0} \\ x_{01} & x_{11} & & x_{k1} \\ \vdots & \vdots & & \vdots \\ x_{0n} & x_{1n} & & x_{kn} \end{pmatrix}$$

$\boxed{y = \varphi_0 x_0 + \varphi_1 x_1 + \varphi_2 x_2 \dots \varphi_k x_k}$ k-мерные ур-ие

X — матрица признаков

y — вектор переменных

φ — вектор коэф (параметров)

$\boxed{\hat{y} = f(\vec{x}, \vec{\varphi})}$ $\qquad \hat{y}_n = f(\vec{x}_n, \vec{\varphi})$

Задача регрессии: найти наилучшие параметры φ, которые наилучшим образом опишут $y(\vec{x})$

Почему мы работаем с линейной моделью (как правило)

- Легко решается математически
- Легко анализируются результаты
- при больших K линейные модели очень хорошо описывают реальность

$y = \varphi_0 + \varphi_1 x$ $K = 1$

σ C

$y = \varphi_0 x_0 + \varphi_1 x_1 + \varphi_2 x_2 + \varphi_3 x_3 + \varphi_4 x_4$ $\uparrow K = 4$

$x_0 = 1$ C Mn Cr Ni

* Не линейная модель:

$y = \varphi_0 + \varphi_1 x + \varphi_2 x^2 + \varphi_3 x^3$

φ ищем с помощью ф-ии потерь (cost function)

$$\boxed{E(\vec{\varphi}) = \sum_{n=1}^{N} |y_n - \hat{y}_n|^2 = \sum_{n=1}^{N} |y_n - f(\vec{x}_n; \vec{\varphi})|^2}$$

n - кол-во экспериментальных данных

y_n - реальное значение y в образце # n.

$\hat{y}_n$ - расчетное значение y

$$\boxed{\varphi^* = \underset{\varphi}{\operatorname{argmin}} \sum_{n=1}^{N} |y_n - f(\vec{x}_n; \vec{\varphi})|^2}$$

наилучший набор φ метод наименьших квадратов

Определение нормы вектора:

$$||v|| = \sqrt{\sum_{i=1}^{n} v_i^2}$$

$$\vec{\varphi}^* = \underset{\varphi}{\arg\min} \left\| \underbrace{\begin{pmatrix} y_0\\ y_1\\ y_2\\ \vdots\\ y_N\end{pmatrix}}_{\vec{y}} - \underbrace{\begin{pmatrix} \leftarrow x_0 \rightarrow \\ x_1 \\ \vdots \\ x_N\end{pmatrix}}_{X} \cdot \underbrace{\begin{pmatrix}\varphi_0\\ \varphi_1\\ \vdots \\ \varphi_N\end{pmatrix}}_{\varphi} \right\|^2$$

$$\varphi^* = \underset{\varphi}{\arg\min} \| X \cdot \vec{\varphi} - \vec{y} \|^2$$

$$E(\varphi) = \| X \cdot \vec{\varphi} - \vec{y} \|^2$$

$$\boxed{\frac{\partial E}{\partial \varphi} = \frac{2}{N} \cdot X^T \cdot \left(X \cdot \vec{\varphi} - \vec{y} \right)}$$

градиент cost function

- Градиентный спуск
- Аналитическое решение

$$\cancel{\frac{2}{N}} \cdot X^T \cdot (X \cdot \vec{\varphi} - \vec{y}) = 0$$

$$X^T X \cdot \vec{\varphi} = X^T \cdot y \quad \Big| \cdot (X^T \cdot X)^{-1}$$

$$\boxed{\vec{\varphi} = \underbrace{(X^T \cdot X)^{-1}} X^T \cdot y}$$

Д.З. 1. Посмотрите тесты по библиотеке Matplotlib

2. Взять файл lection10_02.txt и по данным найти $\boxed{\varphi_0, \varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6, \varphi_7}$

без графиков