正规方程

正规方程

最小二乘法矩阵表示

最小二乘法可以将误差方程转化为有确定解的**代数方程组**（其方程式数目正好等于未知数个数），从而可求解出这些未知参数。这个有确定的代数方程组称为最小二乘估计的**正规方程**。公式如下

$$（1）\ \theta = (X^T X)^{-1} X^T \quad 或者 \quad W = (X^T X)^{-1} X^T y \tag{1}$$

其中的W、θ即是方程的解，也就是斜率和截距

公式是如何推导的?

最小二乘法公式如下：

$$（2）\ J(\theta) = \frac{1}{2} \sum_{i=0}^{n} (h_\theta(x_i) - y_i)^2 \tag{2}$$

$h_\theta(x_i)$中，θ代表系数，上文中我们使用W表示 (3)

而整个$h_\theta(x_i)$表示最终算法计算估计的值（预测值）也就是$\hat{y}$ (4)

y_i表示第$_i$个真实值 (5)

根据这个方程就可以推导出正规方程，也就是上述公式（1）

使用矩阵表示：

$$J(\theta) = \frac{1}{2} \sum_{i=0}^{n} (h_{\theta(x_i)} - y)(h_{\theta(x_i)} - y) \tag{6}$$

$$J(\theta) = \frac{1}{2} (X\theta - y)^T (X\theta - y) \tag{7}$$

A是$m * n$的矩阵，那么$A^T A$也就是$n \times n$，这被叫做对称半正定矩阵 (8)

矩阵当中是没办法直接A^2的，因为矩阵乘法是第一个矩阵的行、乘第二个矩阵的列 (9)

想要对应数据相乘，就必须转置 (10)

$J(\theta)$是损失函数 (11)

举例

一元二次：

$$\begin{cases} x+y=14 \\ 2x-y=10 \end{cases} \tag{1}$$

$$\begin{gathered} \text{其中，令} X=\begin{pmatrix} 1 & 1 \\ 2 & -1 \end{pmatrix} \\ y=\begin{pmatrix} 14 \\ 10 \end{pmatrix} \\ X\text{是系数矩阵、}y\text{是目标向量} \end{gathered} \tag{2}$$

根据（1）$\theta=(X^TX)^{-1}X^T$　或者　$W=(X^TX)^{-1}X^Ty$

我们就可以求解出最终θ或者W：

代码块

```
X = np.array([[1,1],[2,-1]])
y = np.array([14,10])
w = np.linalg.inv(X.T.dot(X)).dot(X.T).dot(y)
print(w)
# array([8,6])
# .inv就是逆
```

矩阵转置公式与求导公式（为推导做准备）

转置公式：

$(mA)^T = mA^T$

$(A+B)^T = A^T + B^T$

$(AB)^T = B^T A^T$

$(A^T)^T = A$

求导公式：

$\frac{\partial X^T}{\partial X} = I$求解出来是单位矩阵

$\frac{\partial X^T A}{\partial X} = A$

$\frac{\partial AX^T}{\partial X} = A$

$\frac{\partial AX}{\partial X} = A^T$

$\frac{\partial XA}{\partial X} = A^T$

$\frac{\partial X^T AX}{\partial X} = (A + A^A)X; A$不是对称矩阵

$\frac{\partial X^T AX}{\partial X} = 2AX; A$是对称矩阵

推导正规方程

原公式：

$$J(\theta) = \frac{1}{2}\sum_{i=0}^{n}(h_\theta(x_i) - y_i)^2$$

平方，代表他是一个凹的二次函数，存在低谷

1. 矩阵乘法公式展开：

$$\begin{aligned} J(\theta) &= \frac{1}{2}(X\theta - y)^T(X\theta - y) \\ &= \frac{1}{2}(\theta^T X^T - y^T)(X\theta - y) \\ &= \frac{1}{2}(\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y) \end{aligned}$$

2. 进行求导（注意X、y是已知量，θ是未知数）：

$$J'(\theta) = \frac{1}{2}(\theta^T X^T X\theta - \theta^T X^T y - y^T X\theta + y^T y)'$$

3. 根据上边的公式进行推导运算

$$
\begin{aligned}
J'(\theta) &= \frac{1}{2}(X^T X\theta - (\theta^T X^T X)^T - X^T y - (y^T X)^T) \\
&= \frac{1}{2}(X^T X\theta + X^T X\theta - X^T y - X^T y) \\
&= \frac{1}{2}(2X^T X\theta - 2X^T y) \\
&= X^T X\theta - X^T y \\
&= X^T(X\theta - y)
\end{aligned} \tag{12}
$$

、矩阵分配律运算

二次型求导：$\nabla\theta(\theta^T A\theta) = (A + A^T)\theta$
线性项求导：$\nabla\theta(b^T\theta) = b$
常数项求导：$\nabla\theta(c) = 0$

这里二次型是第一项，一次型是第二、三项，常数型是最后一项。

4. 令导数 $J'(\theta) = 0$ （导数为零最小）：

$$
\begin{aligned}
0 &= X^T X\theta - X^T y \\
X^T X\theta &= X^T y
\end{aligned}
$$

5. 矩阵没有除法，使用逆矩阵进行转化：

$$
(X^T X)^{-1} X^T X\theta = (X^T X)^{-1} X^T y \tag{13}
$$

$$
I\theta = (X^T X)^{-1} X^T y \tag{14}
$$

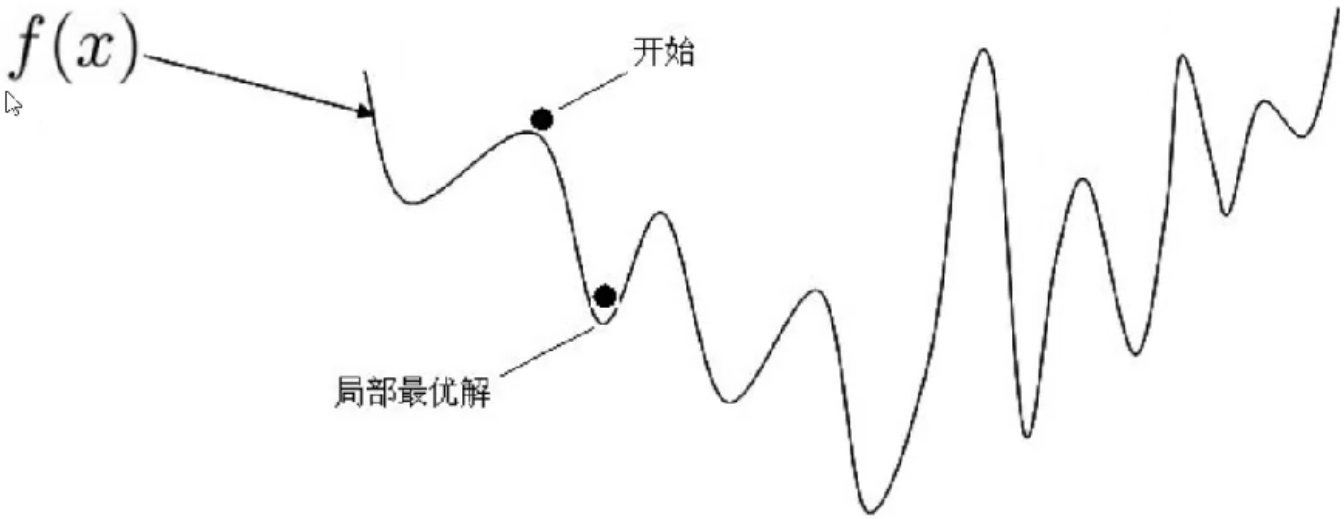
$$
\theta = (X^T X)^{-1} X^T y \tag{15}
$$

这就是正规方程！

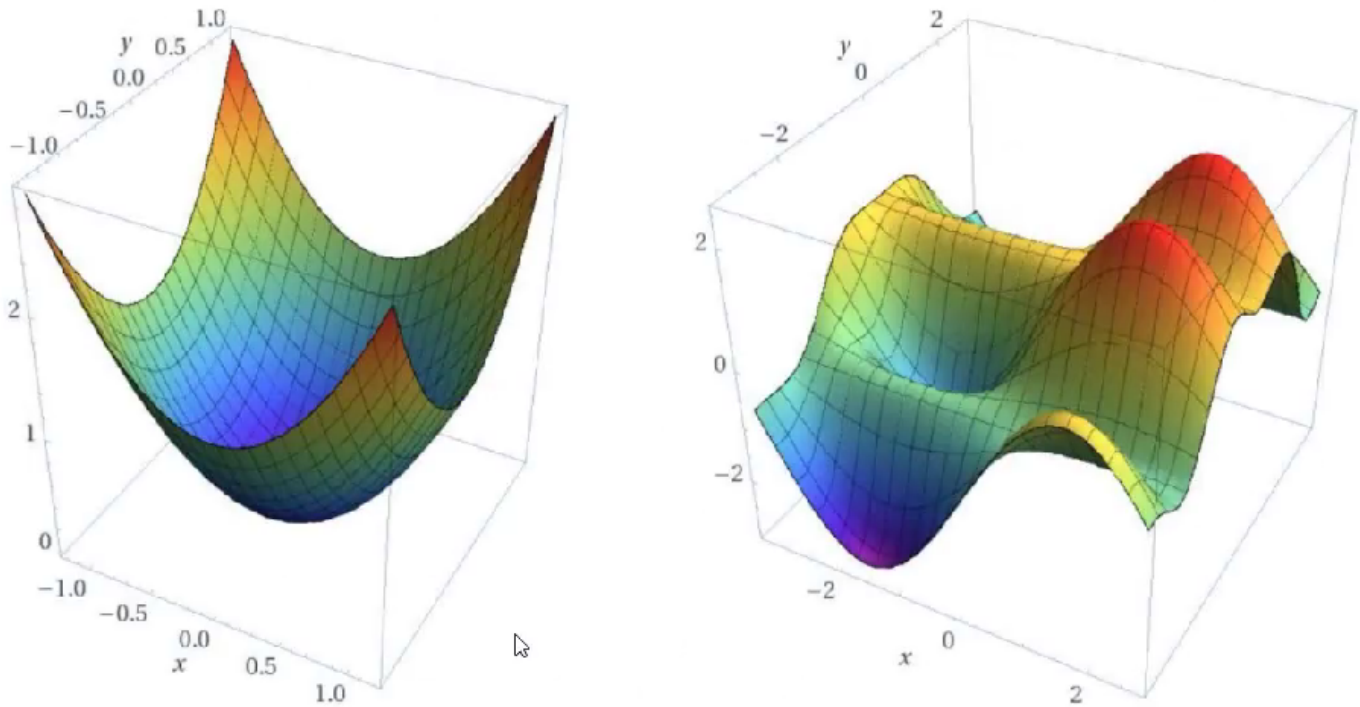
凸函数判定

判断损失函数是凸函数的好处在于我们可能很肯定的指导我们求得的机制即是最优解，一定是全局最优解。

但如果是非凸函数，那就不一定可以获取全局最优解：



立体图：



这里需要注意的是，在国内的普遍认为函数的凹凸是象形的，也就是凹凸的长相就是字的长相，但是由于人工智能大部分算法是国外发明的，所以在凹凸上会对调。

判定凸函数的方式：判定凸函数的方式有很多，其中一个方式是看**黑塞矩阵**是否是**半正定**的。

黑塞矩阵（hessian matrix）是由目标函数在点 X 处的二阶偏导数组成的对称矩阵。

对于我们的式子来说就是在导函数的基础上再次对 θ 进行**求偏导**，结果就是 $X^T X$ 。所谓的正定就是 $X^T X$ 的特征值全为正整数，半正定就是 $X^T X$ 的特征值大于等于0，就是半正定，凸函数。

$$J'(\theta) = X^T X\theta - X^T y \tag{16}$$
$$J''(\theta) = X^T X \tag{17}$$
$$\tag{18}$$
第一项：一阶求导，就是$X^T X$ (19)
第二项：常数阶求导，就是0 (20)

这里我们对 $J(\theta)$ 算是函数进行**求二阶导数**的黑塞矩阵是 $X^T X$ ，得到的一定是半正定的，自己和自己做点乘就是。

因为是 $X^T X$ ，你会发现，它还是平方数，也就是大于等于 0，也就是凸函数

这里不用数学推导证明这一点。在机器学习中往往损失函数都是**凸函数**，到**深度学习**中损失函数往往是**非凸函数**，即找到的解围殴鄙视全局最优解，只要模型堪用就好！机器学习特点就是：不强调模型100%正确，只要是有价值的，堪用的，就好