# 简单线性回归算法公式推导

最小二乘法求解：

本质是试图找到一条直线，使得样本上的点到直线的欧式距离之和最小

对求导并且令 ,

(一)

化简

两边同除m

（二）

化简并将b带入

因为

所以

**模型输出**

# 多元线性回归公式推导

找出,使得

模型输出

参数

|  |  |  |  |
| --- | --- | --- | --- |
| 超参数 | 解释 | 数值类型(默认值) | optional |
| **fit\_intercept** | 是否计算此模型的截距。 如果设置为 False，则不会在计算中使用截距（例如，预计数据已经居中 | Boolean（True） | True/False |
| **normalize** | fit\_intercept 设置为 False 时，将忽略此参数。 如果为真，则回归量 X 将在回归之前通过减去平均值并除以 l2 范数来归一化。 如果您希望标准化，请 在使用估算器[sklearn.preprocessing.StandardScaler](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler)之前fit使用normalize=False。 | Boolean（False） | True/False |
| **copy\_X** | 如果为 True，则将复制 X; 否则，它可能会被覆盖。(不太懂) | Boolean（True） | True/False |
| **n\_jobs** | 确定 cpu 的核数，-1 表示使用所有处理器 | Int（ None） |  |

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 解释 | 类型 | Shape |
| **coef\_** | 回归系数 (斜率) | array | （n\_features，）（n\_targets，n\_features） |
| **intercept\_** | 截距 | Float |  |

|  |  |  |
| --- | --- | --- |
| 方法 | 解释 | 类型 |
| fit(X, y, sample\_weight=None) | 训练线性模型 | X : array-like or 稀疏矩阵，y : array\_like，sample\_weight : numpy array |
| [get\_params](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression.get_params) (deep=True) | 如果为 True，将返回此估计器的参数并包含作为模型的子对象。 | ***boolean*** |
| [predict](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression.predict)(X) | 使用线性模型预测 | X : array-like or 稀疏矩阵 |
| [score](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression.score)(X, y[, sample\_weight]) | 评分标准 | float |
| [set\_params](http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression.set_params)(\*\*params) | 设置参数 |  |