

监督学习

ML17

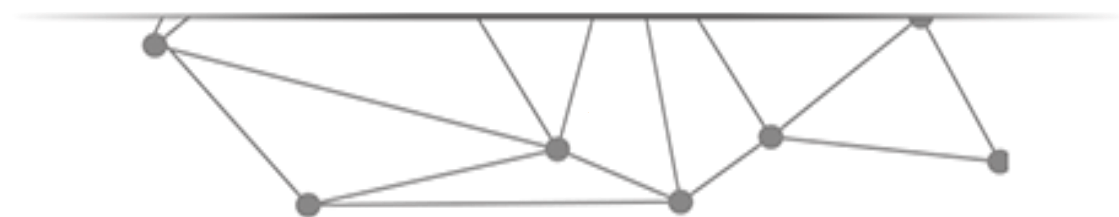


礼欣

www.python123.org



多项式回归+房价与房屋尺寸关系的非线性拟合



多项式回归

- 多项式回归(Polynomial Regression)是研究一个因变量与一个或多个自变量间多项式的回归分析方法。如果自变量只有一个时，称为一元多项式回归；如果自变量有多个时，称为多元多项式回归。

- 一元m次多项式回归方程为：

$$\hat{y} = b_0 + b_1x + b_2x^2 + \cdots + b_mx^m$$

- 二元二次多项式回归方程为：

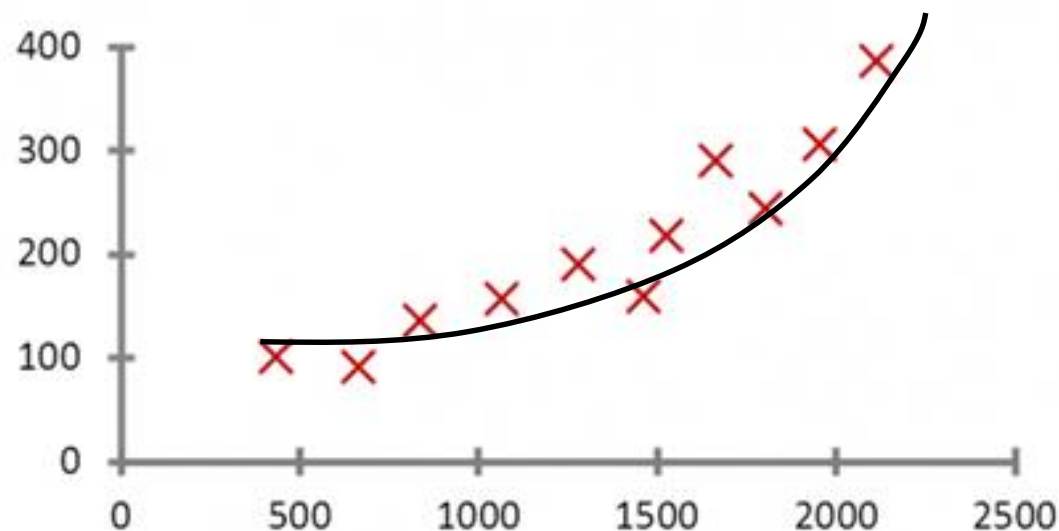
$$\hat{y} = b_0 + b_1x_1 + b_2x_2 + b_3x_1^2 + b_4x_2^2 + b_5x_1x_2$$

多项式回归

- 在一元回归分析中，如果依变量 y 与自变量 x 的关系为非线性的，但是又找不到适当的函数曲线来拟合，则可以采用一元多项式回归。
- 多项式回归的最大优点就是可以通过增加 x 的高次项对实测点进行逼近，直至满意为止。
- 事实上，多项式回归可以处理相当一类非线性问题，它在回归分析中占有重要的地位，因为任一函数都可以分段用多项式来逼近。

多项式回归

之前提到的线性回归实例中，是运用直线来拟合数据输入与输出之间的线性关系。不同于线性回归，多项式回归是使用曲线拟合数据的输入与输出的映射关系。



多项式回归的应用

应用背景：我们在前面已经根据已知的房屋成交价和房屋的尺寸进行了线性回归，继而可以对已知房屋尺寸，而未知房屋成交价格的实例进行了成交价格的预测，但是在实际的应用中这样的拟合往往不够好，因此我们在此对该数据集进行多项式回归。

目标：对房屋成交信息建立多项式回归方程，并依据回归方程对房屋价格进行预测

技术路线：`sklearn.preprocessing.PolynomialFeatures`

实例数据

成交信息包括房屋的面积以及对应的成交价格：

- 房屋面积单位为平方英尺 (ft^2)
- 房屋成交价格单位为万

编号	房屋面积/ ft^2	成交价格/万	编号	房屋面积/ ft^2	成交价格/万
1	1000	168	26	2700	285
2	792	184	27	2612	292
3	1260	197	28	2705	482
4	1262	220	29	2570	462
5	1240	228	30	2442	352
6	1170	248	31	2387	440
7	1230	305	32	2292	462
8	1255	256	33	2308	325
9	1194	240	34	2252	298
10	1450	230	35	2202	352
11	1481	202	36	2157	403
12	1475	220	37	2140	308
13	1482	232	38	4000	795
14	1484	460	39	4200	765
15	1512	320	40	3900	705
16	1680	340	41	3544	420
17	1620	240	42	2980	402
18	1720	368	43	4355	762
19	1800	280	44	3150	392
20	4400	710	45	3025	320
21	4212	552	46	3450	350
22	3920	580	47	4402	820
23	3212	585	48	3454	425
24	3151	590	49	890	272
25	3100	560			

实验过程

使用算法：**线性回归**

实现步骤：


1. 建立工程并导入sklearn包
2. 加载训练数据，建立回归方程
3. 可视化处理

关于一些相关包的介绍：

- NumPy是Python语言的一个扩充程序库。支持高级大量的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库。
- matplotlib的pyplot子库提供了和matlab类似的绘图API，方便用户快速绘制2D图表。

实现步骤——1.建立工程并导入sklearn包

- 创建house.py文件
- 导入sklearn相关包

- `import matplotlib.pyplot as plt`  matplotlib的pyplot子库，它提供了和matlab类似的绘图API。
- `import numpy as np`
- `from sklearn import linear_model`
- `from sklearn.preprocessing import PolynomialFeatures`



导入线性模型和多项式特征构造模块

实现步骤——1.建立工程并导入sklearn包

sklearn中多项式回归：

这里的多项式回归实际上是先将变量 x 处理成多项式特征，然后使用线性模型学习多项式特征的参数，以达到多项式回归的目的。

例如： $X = [x_1, x_2]$

1.使用PolynomialFeatures构造 X 的二次多项式特征 X_{Poly} ：

$$X_{Poly} = [x_1, x_2, x_1x_2, x_1^2, x_2^2]$$

2.使用linear_model学习 X_{Poly} 和 y 之间的映射关系，即参数：

$$w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4x_1^2 + w_5x_2^2 = y$$

实现步骤——2.加载训练数据，建立回归方程

- `datasets_X = []`
 - `datasets_Y = []`
 - `fr = open('prices.txt','r')`
 - `lines = fr.readlines()`
 - `for line in lines:`
 - `items = line.strip().split(',')`
 - `datasets_X.append(int(items[0]))`
 - `datasets_Y.append(int(items[1]))`
 - `length = len(datasets_X)`
 - `datasets_X = np.array(datasets_X).reshape([length,1])`
 - `datasets_Y = np.array(datasets_Y)`
- 建立`datasets_X`和`datasets_Y`用来存储数据中的房屋尺寸和房屋成交价格。
- 打开数据集所在文件 `prices.txt`，读取数据。
- 一次读取整个文件。

实现步骤——2.加载训练数据，建立回归方程

- `datasets_X = []`
- `datasets_Y = []`
- `fr = open('prices.txt','r')`
- `lines = fr.readlines()`
- `for line in lines:` → 逐行进行操作，循环遍历所有数据
- `items = line.strip().split(',')` → 去除数据文件中的逗号
- `datasets_X.append(int(items[0]))`
- `datasets_Y.append(int(items[1]))` → 将读取的数据转换为int型，并分别写入
datasets_X和datasets_Y。
- `length = len(datasets_X)`
- `datasets_X = np.array(datasets_X).reshape([length,1])`
- `datasets_Y = np.array(datasets_Y)`

实现步骤——2.加载训练数据，建立回归方程

- datasets_X = []
- datasets_Y = []
- fr = open('prices.txt','r')
- lines = fr.readlines()
- for line in lines:
- items = line.strip().split(',')
• datasets_X.append(int(items[0]))
• datasets_Y.append(int(items[1]))
- length = len(datasets_X) —————→ 求得datasets_x的长度，即为数据的总数。
- datasets_X = np.array(datasets_X).reshape([length,1]) —————→ 将datasets_x转化为数组，并变为二维，以符合线性回归拟合函数输入参数要求。
- datasets_Y = np.array(datasets_Y) —————→ 将datasets_Y转化为数组

实现步骤——2.加载训练数据，建立回归方程

- `minX = min(datasets_X)`
- `maxX = max(datasets_X)`
- `X = np.arange(minX,maxX).reshape([-1,1])`



以数据`datasets_X`的最大值和最小值为范围，建立等差数列，方便后续画图。

- `poly_reg = PolynomialFeatures(degree = 2)`
- `X_poly = poly_reg.fit_transform(datasets_X)`
- `lin_reg_2 = linear_model.LinearRegression()`
- `lin_reg_2.fit(X_poly, datasets_Y)`



`degree=2`表示建立`datasets_X`的二次多项式特征`X_poly`。然后创建线性回归，使用线性模型学习`X_poly`和`datasets_Y`之间的映射关系（即参数）。

实现步骤——3.可视化处理

- `plt.scatter(datasets_X, datasets_Y, color = 'red')`
- `plt.plot(X, lin_reg_2.predict(poly_reg.fit_transform(X)), color = 'blue')`
- `plt.xlabel('Area')`
- `plt.ylabel('Price')`
- `plt.show()`



`scatter`函数用于绘制数据点，这里表示用红色绘制数据点；

`plot`函数用来绘制回归线，同样这里需要先将`x`处理成多项式特征；

`xlabel`和`ylabel`用来指定横纵坐标的名称。

结果展示

通过多项式回归拟合的曲线与数据点的关系如右图所示。依据该多项式回归方程即可通过房屋的尺寸，来预测房屋的成交价格。

