

資料科學期末報告

資工三 408410086 常博愛

與期中報告相同，資料非取自 kaggle.

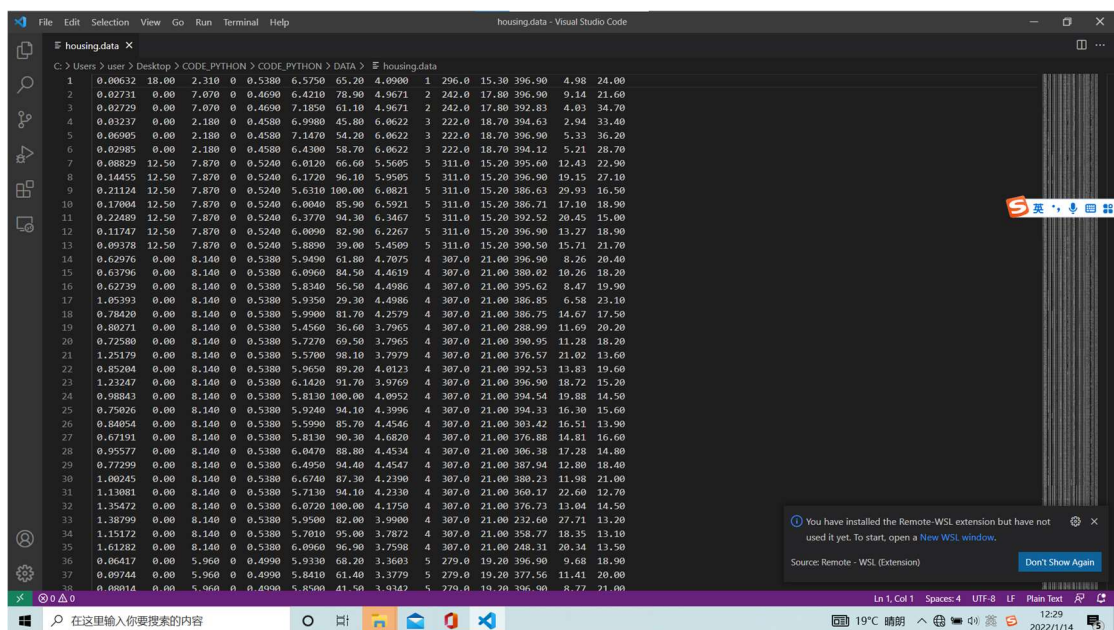
1. 問題陳述與動機：

陳述：以 Boston 為例，利用幾年來不同地區的房屋買賣與出租成交價預測 未來房屋價格走勢並分析決定價格高低的主要因素。

動機 隨著時代競爭力的激烈發展 年輕人購買一套屬於自己的房屋已經 越來越困難 同時在外工作，住宿問題同樣重要，所以通過分析資料提供一個好 的演算法以便現代年輕人可以選擇到合適的房屋並了解其定價規律非常重要。

2. 資料描述

在 Boston 的 B vector 中，代表 house 的房價，左邊的房屋特征 X_n 分別代表房屋的住房面積，臥室的個數，在那個小區，是否靠近高速公路，附近是否有學校，人口密度等一共 13 個因子。



3. 資料分析方法描述（演算法）：

Economy SVD, Least square regression.

我們要利用已知的 A 與 b 算出一個大致的 X 以滿足 $Ax=b$.其中利用 SVD 與 inverse(A) *b(pseudo inverse)求出大致擬合直線。

演算法介紹：

最小二乘法（英语：least squares method），又称**最小平方方法**，是一种数学优化建模方法。它通过最小化误差的平方和寻找数据的最佳函数匹配。

利用最小二乘法可以简便的求得未知的数据，并使得求得的数据与实际数据之间误差的平方和为最小。

“最小二乘法”是对线性方程组，即方程个数比未知数更多的方程组，以回归分析求得近似解的标准方法。在这整个解决方案中，最小二乘法演算为每一方程的结果中，将残差平方和的总和最小化。

最重要的应用是在曲线拟合上。最小二乘所涵义的最佳拟合，即残差（残差为：观测值与模型提供的拟合值之间的差距）平方总和的最小化。当问题在自变量

(x 变量) 有重大不确定性时, 那么使用简易回归和最小二乘法会发生问题; 在这种情况下, 须另外考虑变量-误差-拟合模型所需的方法, 而不是最小二乘法。

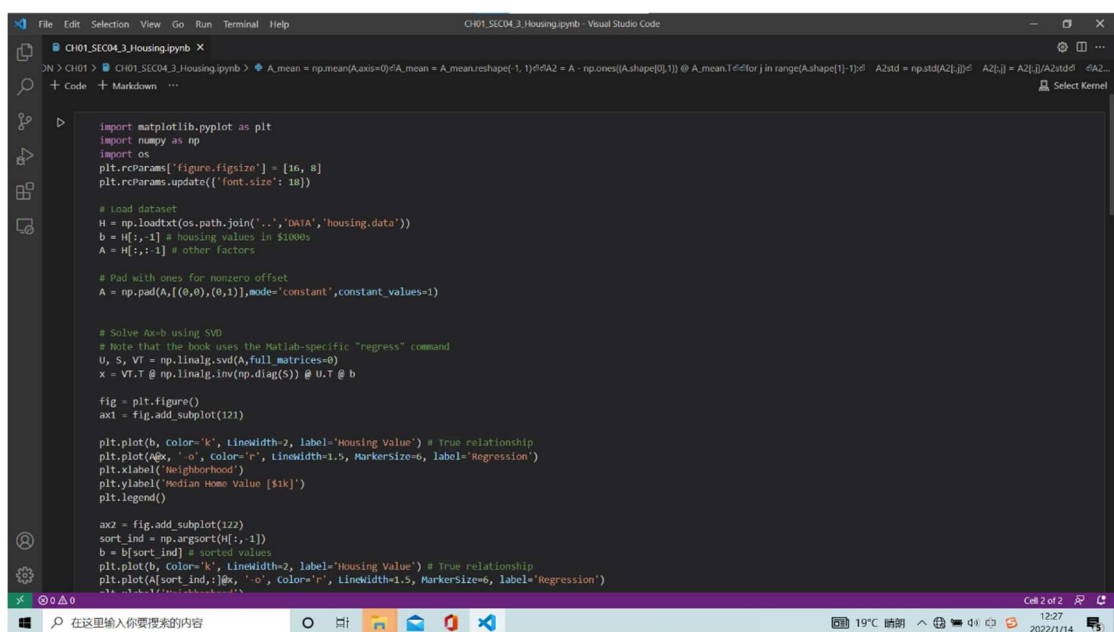
最小二乘问题分为两种: 线性或普通的最小二乘法, 和非线性的最小二乘法, 取决于在所有未知数中的残差是否为线性。线性的最小二乘问题发生在统计回归分析中; 它有一个封闭形式的解决方案。非线性的问题通常经由迭代细致化来解决; 在每次迭代中, 系统由线性近似, 因此在这两种情况下核心演算是相同的。

最小二乘法所得出的多项式, 即以拟合曲线的函数来描述自变量与预计应变量的方差关系。

当观测值来自指数族且满足轻度条件时, 最小二乘估计和最大似然估计是相同的。最小二乘法也能从动差法得出。

以下讨论大多是以线性函数形式来表示, 但对于更广泛的函数族, 最小二乘法也是有效和实用的。此外, 迭代地将局部的二次近似应用于或然性 (借由费希尔信息), 最小二乘法可用于拟合广义线性模型。

結果:



```
File Edit Selection View Go Run Terminal Help
CH01_SECM4.3.Housing.ipynb - Visual Studio Code

CH01_SECM4.3.Housing.ipynb X
ON > CH01 > CH01_SECM4.3.Housing.ipynb > A_mean = np.mean(A,axis=0);A_mean = A_mean.reshape(-1,1);dA2 = A - np.ones((A.shape[0],1)) @ A_mean.T;dA2 = A2std = np.std(A2[:,j]); A2[:,j] = A2[:,j]/A2std; dA2...
+ Code + Markdown ...
Select Kernel

import matplotlib.pyplot as plt
import numpy as np
import os
plt.rcParams['figure.figsize'] = [16, 8]
plt.rcParams.update({'font.size': 18})

# Load dataset
H = np.loadtxt(os.path.join('.', 'DATA', 'housing.data'))
b = H[:,1] # housing values in $1000s
A = H[:,2:1] # other factors

# Pad with ones for nonzero offset
A = np.pad(A,[(0,0),(0,1)],mode='constant',constant_values=1)

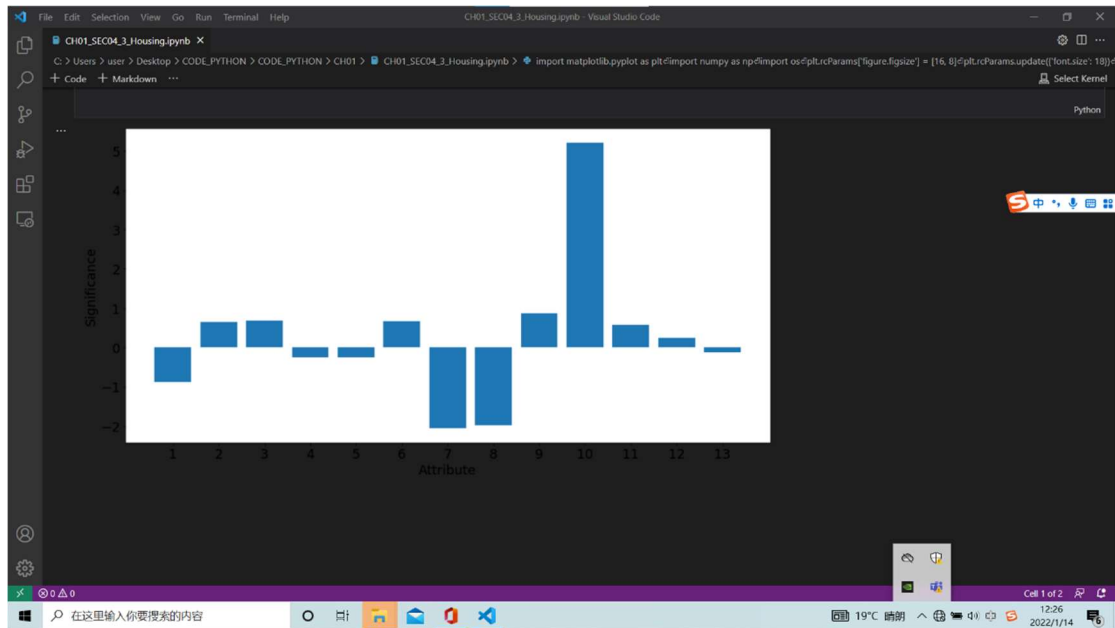
# Solve Ax=b using SVD
# Note that the book uses the Matlab-specific "regress" command
U, S, VT = np.linalg.svd(A,full_matrices=0)
x = VT.T @ np.linalg.inv(np.diag(S)) @ U.T @ b

fig = plt.figure()
ax1 = fig.add_subplot(121)

plt.plot(b, color='k', linewidth=2, label='Housing Value') # True relationship
plt.plot(A@x, '-o', color='r', linewidth=1.5, markersize=6, label='Regression')
plt.xlabel('Neighborhood')
plt.ylabel('Median Home Value [$1k]')
plt.legend()

ax2 = fig.add_subplot(122)
sort_ind = np.argsort(H[:,1])
b = b[sort_ind] # sorted values
plt.plot(b, color='k', linewidth=2, label='Housing Value') # True relationship
plt.plot(A[sort_ind,:].@x, '-o', color='r', linewidth=1.5, markersize=6, label='Regression')

Cell 2 of 2
12:27
2022/1/14
```

4. 心得：

- 1.總結討論 (Discussion)：通過資料處理，我們可以最佳化房屋綜合特征對於房價的線性影響，以及各個因素的正負相關影響比例等，然而，我們在選取資料時仍需注意 biased data.避免錯誤廢棄樣本對整體準確性的影響。
- 2.有助於我們快速評估選擇房屋，並預測未來一段時間內的房屋漲幅趨勢。
- 3.課程建議：希望能多 demo 一些應用演算法解決的實際案例。