

HW2

Lasso_beta_optimization

H24101222_陳凱騫

2024-11-30

Table of contents

1. Implement the coordinate descent method to identify the optimal tuning parameter for the LASSO penalty using a cross-validation approach, and then compute the LASSO estimates for the linear regression model.	1
2. Using the same tuning parameter, apply Particle Swarm Optimization (PSO) to calculate the LASSO estimates. Beyond the default PSO parameters, evaluate the impact of particle size and the number of iterations on the results.	3
演算法步驟	3
參數設置	4

1. Implement the coordinate descent method to identify the optimal tuning parameter for the LASSO penalty using a cross-validation approach, and then compute the LASSO estimates for the linear regression model.

首先生成一組模擬數據包含 $n = 50$ 筆觀測值及 $p = 200$ 個特徵，並設定前5個回歸係數為3、-3.5、4、-2.8、3.2，而其餘為零以模擬稀疏特性。

接著實作座標下降法 (Coordinate Descent)，以逐一更新回歸係數的方式解決 LASSO 的優化問題。然後使用 k -摺交叉驗證 ($k = 5$) 來選擇正則化參數 λ 。針對不同 λ 值計算驗證集的均方誤差 (MSE)，以選出最佳參數。

最後使用選出的最佳 λ 值，在完整資料上重新計算回歸係數。其結果如下：

```
[1] "Optimal Lambda: 0.5644444444444445"
```

```
[1] "Final Coefficients:"
```

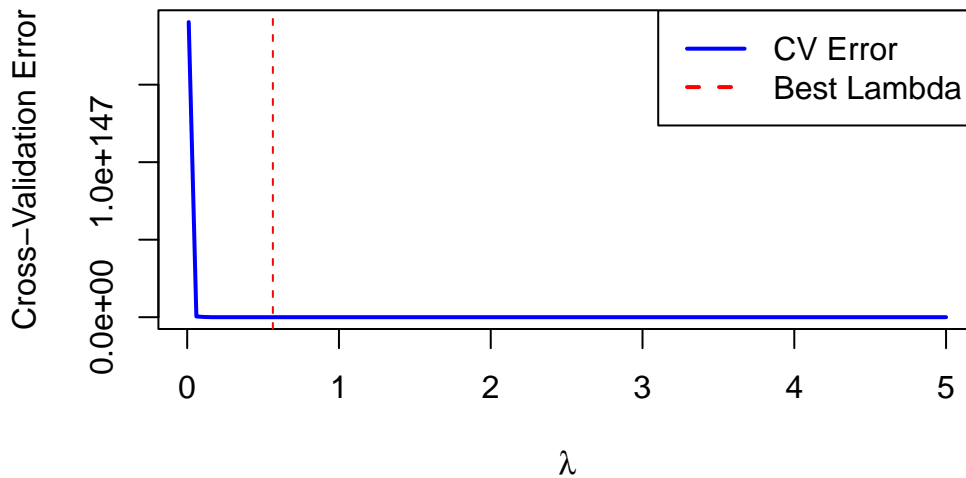
```
[1] 2.20655301 -2.76063850 3.31010139 -2.07957613 2.24463875 0.00000000  
[7] 0.00000000 0.00000000 -0.01214270 0.00000000 0.00000000 0.00000000  
[13] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000  
[19] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000  
[25] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000  
[31] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000  
[37] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000  
[43] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000  
[49] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
```

```

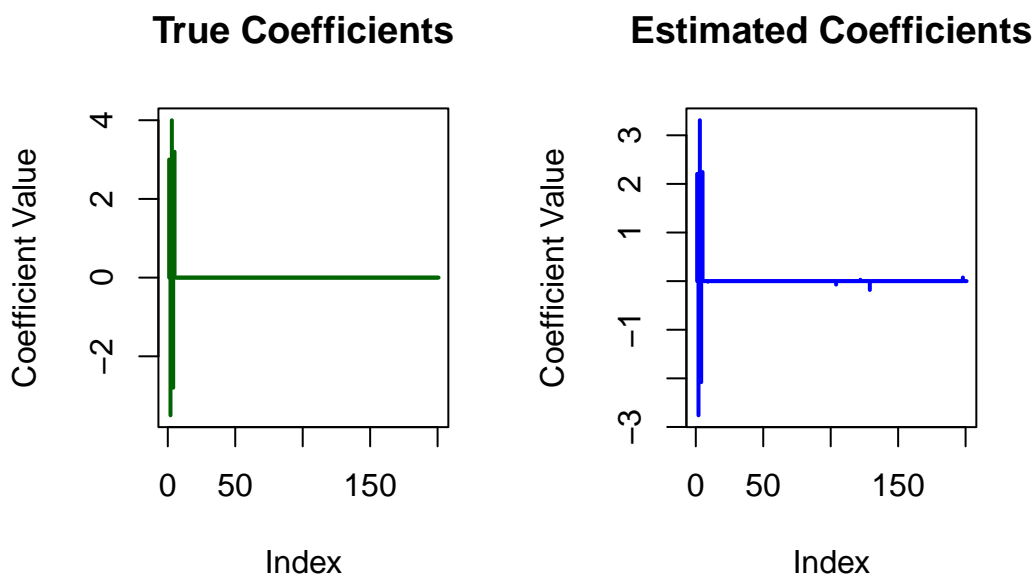
[55] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[61] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[67] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[73] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[79] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[85] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[91] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[97] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[103] 0.00000000 -0.07133204 0.00000000 0.00000000 0.00000000 0.00000000
[109] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[115] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[121] 0.00000000 0.02924180 0.00000000 0.00000000 0.00000000 0.00000000
[127] 0.00000000 0.00000000 -0.18163437 0.00000000 0.00000000 0.00000000
[133] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[139] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[145] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[151] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[157] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[163] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[169] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[175] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[181] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[187] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000
[193] 0.00000000 0.00000000 0.00000000 0.00000000 0.00000000 0.07514449
[199] 0.00000000 0.00000000

```

Cross-Validation for LASSO



從上方之圖可知 $\lambda = 0.5644$ 時可使 cv error 最小。



接著也可從上方之圖可看出其coordinate descent所選出來的大致是正確的，前五個beta都無趨近於0，其餘的大部分都接近0。

2. Using the same tuning parameter, apply Particle Swarm Optimization (PSO) to calculate the LASSO estimates. Beyond the default PSO parameters, evaluate the impact of particle size and the number of iterations on the results.

演算法步驟

1. 初始化：
 - 在解空間內隨機生成粒子的位置和速度。
 - 計算初始個人最佳和全局最佳分數。

2. 粒子更新：
 - 使用公式更新速度：

$$v_i = w \cdot v_i + c_1 \cdot r_1 \cdot (p_i - x_i) + c_2 \cdot r_2 \cdot (g - x_i)$$

其中：

- w ：慣性權重。
- c_1, c_2 ：個人與全局吸引力係數。
- r_1, r_2 ：隨機數。
- 根據速度更新位置，並檢查是否超出邊界。

3. 動態重置：
 - 如果粒子多次接觸邊界，則重置位置至全局最佳附近。
 - Note: 之所以如此設定是因為在無此設定之前，最後的到 β 之結果許多的 β 值卡在邊界。
4. 適應度評估：
 - 計算每個粒子的目標函數值。
 - 更新個人最佳與全局最佳。
5. 終止條件：
 - 達到最大迭代次數，或全局最佳分數在多次迭代內無顯著變化。

參數設置

- 粒子群大小：200。
- 最大迭代次數：10000。
- 位置邊界： $[-5, 5]$ 。
- 動態慣性權重：從 0.9 緩減至 0.4。(發現動態的效果比較好，可能因為平衡了全局與局部搜索的能力)

最終結果如下，可看到大部分都收斂到0，只有前5個 β 和少部分之值無收斂到0：

Iteration 1000, Best Score: 25.851905
Iteration 2000, Best Score: 12.701040
Iteration 3000, Best Score: 11.767462
Iteration 4000, Best Score: 11.465513
Iteration 5000, Best Score: 10.495879
Iteration 6000, Best Score: 10.370202
Iteration 7000, Best Score: 10.369861
Iteration 8000, Best Score: 10.369861
Iteration 9000, Best Score: 10.369861
Iteration 10000, Best Score: 10.369861

Optimized Coefficients (PSO-LASSO with Reset):

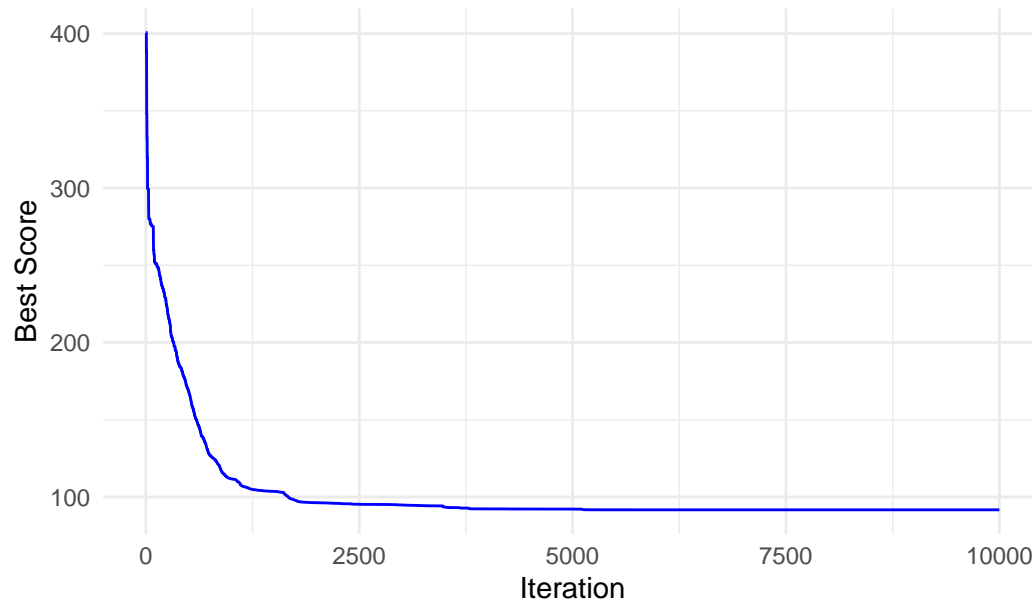
[1] 1.393643e+00 -3.093354e+00 3.110693e+00 -1.849919e+00 1.718469e+00
[6] -1.915779e-05 2.747184e-06 -1.161103e-08 -7.419120e-01 -7.071869e-07
[11] -1.401501e-10 -2.714492e-02 6.522209e-05 1.223210e-05 -4.169970e-07
[16] -4.741327e-07 9.975047e-07 -7.353702e-07 -8.007943e-05 2.812464e-01
[21] -6.638371e-05 1.298495e-05 7.205010e-07 -9.114125e-07 4.954967e-05
[26] -2.414045e-04 -1.032648e-01 6.476990e-04 -1.039458e-03 2.840033e-02
[31] -3.008034e-05 -3.993399e-10 -1.366197e-08 4.004338e-06 5.486971e-06
[36] -2.605157e-05 1.360766e-10 -1.178003e-06 2.966224e-03 4.739706e-05
[41] 4.201876e-06 2.520665e-05 1.240693e-05 -4.465969e-11 -6.542101e-02
[46] -6.916173e-06 9.305578e-04 2.981110e-06 -5.439859e-09 -2.235109e-15
[51] 8.023861e-09 2.915885e-05 1.977669e-09 3.292557e-06 -1.853258e-05
[56] -2.780993e-02 -1.107361e-04 -8.219632e-08 1.896313e-07 -2.408342e-04
[61] -6.004919e-08 -2.421562e-05 -1.770851e-06 -4.600583e-11 -4.535773e-02
[66] 7.120675e-04 1.643953e-06 5.678573e-08 -5.106552e-05 1.257318e-06
[71] 1.143056e-04 -2.019078e-04 3.224946e-08 4.722463e-06 1.107147e-10
[76] 3.143963e-08 -6.751172e-06 2.942490e-10 4.175803e-09 -2.556081e-01
[81] 3.118529e-06 -9.952971e-06 2.238897e-07 1.922120e-09 -6.430516e-03
[86] 1.337477e-01 1.038097e-06 4.497580e-05 -1.586985e-09 -9.464349e-06
[91] 1.626528e-05 3.216277e-04 7.722251e-10 1.249302e-08 -2.739605e-05
[96] -2.803411e-04 1.469751e-01 6.995104e-05 -3.253821e-06 -4.192447e-15
[101] -9.235062e-06 -4.664919e-05 1.564919e-05 -8.698756e-02 -1.063778e-05
[106] -8.158368e-09 -3.588154e-06 6.493985e-05 -1.371093e-08 -3.053824e-05
[111] -1.063659e-10 2.975813e-01 -1.769787e-07 4.768020e-08 1.235247e-05
[116] -5.581736e-07 -1.348856e-07 4.810460e-01 3.984387e-07 1.743362e-16
[121] -6.409040e-04 6.098610e-02 2.220540e-04 -1.840279e-05 9.000315e-06
[126] 3.577974e-05 -6.721093e-14 5.877404e-05 -3.399024e-01 3.407821e-11
[131] 1.931834e-14 -1.386213e-01 -3.813215e-05 -6.770265e-06 -1.122796e-04
[136] -8.650075e-10 -1.825910e-06 -1.257917e-09 -2.827265e-09 7.635623e-07
[141] -3.002339e-01 -7.250518e-06 4.023687e-13 -1.013352e-05 3.518076e-06
[146] -4.569221e-05 8.689707e-06 -2.239636e-13 1.874278e-06 -3.200819e-11
[151] 1.676153e-01 -1.734815e-01 -3.511214e-10 -2.144224e-05 4.517491e-04
[156] 4.347631e-05 -1.995448e-05 1.580661e-06 -3.092856e-07 3.482252e-05
[161] -1.440494e-01 1.925739e-15 -1.121280e-05 3.714706e-09 -2.873302e-05
[166] -8.215899e-06 -3.956032e-15 6.764011e-05 -6.732971e-06 1.928717e-10

```

[171] 2.974390e-14 -3.790123e-05 -4.155289e-01 8.922804e-15 -6.318067e-05
[176] 2.763773e-05 3.841608e-06 7.778859e-05 -2.520318e-07 -1.316217e-09
[181] 5.175623e-05 -1.615174e-13 2.259884e-05 2.890983e-09 6.656781e-06
[186] 9.465937e-07 2.352937e-08 -2.671252e-05 -8.633599e-10 -3.287820e-05
[191] -2.264878e-05 -2.752722e-09 -8.781520e-03 -1.269557e-05 -1.541819e-10
[196] 1.417282e-05 -1.291476e-04 1.380865e-01 -2.613233e-08 1.541570e-05

```

Global Best Score Over Iterations



而從上圖也可看到其Best Score有持續在降低。