

Machine Learning HW3

312512005 黃名諄

1. Linear SVM:

- alpha:

| | |
|-------|--|
| C=1 | <pre>linear_SVM(C=1) : alpha : [1. 0.0667 1. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0.0667 0. 0.0667 0. 1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0.2 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 0. 1. 0.] total sum of alpha : 14.4</pre> |
| C=10 | <pre>linear_SVM(C=10) : alpha : [0. 0. 9. 0. 0. 0. 10. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 9. 0. 0. 0. 8. 0. 0. 0. 10. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 0. 0. 10. 0.] total sum of alpha : 76.0</pre> |
| C=100 | <pre>linear_SVM(C=100) : alpha : [0. 0.0003 44.4438 0. 0.0004 0. 100. 0. 0. 0. 0. 0.0001 0. 0. 0. 0. 0.0003 0. 0.0003 0. 100. 0. 44.4438 0. 0. 0. 0. 0. 0.0001 0. 0. 100. 0.0001 0.0001 0. 0. 0. 0. 0. 0. 0. 0.0002 0. 0. 100. 0. 0. 0. 88.8884 0.] total sum of alpha : 577.7778</pre> |

- bias and CR(%) :

Linear SVM

| penalty weight C | bias | CR(%) |
|------------------|---------|---------|
| 1 | 10.5400 | 94.0000 |
| 10 | 15.1400 | 96.0000 |
| 100 | 11.0045 | 92.0000 |

2. RBF kernel-based SVM (C=10):

- alpha:

| | |
|------------|---|
| sigma=5 | <pre> nonlinear_SVM_RBF(C=10, sigma=5.00) : alpha : [10. 8.9618 10. 0. 10. 0. 10. 0. 0. 0. 0. 0. 0. 10. 0. 0. 8.9618 0. 8.9618 0. 10. 0. 10. 0. 0. 0. 10. 0. 6.595 0. 0. 10. 0. 0. 0. 10. 10. 0. 10. 0.2905 0. 10. 0. 0. 10. 0. 10. 0. 10. 0.] total sum of alpha : 193.771 </pre> |
| sigma=1 | <pre> nonlinear_SVM_RBF(C=10, sigma=1.00) : alpha : [0. 0. 8.8908 0. 0. 0. 10. 0.346 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 8.8908 0. 0. 0. 8.1275 0. 0. 0. 0. 10. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 0.0001 0. 10. 0.] total sum of alpha : 76.2552 </pre> |
| sigma=0.5 | <pre> nonlinear_SVM_RBF(C=10, sigma=0.50) : alpha : [0. 0. 5.9852 0. 0. 0. 5.3969 1.0033 0. 0. 0. 3.006 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 5.9852 0. 0. 0.2548 0. 0. 0. 0. 0. 10. 0.2937 0. 0.1434 0. 0. 0. 0. 0. 0. 0. 0. 0.5835 10. 0. 0.1012 0. 10. 0.] total sum of alpha : 62.7532 </pre> |
| sigma=0.1 | <pre> nonlinear_SVM_RBF(C=10, sigma=0.10) : alpha : [0.2801 0.3292 1.2411 0.3631 0.0704 0.703 0.7071 1.0221 0.0394 0.867 1.0131 1.0463 0.2517 0.2801 1.145 0.1056 0.3292 0.9249 0.3292 0.9196 2.5408 0.3631 1.2411 1.0054 0.9109 0.5211 0.1937 0.7048 0.4102 0.1378 0.415 1.0583 0.8405 0.7712 0.5245 0.537 0.7111 0.7242 0. 0.7778 0.7622 0.5161 0.5396 0.7948 2.3432 0.6959 0.6502 0.6137 2.2924 0.4936] total sum of alpha : 36.0576 </pre> |
| sigma=0.05 | <pre> nonlinear_SVM_RBF(C=10, sigma=0.05) : alpha : [0.5377 0.3213 0.6269 0.5453 0.9401 0.9632 1.0929 1.1103 0.9401 1.0907 1.1103 1.1105 0.9579 0.5377 1.1106 1.0552 0.3213 0.9809 0.3213 1.093 1.2539 0.5453 0.6269 1.093 1.0909 0.7834 0.783 0.8737 0.7831 0.8414 0.8574 0.8898 0.8894 0.8891 0.7834 0.692 0.889 0.8891 0.6734 0.8893 0.8893 0.7834 0.8733 0.8893 1.0591 0.8737 0.7977 0.8734 1.0587 0.8731] total sum of alpha : 42.7545 </pre> |

- bias and CR(%) :

Nonlinear SVM (RBF)

| penalty weight C | sigma | bias | CR(%) |
|------------------|-------|---------|---------|
| 10 | 5.00 | 0.0678 | 90.0000 |
| 10 | 1.00 | -0.2189 | 96.0000 |
| 10 | 0.50 | -0.2439 | 94.0000 |
| 10 | 0.10 | -0.1594 | 88.0000 |
| 10 | 0.05 | -0.1106 | 82.0000 |

3. Polynomial kernel-based SVM (C=10):

1. alpha:

| | |
|-----|---|
| p=1 | <pre> nonlinear_SVM_polynomial(C=10, p=1) : alpha : [0. 0. 9. 0. 0. 0. 10. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 9. 0. 0. 0. 8. 0. 0. 0. 10. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 0. 0. 10. 0.] total sum of alpha : 76.0 </pre> |
| p=2 | <pre> nonlinear_SVM_polynomial(C=10, p=2) : alpha : [0. 0. 3.8295 0. 0. 0. 9.4708 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 3.8295 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 0. 0. 7.1297 0.] total sum of alpha : 54.2594 </pre> |
| p=3 | <pre> nonlinear_SVM_polynomial(C=10, p=3) : alpha : [0. 0. 3.8735 0. 0. 0. 8.5547 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 3.8735 0.2324 0. 0. 0. 0. 0. 0. 0. 10. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 0. 0. 6.5342 0.] total sum of alpha : 53.0684 </pre> |
| p=4 | <pre> nonlinear_SVM_polynomial(C=10, p=4) : alpha : [0. 0. 3.2052 0. 0. 0. 6.111 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 3.2052 2.0531 0. 0. 0. 0. 0. 0. 0. 10. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 0. 0. 4.5745 0.] total sum of alpha : 49.1489 </pre> |

| | |
|-----|---|
| p=5 | <pre> nonlinear_SVM_polynomial(C=10, p=5) : alpha : [0. 0. 6.7459 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 10. 0. 6.7459 0.4632 0. 0. 0. 0. 0. 0. 0. 10. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.7114 0. 0. 10. 0. 0. 0. 2.2436 0.] total sum of alpha : 47.91 </pre> |
|-----|---|

- bias and CR(%) :

Nonlinear SVM (polynomial)

| penalty weight C | p | bias | CR(%) |
|------------------|------|---------|---------|
| 10 | 1.00 | 15.1400 | 96.0000 |
| 10 | 2.00 | 13.7329 | 94.0000 |
| 10 | 3.00 | 6.4345 | 96.0000 |
| 10 | 4.00 | 4.6678 | 96.0000 |
| 10 | 5.00 | -3.6896 | 34.0000 |

4. 結果討論:

1. Linear SVM 與 kernel-based SVM 所訓練的 hyperplane 有何差異?

Ans: Linear SVM 是只在原特徵空間中去找 hyperplane，而 kernel-based SVM 是將特徵映射到更高維空間中，再去找最佳分離 hyperplane，在原特徵空間不好分離的兩類資料，映射到高維空間可能就能有較明顯的分離情況。

2. 隨著 kernel parameter 的改變，RBF kernel 與 polynomial kernel 所訓練的 hyperplane 可能有什麼變化？其與分類率的變化有何關聯？請嘗試解釋之。

Ans: 改變 kernel parameter，相當於將映射到不同空間中找分離 hyperplane，造成不同的 decision function 結果，使模型及分類率有所變化，具體來說，在 RBF kernel 情況下，sigma 越大，標準差越大，RBF kernel 更平緩而不集中，代表泛化能力較高，但分類能力可能就不是最好的；反之，sigma 越小，其標準差越小，RBF kernel 集中於局部，可能導致 model 過擬合，使其對 test data 無法有效分類，使 CR

降低，在實驗結果中確實有這樣的趨勢驗證此觀點。另一方面對於 polynomial kernel， p 越小表多項式次數較小，使 model 較簡單平滑，泛化性較高，但相對分類能力可能不是最好的；反之， p 較大表多項式次數較大，model 及決策邊界更複雜，能更貼近描述 training data 關係，但反而會使 model 過擬合，使在 test data 上分類率下降，像實驗結果中 $p=5$ 之分類率就掉到 34%，我還有試 $p=10$ ，分類率剩 28%，驗證了上述過擬合的情況。

3. 設定 kernel parameter 時，是否有方法避免 hyperplane 過度擬合 (overfitting) 的現象發生?若有請詳細討論。

Ans: 可使用格子搜尋法及 k-fold CV 去尋找有最好分類表現的參數，避免某一參數訓練之模型發現過擬合情形，且由題 2.之討論，也可大概知道若要避免過擬合發生，在 RBF kernel 情況下時，要避免 sigma 太小；polynomial kernel 情況下，則是要避免 p 過大。

4. 我和同學在實驗中發現，因助教有加上一段對 alpha 的處理及四捨五入至第 6 位，所以後續 model 計算我是以四捨五入後之 alpha 做的，這個四捨五入位數之影響很大，我試過取 4 位時，在 polynomial kernel $p=5$ 下 CR 結果就差了 20%，這是我發現的現象，對分類造成的影響似乎挺大的。