NYCU Machine Learning HW3

312605001

機器人碩士 歐庭維

Part 1 : Linear SVM

1. Initialization: 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.    ]
alpha_sum =  14.4
b =  [10.54, 10.54, 10.54, 10.54]
Mode: linear , Classification Rate (CR): 94.00%
```

2. Initialization: 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.  0. 10.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0. 10.  0.  0.  0. 10.  0.]
alpha_sum =  76.0
b =  [15.14, 15.14, 15.14]
Mode: linear , Classification Rate (CR): 96.00%
```

3. Initialization: C = 100

```
alpha =  [0.00000e+00 3.00000e-04 4.44438e+01 0.00000e+00 4.00000e-04 0.00000e+00
 1.00000e+02 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 1.00000e-04
 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 3.00000e-04 0.00000e+00
 3.00000e-04 0.00000e+00 1.00000e+02 0.00000e+00 4.44438e+01 0.00000e+00
 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 1.00000e-04 0.00000e+00
 0.00000e+00 1.00000e+02 1.00000e-04 1.00000e-04 0.00000e+00 0.00000e+00
 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 0.00000e+00 2.00000e-04
 0.00000e+00 0.00000e+00 1.00000e+02 0.00000e+00 0.00000e+00 0.00000e+00
 8.88884e+01 0.00000e+00]
alpha_sum =  577.7778
b =  [11.004, 11.0044, 11.0041, 11.0038, 11.004, 11.004, 11.0044, 11.005, 11.0056, 11.0052, 11.0049, 11.0044
]
Mode: linear , Classification Rate (CR): 92.00%
```

Part 2 : RBF kernel-based SVM

1. Initialization: C = 10, sigma = 5

```
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.     ]
alpha_sum =  193.771
b =  [0.0678, 0.0678, 0.0678, 0.0678, 0.0678]
Mode: rbf , Classification Rate (CR): 90.00%
```

2. Initialization: C = 10, sigma = 1

```
alpha =  [0.0000e+00 0.0000e+00 8.8908e+00 0.0000e+00 0.0000e+00 0.0000e+00
 1.0000e+01 3.4600e-01 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 1.0000e+01 0.0000e+00 8.8908e+00 0.0000e+00
 0.0000e+00 0.0000e+00 8.1275e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 1.0000e+01 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00 0.0000e+00
 0.0000e+00 0.0000e+00 1.0000e+01 0.0000e+00 1.0000e-04 0.0000e+00
 1.0000e+01 0.0000e+00]
alpha_sum =  76.2552
b =  [-0.2193, -0.2193, -0.2193, -0.2193, -0.2172]
Mode: rbf , Classification Rate (CR): 96.00%
```

3. Initialization: C = 10, sigma = 0.5

```
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.    ]
alpha_sum =  62.7532
b =  [-0.2444, -0.2444, -0.2444, -0.2444, -0.2444, -0.2444, -0.2444, -0.2444, -0.2444, -0.2444, -0.2385]
Mode: rbf , Classification Rate (CR): 94.00%
```

4. Initialization: C = 10, sigma = 0.1

```
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]
alpha_sum =  36.0576
b =  [-0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.
1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.
1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.
1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.1594, -0.
1594, -0.1594]
Mode: rbf , Classification Rate (CR): 88.00%
```

5. Initialization: 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]
alpha_sum =  42.7545
b =  [-0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.
1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.
1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.
1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.1106, -0.
1106, -0.1106, -0.1106]
Mode: rbf , Classification Rate (CR): 82.00%
```

Part 3 : Polynomial kernel-based SVM

1. Initialization: C = 10, p = 1

```
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.  0. 10.  0.  0.  0.  0.
  0.  0.  0.  0.  0.  0.  0.  0. 10.  0.  0.  0. 10.  0.]
alpha_sum =  76.0
b =  [15.14, 15.14, 15.14]
Mode: polynomial , Classification Rate (CR): 96.00%
```

2. Initialization: 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.      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.    ]
alpha_sum =  54.2594
b =  [13.7329, 13.7329, 13.7329, 13.7328]
Mode: polynomial , Classification Rate (CR): 94.00%
```

3. Initialization: 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.      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.    ]
alpha_sum =  53.0684
b =  [6.4345, 6.4346, 6.4345, 6.4346, 6.4344]
Mode: polynomial , Classification Rate (CR): 96.00%
```

4. Initialization: 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.      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.    ]
alpha_sum =  49.1489
b =  [4.6677, 4.6678, 4.6677, 4.668, 4.6676]
Mode: polynomial , Classification Rate (CR): 96.00%
```

5. Initialization: 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.    ]
alpha_sum =  47.91
b =  [-2.8302, -2.8302, -1.2454, -7.9984, -3.5436]
Mode: polynomial , Classification Rate (CR): 34.00%
```

Part 4 : Discussion and results presenting

1. Linear SVM 與 kernel-based SVM 所訓練的 hyperplane 有何差異?
    a. Linear kernel-based SVM 方法是指在特徵空間中找到最佳的超平面去將數據區分開，Linear SVM 的決策邊界在二維中（P = 1）時是一條直線，在三維中是平面（P = 2），在高維空間中是高維平面（P > 2）。

    b. Kernel-based SVM 方法是使用 kernel function 將原始的特徵空間 mapping 到高維度的空間，並在這個高維度空間中以一個線性超平面區分數據，與 Linear SVM 的超平面不同的是，kernel-based SVM 的超平面在原始的特徵空間中是非線性的，但是經過 mapping 到高維空間的時候會變成線性的，這樣使其能夠處理在原始特徵空間中非線性可分離的數據。

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。
    a. RBF kernel 的參數 gamma(紅字手寫部分)，主要功能是控制 RBF 函數的形狀，當 gamma 變大時，代表所訓練的超平面可能會變得更複雜，有可能更好的擬合數據的分布狀況。相同的，當 sigma 越大，代表 rbf kernel 更平緩（泛化能力強），而在實驗中，當 sigma = 1 時的 CR 最佳，有 96%，但隨著參數的調整，也有可能會由於考慮太多 noise 而產生 overfitting 的情況，使 CR 降低，可由 part2 結果得知。

② 幅射基底函數(Radial Basis Function kernel , RBF)

$$\rightarrow K(\mathbf{x}_i, \mathbf{x}_j) = \emptyset(\mathbf{x}_i) \cdot \emptyset(\mathbf{x}_j) = \emptyset(\mathbf{x}_i)^{\mathrm{T}} \emptyset(\mathbf{x}_j) = \emptyset(\mathbf{x}_j)^{\mathrm{T}} \emptyset(\mathbf{x}_i)$$

$$= e^{\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}$$

其中 σ : parameter of the RBF kernel ， $\sigma \in [-\infty, \infty]$

Note

b. Polynomial kernel 的參數代表多項式的次數，同樣的也控制多項式的複雜程度，當 degree 變大時，代表 kernel 以及超平面會變得更複雜，能夠更好的擬合數據，但同樣也有可能會有 overfitting 的問題。由 part 3 的結果可以得知，當 p = 1 ~ p = 4 時，CR 都大致能維持在 96%，但當 p = 5 時 CR 掉到了 34 % ，我推測就是由於 overfitting 的發生導致 CR 的下降。

$$\rightarrow K(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j\right)^P \text{ or } (\mathbf{x}_i^{\mathrm{T}} \mathbf{x}_j)^P \equiv \emptyset(\mathbf{x}_i) \cdot \emptyset(\mathbf{x}_j)$$

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。

通常在設定 kernel parameters 時，會透過大量調整不同的參數去找到最佳的 CR，但這通常代表要嘗試非常多次，當取的參數不夠多時，也無法得知當前看似最好的參數是不是 CR 表現的"global maximum"或是"local maximum"，或是 overfitting 發生的時機。為了找到最好的 CR，可以透過 Grid-search 以及交叉驗證來對幾乎所有的參數組合作迭代，這會花很多時間，但能確保能找到一組參數組合有最好的表現，當然也可以透過簡化模型或是增加數據量來避免 overfitting 的問題，但是這樣做的工作量會增加，以及能否真的能避免也會是個需要實驗的問題，這就看使用者的需求決定。