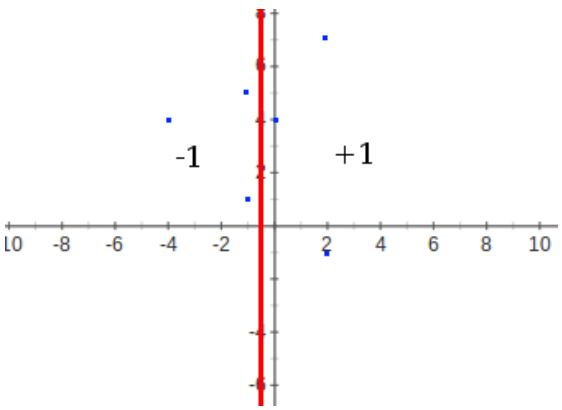
Machine Learning Homework #5

Qing-Cheng Li

R01922024

1.

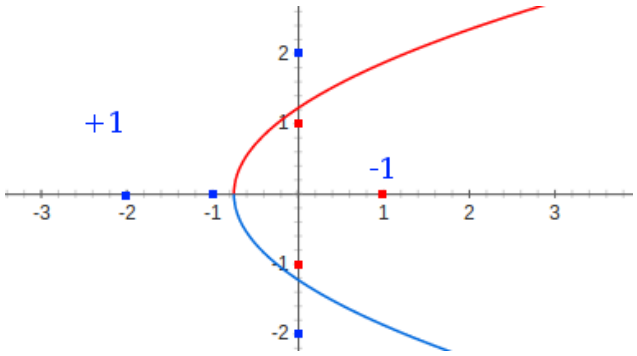


$ python p1.py

After running the program, $w = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$,$b = 1$. The optimal separating "hyperplane" in Z space is $z[1] = -0.5$. On-the-fat-boundary vectors are x_2 (-1,1), x_3 (-1,5) and x_4 (0,4).

2.

Let $(x[2])^2 - 2x[1] - 2 = z[1] = -0.5$, $x[2] = \pm\sqrt{\frac{3}{2} + 2x[1]}$ The on-the-fat-boundary vectors are the same.



3.

$ python p1.py

$$\alpha = \begin{bmatrix} 0 \\ 0.4667 \\ 0.4667 \\ 0.5334 \\ 0.2 \\ 0.2 \\ 0 \end{bmatrix}$$

4.

Using x_5 to calculate b, $b = 1 - (-0.4667 \times 16 + 0.5334 \times 4 + 0.2 \times 4 + 0.2 \times 36) = -1.6664$

Let $\sum_{n=1}^{N} \alpha_n y_n K(x_n, x) + b = 0$

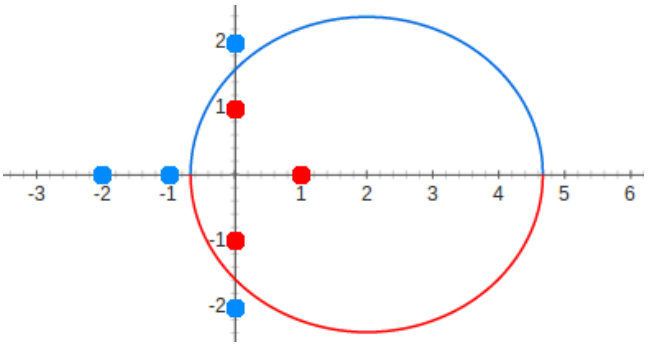
$$\sum_{n=1}^{N} \alpha_n y_n (2 + x_n[1]x[1] + x_n[2]x[2])^2 + b = 0$$

$$b - 0.4667(2 + x[2])^2 - 0.4667(2 - x[2])^2 + 0.5334(2 - x[1])^2 + 0.2(2 + 2x[2])^2 + 0.2(2 - 2x[2])^2 = 0$$

$$0.6666x[2]^2 + 0.5334x[1]^2 - 2.1336x[1] = -b$$

$$0.6666x[2]^2 + 0.5334(2 - x[1])^2 = -b + 4 \times 0.5334 = 3.8$$

$$\frac{(2-x[1])^2}{2.669102749^2} + \frac{x[2]^2}{2.38758666^2} = 1$$ (æ©¢åœ“)



5.

The curves and candidate support vectors are not the same. Because the kernel function represents some transform function, those functions are different, so the curves and candidate support vectors are different.

The number of candidate support vectors of Problem 3 is more than Problem 1, and the curve of Problem 3 “close” -1 sample in some area, but Problem dosen’t.

6.

If a vector is on the bundary, $\xi_n = 0$, if it in the boundary area, $\xi_n > 0$ (violation). In this problem, we want to minimize $\frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n^2$, if vector not in the margin area, the optimal result is set $\xi_n = 0$, not set $\xi_{n'} < 0$ because $\xi_n^2 = 0 < \xi_{n'}^2$, so the constraints $\xi_n \geq 0$ are not necessary.

7.

$$L((b, w, \xi), \alpha) = \frac{1}{2} w^T w + C \sum_{n=1}^{N} \xi_n^2 + \sum_{n=1}^{N} \alpha_n (1 - \xi_n - y_n (w^T x_n + b))$$

8.

$\frac{\partial L((b,w,\xi),\alpha)}{\partial b} = -\sum_{n=1}^{N} \alpha_n y_n = 0$, no loss of optimality if solving with constraint $\sum_{n=1}^{N} \alpha_n y_n = 0$.

$\frac{\partial L((b,w,\xi),\alpha)}{\partial w_i} = w_i - \sum_{n=1}^{N} \alpha_n y_n x_{n,i} = 0$, no loss of optimality if solving with constraint $w = \sum_{n=1}^{N} \alpha_n y_n x_n$. (Replace w with α)

$\frac{\partial L((b,w,\xi),\alpha)}{\partial \xi_n} = 2C\xi_n - \alpha_n = 0$, no loss of optimality if solving with constraint $\xi_n = \frac{\alpha_n}{2C}$. (Replace ξ with α)

$$\max_{All\alpha_n > 0,\ \sum_{n=1}^{N} \alpha_n y_n = 0,\ w = \sum_{n=1}^{N} \alpha_n y_n x_n,\ \xi_n = \frac{\alpha_n}{2C}} \quad -\frac{1}{2}\left\| \sum_{n=1}^{N} \alpha_n y_n x_n \right\|^2 + \sum_{n=1}^{N} \alpha_n - \frac{1}{4C}\sum_{n=1}^{N} \alpha_n^2$$

->

$$\min_{\alpha} \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m - \sum_{n=1}^{N} \alpha_n + \frac{1}{4C}\sum_{n=1}^{N} \alpha_n^2$$

Subject to: $\alpha_n \geq 0$, $\sum_{n=1}^{N} \alpha_n y_n = 0$

9.

Replace x_n with $z_n = \phi(x_n)$, $\frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m x_n^T x_m - \sum_{n=1}^{N} \alpha_n + \frac{1}{4C}\sum_{n=1}^{N} \alpha_n^2$

$$= \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \phi(x_n)^T \phi(x_m) - \sum_{n=1}^{N} \alpha_n + \frac{1}{4C}\sum_{n=1}^{N} \alpha_n^2$$

$$= \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m K(x_n, x_m) - \sum_{n=1}^{N} \alpha_n + \frac{1}{4C}\sum_{n=1}^{N} \alpha_n^2$$

The optimization problem:

$$\min_{\alpha} \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m K(x_n, x_m) - \sum_{n=1}^{N} \alpha_n + \frac{1}{4C}\sum_{n=1}^{N} \alpha_n^2$$

Subject to: $\alpha_n \geq 0$, $\sum_{n=1}^{N} \alpha_n y_n = 0$

10.

If $\phi_1(x)$ is a $N \times 1$ matrix, and $\phi_2(x)$ is a $M \times 1$ matrix.

Let $\phi(x)$ be $\begin{bmatrix} \phi_1(x)[1]\phi_2(x)[1] \\ \vdots \\ \phi_1(x)[1]\phi_2(x)[M] \\ \phi_1(x)[2]\phi_2(x)[1] \\ \vdots \\ \vdots \\ \phi_1(x)[N]\phi_2(x)[M] \end{bmatrix}_{NM\times 1}$

so that $K(x_n, x_m) = \phi(x_n)^T \phi(x_m) = \sum_{i=1}^{N} \sum_{j=1}^{M} \phi_1(x_n)[i]\phi_2(x_n)[j]\phi_1(x_m)[i]\phi_2(x_n)[j]$

$= (\phi_1(x_n)^T \phi_1(x_m))(\phi_2(x_n)^T \phi_2(x_m)) = K_1(x_n, x_m)K_2(x_n, x_m)$.

11.

Let $x^T x'$ be X, $K(x, x') = -2X^2 + 3X = X(3 - 2X)$, if $X > \frac{3}{2}$ for all $x_i^T x_j$ for $1 \le i, j \le N$. Then all $k_{i,j}$ in matrix K is negative, if v be $\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}_{N\times 1}$, $v^T K v < 0$, matrix K is not postive semi-definite, so $K(x, x')$ is not a valid kernel function.

12.

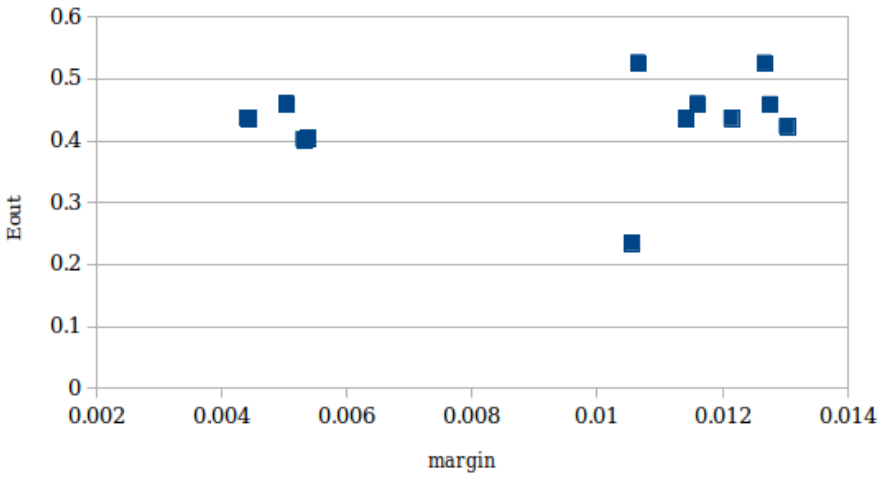
$K(x, x') = e^{k_1(x,x')} = e^{\phi_1(x)^T \phi_1(x')}$

$= \sum_{i=0}^{\infty} \frac{(\phi_1(x)^T \phi_1(x'))^i}{i!} = \sum_{i=0}^{\infty} (\sqrt{\frac{1}{i!}}\phi_1(x)^i)(\sqrt{\frac{1}{i!}}\phi_1(x')^i) = \phi(x)^T \phi(x')$

$\phi(x) = (1, \sqrt{\frac{1}{1!}}\phi_1(x), \sqrt{\frac{1}{2!}}\phi_1(x)^2, \ldots)$

13.

```
$ python p13.py
```



Larger margin, larger Eout. And it seems to have 2 clusters.

14.

Generate hw5_14_train.in for Libsvm input format.

```
$ python p14.py
```

Gamma = $\frac{1}{2\alpha^2}$

Using script p14.sh

$ sh p14.sh c gamma

c = 0.001, g=32 (α=0.125)

$$E_{in} = 41.6\% \ , E_{vc} = 41.6\% \ , \frac{\#SV}{N} = 1$$

c = 0.001, g=2 (α=0.5)

$$E_{in} = 41.6\% \ , E_{vc} = 41.6\% \ , \frac{\#SV}{N} = 0.84$$

c = 0.001, g=0.125 (α=2)

$$E_{in} = 41.6\% \ , E_{vc} = 41.6\% \ , \frac{\#SV}{N} = 0.844$$

c = 1, g=32 (α=0.125)

$$E_{in} = 0\% \ , E_{vc} = 41.6\% \ , \frac{\#SV}{N} = 1$$

c = 1, g=2 (α=0.5)

$$E_{in} = 0\% \ , E_{vc} = 41.6\% \ , \frac{\#SV}{N} = 1$$

c = 1, g=0.125 (α=2)

$$E_{in} = 12.6\% \ , E_{vc} = 37.2\% \ , \frac{\#SV}{N} = 0.95$$

c = 1000, g=32 (α=0.125)

$$E_{in} = 0\% \ , E_{vc} = 41.6\% \ , \frac{\#SV}{N} = 1$$

c = 1000, g=2 (α=0.5)

$$E_{in} = 0\% \ , E_{vc} = 41.6\% \ , \frac{\#SV}{N} = 1$$

c = 1000, g=0.125 (α=2)

$$E_{in} = 0\% \ , E_{vc} = 45.6\% \ , \frac{\#SV}{N} = 0.934$$

- Larger α , lesser $\frac{\#SV}{N}$
- When c >= 1, The Ein is 0, but (c=1,g=0.125)'s Ein is 12.6, and its Ecv is more less than others%
- Ecv seems always 41.6%

15.

Generate data

$ sh genP15CVdata.sh

Get result

$ sh p15.sh c gamma

c = 0.001, g=32 (α=0.125)

$E_{in} = 0\%$, $E_{vc} = 41.6\%$, $\frac{\#SV}{N} = 0.996$

c = 0.001, g=2 (α=0.5)

$E_{in} = 11.8\%$, $E_{vc} = 42.8\%$, $\frac{\#SV}{N} = 0.998$

c = 0.001, g=0.125 (α=2)

$E_{in} = 12.8\%$, $E_{vc} = 46\%$, $\frac{\#SV}{N} = 0.998$

c = 1, g=32 (α=0.125)

$E_{in} = 0\%$, $E_{vc} = 41.6\%$, $\frac{\#SV}{N} = 1$

c = 1, g=2 (α=0.5)

$E_{in} = 0\%$, $E_{vc} = 41.6\%$, $\frac{\#SV}{N} = 1$

c = 1, g=0.125 (α=2)

$E_{in} = 1.8\%$, $E_{vc} = 44.6\%$, $\frac{\#SV}{N} = 0.946$

c = 1000, g=32 (α=0.125)

$E_{in} = 0\%$, $E_{vc} = 41.6\%$, $\frac{\#SV}{N} = 1$

c = 1000, g=2 (α=0.5)

$E_{in} = 0\%$, $E_{vc} = 41.6\%$, $\frac{\#SV}{N} = 1$

c = 1000, g=0.125 (α=2)

$E_{in} = 0\%$, $E_{vc} = 46.6\%$, $\frac{\#SV}{N} = 1$

- The number of suppert vectors seems very large, almost the whole training data set.
- Larger α , larger Ecv (sometime larger Ein)
- Larger c, lesser Ein

16.

```
$ python lssvm.py sigma lambda inputfile
```

For $\sigma = 0.125, 0.5, 2$, $\lambda = 0.001, 1, 1000$, I got $E_{in} = 0.326$,$E_{cv} = 0.334$ and $\frac{\#SV}{N} = 1$, $(0.5, 1000)$'s $E_{in} = 0.32$,$E_{cv} = 0.338$, $(2, 1000)$'s $E_{in} = 0.322$,$E_{cv} = 0338$.

I found the performance are better when using larger λ and σ .