B06705002
資管三
黃啟宏

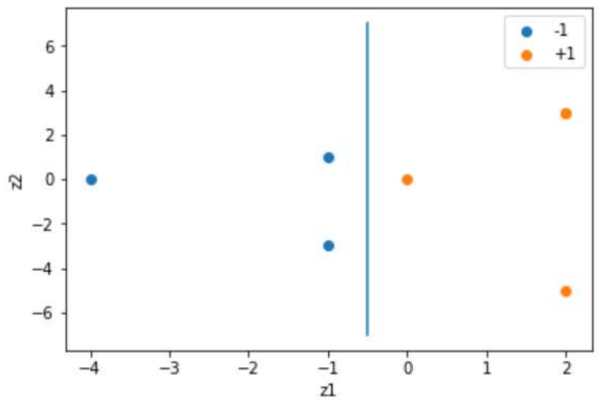
1.

In [28]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
```

In [29]:
```
x = np.array([[1, 0], [0, 1], [0, -1], [-1, 0], [0, 2], [0, -2], [-2, 0]])
y = [-1, -1, -1, 1, 1, 1, 1]
```

In [30]:
```
z1 = (x[:, 1] ** 2) - 2 * x[:, 0] - 2
z2 = x[:, 0] ** 2 -2 * x[:, 1] - 1
z = [[z1[i], z2[i]] for i in range(len(x))]
plt.scatter(z1[:3], z2[:3], label="-1")
plt.scatter(z1[3:], z2[3:], label="+1")
plt.xlabel("z1")
plt.ylabel("z2")
plt.legend()
plt.plot([-0.5, -0.5], [-7, 7])
```

Out[30]: [<matplotlib.lines.Line2D at 0x231df9e05f8>]



The equation of the optimal separating "hyperplane" in the Z space is Z1 = -0.5. As above.

2.

2.

In [4]:
```
clf = svm.SVC(C=1e100, kernel='poly', degree=2, gamma=1, coef0=1, shrinking = False)
clf.fit(x, y)

alpha = clf.dual_coef_
sv = clf.support_vectors_
print("the optimal α is:", alpha)
```

```
the optimal α is: [[-0.64491963 -0.76220325  0.88870349  0.22988879  0.2885306 ]]
```

In [5]:
```
print("Based on those α, the support vectors are: ", sv)
```

```
Based on those α, the support vectors are:  [[ 0.  1.]
 [ 0. -1.]
 [-1.  0.]
 [ 0.  2.]
 [ 0. -2.]]
```

3.

3.

```
In [6]: b = clf.intercept_
        print("gsvm = ")
        for i in range(0, len(sv)):
            print(alpha[0][i], "*(1+", sv[i][0], "*x1+", sv[i][1], "*x2)^2")
        print("+", b)
```

```
gsvm =
-0.6449196277436483 *(1+ 0.0 *x1+ 1.0 *x2)^2
-0.76220324878158 *(1+ 0.0 *x1+ -1.0 *x2)^2
0.8887034937554661 *(1+ -1.0 *x1+ 0.0 *x2)^2
0.22988878612539818 *(1+ 0.0 *x1+ 2.0 *x2)^2
0.288530596644364 *(1+ 0.0 *x1+ -2.0 *x2)^2
+ [-1.66633141]
```

K(xn,x)with（sv1, sv2）, K=(1+(sv1, sv2)T(x1,x2))^2 = (1+x2)^2 We can know that nonlinear curve in the X space is: gsvm = -0.6449196277436483 * (1 + x2)^2 -0.76220324878158 * (1 - x2)^2 + 0.8887034016088986 * (1 -x1)^2 + 0.22988878612539818 *(1 + 2x2)^2 + 0.288530596644364*(1 -2x2)^2 -1.66633141

4.

No, they should not be the same. For Question 1, we project the data into a 2-d space(Z). However, we project the data into a 6-d space for Question 3.

5.

$(P_1')\ \min_{w,b,\varepsilon} \frac{1}{2}w^Tw + C\sum_{n=1}^{N}\varepsilon_n$

s.t. $y_n(w^Tx_n+b) \ge \rho_n - \varepsilon_n$

$\varepsilon_n \ge 0$, for $n = 1, 2, \ldots, N$.

$\mathcal{L}((b,w,\varepsilon),(\alpha,\beta)) = \frac{1}{2}w^Tw + C\sum_{n=1}^{N}\varepsilon_n + \sum_{n=1}^{N}\alpha_n[\rho_n - \varepsilon_n - y_n(w^Tx_n+b)] + \sum_{n=1}^{N}\beta_n(-\varepsilon_n)$

6.

6. using strong duality,

$$\max_{\alpha_n\ge 0,\beta_n\ge 0}\left(\min_{(b,w,\varepsilon)} \mathcal{L}((b,w,\varepsilon),(\alpha,\beta))\right)$$

by KTT

$\frac{\partial \mathcal{L}}{\partial b} = 0 = \sum_{n=1}^{N}\alpha_n(-y_n) = -\sum_{n=1}^{N}\alpha_n y_n$ ①

$\frac{\partial \mathcal{L}}{\partial w} = w - \sum_{n=1}^{N}\alpha_n y_n x_n = 0$

$w = \sum_{n=1}^{N}\alpha_n y_n x_n$ ②

$\frac{\partial \mathcal{L}}{\partial \varepsilon} = C - \alpha_n - \beta_n = 0$

$\beta_n = C - \alpha_n$ ③

$$\mathcal{L} = \frac{1}{2}w^Tw + C\sum_{n=1}^{N}\varepsilon_n + \sum_{n=1}^{N}\alpha_n[\rho_n - \varepsilon_n - y_n(w^Tx_n+b)] + \sum_{n=1}^{N}(C-\alpha_n)(-\varepsilon_n)$$

$$= \frac{1}{2}w^Tw + \sum_{n=1}^{N}\alpha_n\varepsilon_n + \sum_{n=1}^{N}\alpha_n(\rho_n - \varepsilon_n) - \underbrace{\sum_{n=1}^{N}\alpha_n y_n w^T x_n}_{w^Tw}$$

$$= -\frac{1}{2}w^Tw + \sum_{n=1}^{N}\alpha_n\rho_n$$

Thus, (P_1') turns to

$$\max_{\alpha_n\ge 0,\beta_n\ge 0} -\frac{1}{2}w^Tw + \sum_{n=1}^{N}\alpha_n\rho_n \iff \min \frac{1}{2}w^Tw - \sum_{n=1}^{N}\alpha_n\rho_n$$

s.t. $\sum_{n=1}^{N}\alpha_n y_n = 0$ $\quad \beta_n = C - \alpha_n$

$w = \sum_{n=1}^{N}\alpha_n y_n x_n$ $\quad$ For $n = 1, 2, \ldots, N$

7.

7.

with $p_n = 0.5$, we can know that difference is from $\min \frac{1}{2} w^T w - \sum_{n=1}^{N} 0.5 \alpha_n$

$$w'_* = \sum_{n=1}^{N} \alpha_n' y_n x_n$$

$$w'_* = 2 w_*$$

$$b'_* = b_*$$

8.

For standard hard-margin SVM dual: α∗n >= 0 for n = 1,2,...N For standard soft-margin SVM dual: C >= α∗n >= 0 for n = 1,2,...N SO, if a hard-margin SVM has an optimal solution of some vector α∗, if f C ≥ max1≤n≤N α∗n, it also satisfies the condition of standard soft-margin machine(C > α∗n >= 0 for n = 1,2,...N), which means that the vector α∗ is also an optimal solution to the soft-margin SVM .

9.

9.

using matrix $\begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$

for [a]

$$(1-k_1(x,x'))^1 = \begin{bmatrix} 1-0.9 & 1-0.1 \\ 1-0.1 & 1-0.9 \end{bmatrix} = \begin{bmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{bmatrix}$$

$\lambda_1 = \frac{-4}{5}$

$\lambda_2 = 1$ (not valid) ✗

for [b]

$$(1-k_1(x,x'))^0 = \begin{bmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{bmatrix}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$\lambda_1 = 1$ (valid) ✓

for [c] $(1-k_1(x,x'))^{-1} = \begin{bmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{bmatrix}^{-1} = \begin{bmatrix} -\frac{1}{8} & \frac{9}{8} \\ \frac{9}{8} & -\frac{1}{8} \end{bmatrix}$

$\lambda_1 = -\frac{5}{4}$

$\lambda_2 = 1$ (not valid) ✗

for [d] $(1-k_1(x,x'))^{-2} = \begin{bmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{bmatrix}^{-2} = \begin{bmatrix} \frac{41}{32} & \frac{-9}{32} \\ \frac{-9}{32} & \frac{41}{32} \end{bmatrix}$

$\lambda_1 = 1$

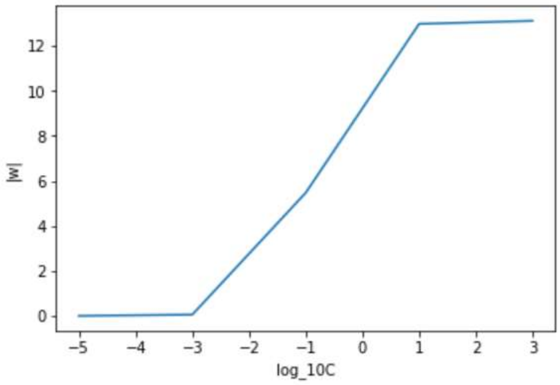
$\lambda_2 = \frac{25}{16}$ ✓

10.

For the dual of soft-margin support vector machine, C >= α∗n >= 0, with using K˜ along with a new C˜ = C/p instead of K with the original C, C˜ >= α∗n >= 0 would lead to C >= α∗n*p >= 0. For solving unique b with free SV, b = ys - Σ(α∗n)(y∗n)K˜(x∗n, x∗s) = ys - Σ(α∗np)(y∗n)K(x∗n, x∗s)* So, gSVM(x) = sign(Σ(α∗n)(y∗n)K˜(x∗n, x∗s) + b) = sign(Σ(α∗n*p)(y∗n)K(x∗n, x∗s) + b)

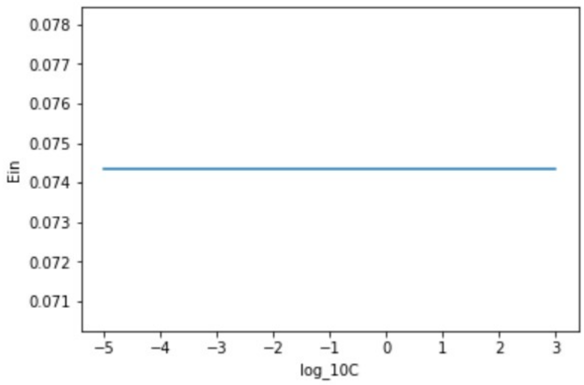
11.

We can know that from the plot '|w| versus log10 C' below, we can get a bigger |w| while C gets bigger.



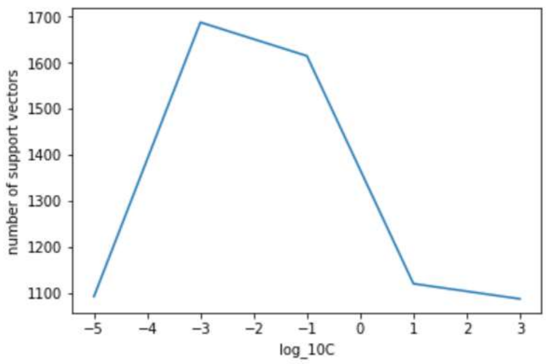
12.

We can know that from the plot 'Ein versus log10 C' below, Ein won't change while C gets bigger or smaller.



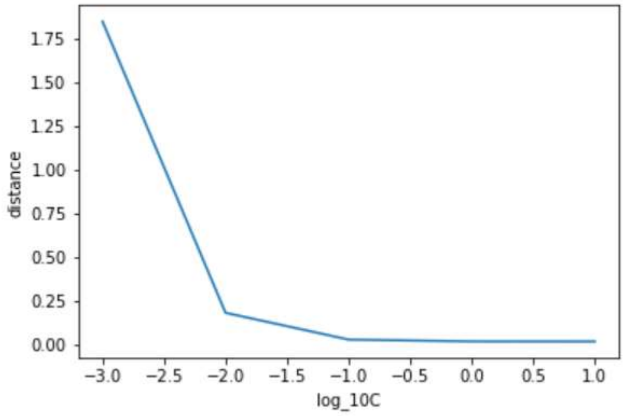
13.

We can know that from the plot 'number of the support vectors versus log10 C' below, the number is the most while C is -3.



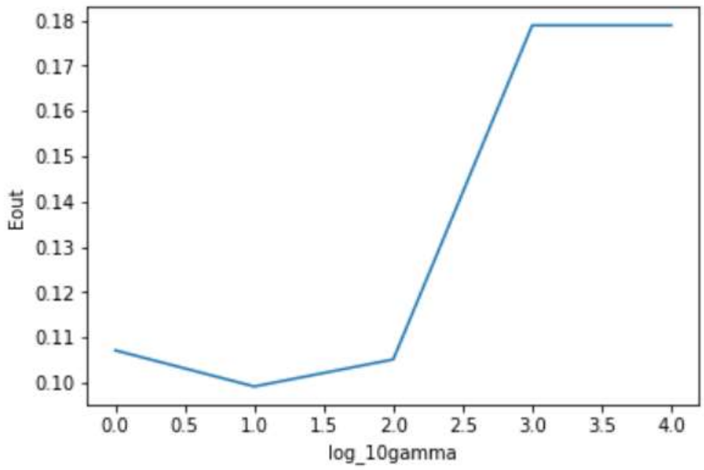
14.

We can know that from the plot 'distance versus log10 C' below, distance gets more closer to zero while C gets bigger. Which means the SVM is harder.

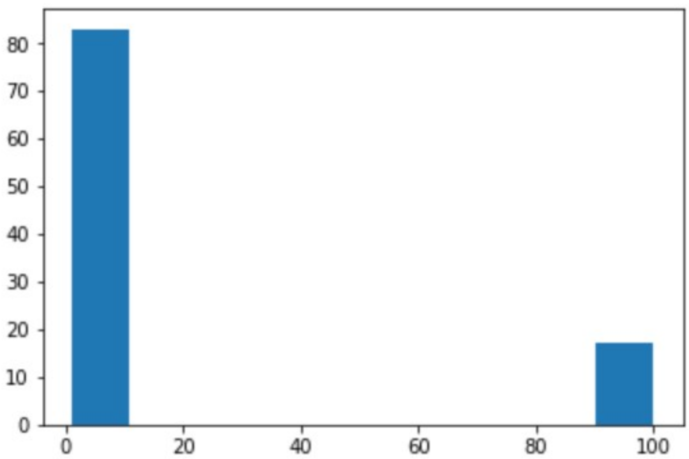


15.

We can know that from the plot 'Eout versus log10 gamma' below, Eout gets bigger(worse) while gamma gets bigger.



16.



17.

17.

We know that w_i is $\sum_{i=1}^{N} \alpha_i y_i z_i$.

From KKT, we know $\sum_{i=1}^{N} \alpha_i y_i = 0$, with the constant z, we can know $w_i = 0$ while $z \cdot \sum_{i=1}^{N} \alpha_i y_i = 0$ while w_i corresponds to constant z.

18.

18.

$$\min_{\alpha \in \mathbb{R}^N} \max_{\lambda_i \ge 0} : \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m x_n^T x_m - \sum_{n=1}^{N} \alpha_n + \lambda_0 \left(\sum_{n=1}^{N} y_n \alpha_n\right) - \sum_{n=1}^{N} \lambda_n \alpha_n$$

$$\Downarrow$$

$f(\lambda, \alpha)$

$$\max_{\lambda_i \ge 0} \min_{\alpha \in \mathbb{R}^N} : \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m x_n^T x_m - \sum_{n=1}^{N} \alpha_n + \lambda_0 \left(\sum_{n=1}^{N} y_n \alpha_n\right) - \sum_{n=1}^{N} \lambda_n \alpha_n$$

對 α 偏微

$$\frac{1}{2} \sum_{n=1}^{N} y_n y_i \alpha_n x_n^T x_i + \frac{1}{2} \sum_{m=1}^{N} y_i y_m \alpha_m x_i^T x_m - 1 + \lambda_0 y_i - \lambda_i$$

$$= \sum_{n=1}^{N} y_n y_i \alpha_n x_i^T x_n - 1 + \lambda_0 y_i - \lambda_i$$

$$\Downarrow$$

$$\underbrace{\sum_{n=1}^{N} y_n y_i \alpha_n x_i^T x_n}_{Q_D \alpha} = 1 - \lambda_0 y_i + \lambda_i$$

$$\Downarrow$$

$$Q_D \alpha = 1 - \lambda_0 y + \lambda$$

算式 $\sum_{m=1}^{N} \sum_{n=1}^{N} y_n y_m \alpha_n \alpha_m x_n^T x_m = \alpha^T Q_D \alpha$

$$\sum_{n=1}^{N} \alpha_n = 1_N^T \alpha$$

$$\sum_{n=1}^{N} \lambda_n \alpha_n = \alpha^T \lambda$$

$$f(\lambda, \alpha) = \frac{1}{2} \alpha^T Q_D \alpha - 1_N^T \alpha + \lambda_0 \alpha^T y - \alpha^T \lambda$$

$$= -\frac{1}{2} \underbrace{\alpha^T}_{= Q_D^{-1}(1_N - \lambda_0 y + \lambda)} (1_N - \lambda_0 y + \lambda)$$

$$\min_{\lambda_i \ge 0} : -\frac{1}{2} (1_N - \lambda_0 y + \lambda)^T Q_D^{-1} (1_N - \lambda_0 y + \lambda)$$

subject to : $\lambda_i \ge 0$ where $n = 0 \ldots N$