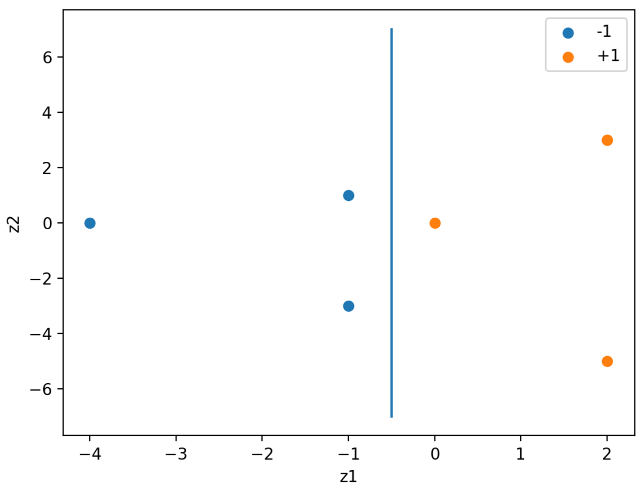
Machine Learning Techniques, Spring 2020, HW1

B07902064 資工二 蔡銘軒

July 22, 2020

Problem 1

After transforming every point from $\mathcal{X}$ space to $\mathcal{Z}$ space, the boundary between the two categories, $+1$ and -1, is clear. It is easy to identify that the optimal separating "hyperplane" is $z_1 = -0.5$, as shown in the figure below.



Problem 2

```python
from sklearn import svm
x = [[1, 0], [0, 1], [0, -1], [-1, 0], [0, 2], [0, -2], [-2, 0]]
y = [-1, -1, -1, 1, 1, 1, 1]
clf = svm.SVC(C = 1000000, kernel = 'poly', coef0 = 1, degree = 1, gamma = 1)
clf.fit(x, y)
print(clf.support_vectors_)
print(clf.dual_coef_)
```

Using the snippet shown above, we have:

$$\text{optimal } \alpha = [0, 0.64491963, 0.76220325, 0.88870349, 0.22988879, 0.2885306, 0]$$

The corresponding support vectors are:

$$(0,1), (0,-1), (-1,0), (0,2), (0,-2)$$

Note that the *dual_coef_* from the result is in fact $\alpha_s \cdot y_s$ for $s \in \{\text{indices of SVs}\}$

Problem 3

We have

$$b = y_s - \sum_{n=1}^{N} \alpha_n y_n K(x_s, x_n)$$

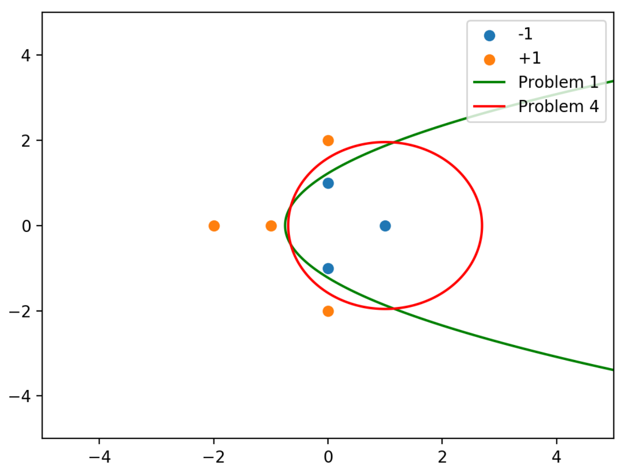
for $s \in \{\text{indices of SVs}\}$, and $N = 7$ in this case. We plug in the values to get $b \approx -1.66633141$.
The curve is given by

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

Plugging in the values gives us

$$\begin{aligned}&-0.64491963 \cdot (1 + x_2)^2 - 0.76220325 \cdot (1 - x_2)^2 + 0.88870349 \cdot (1 - x_1)^2 + \\ &0.22988879 \cdot (1 + 2x_2)^2 + 0.2885306 \cdot (1 - 2x_2)^2 - 1.66633141 = 0\end{aligned}$$

Problem 4



The curve in **Problem 1** is

$$\begin{aligned}z_1 = x_2^2 - 2x_1 - 2 &= -0.5 \\ \implies x_2^2 - 2x_1 - 1.5 &= 0\end{aligned}$$

which is not the same as the one in **Problem 4**. The reason is that the two transformations are different. In **Problem 1**, we transform (x_1, x_2) into another two dimensional space (z_1, z_2). In **Problem 4**, we transform (x_1, x_2) into a six dimensional space, $(1, \sqrt{2}x_1, \sqrt{2}x_2, x_1 x_2, x_1^2, x_2^2)$

Problem 5

We introduce the Lagrange multipliers α_n, β_n for each constraint. And we have

$$\mathcal{L}((b, \mathbf{w}, \xi), (\alpha, \beta)) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + C \cdot \sum_{n=1}^{N} \xi_n + \sum_{n=1}^{N} \alpha_n \cdot (\rho_n - \xi_n - y_n(\mathbf{w}^T\mathbf{x}_n + b)) + \sum_{n=1}^{N} \beta_n \cdot (-\xi_n)$$

We can see that along with the max function, any violated condition will lead to an invalid solution, so the constraints of the original formulation still hold.

Problem 6

To find the minimum, we set the partial derivatives to 0. We have

$$\frac{\partial \mathcal{L}((b, \mathbf{w}, \xi), (\alpha, \beta))}{\partial \xi_n} = C - \alpha_n - \beta_n = 0 \tag{1}$$

$$\frac{\partial \mathcal{L}((b, \mathbf{w}, \xi), (\alpha, \beta))}{\partial b} = -\sum_{n=1}^{N} \alpha_n y_n = 0 \tag{2}$$

$$\frac{\partial \mathcal{L}((b, \mathbf{w}, \xi), (\alpha, \beta))}{\partial \mathbf{w}} = \mathbf{w} - \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n = 0 \tag{3}$$

(1) gives us $\beta_n = C - \alpha_n$. And since $\beta_n \geq 0$, we have $0 \leq \alpha_n \leq C$. Now that β_n can be expressed by C and α_n, it is not involved in the optimization problem anymore. Substituting $C - \alpha_n$ for β_n cancels ξ_n, and we have

$$\max_{0 \leq \alpha_n \leq C} \left(\min_{b, \mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w} + \sum_{n=1}^{N} \alpha_n (\rho_n - y_n(\mathbf{w}^T \mathbf{x}_n + b)) \right)$$

Using (2), we can further simplify the problem to

$$\max_{0 \leq \alpha_n \leq C, \sum_{n=1}^{N} \alpha_n y_n = 0} \left(\min_{\mathbf{w}} \frac{1}{2} \mathbf{w}^T \mathbf{w} + \sum_{n=1}^{N} \alpha_n (\rho_n - y_n(\mathbf{w}^T \mathbf{x}_n)) \right)$$

(3) gives us $\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$. After substitution, the problem is now

$$\max_{0 \leq \alpha_n \leq C, \sum_{n=1}^{N} \alpha_n y_n = 0} \left(-\frac{1}{2} || \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n ||^2 + \sum_{n=1}^{N} \alpha_n \rho_n \right)$$

which is equivalent to

$$\min_{0 \leq \alpha_n \leq C, \sum_{n=1}^{N} \alpha_n y_n = 0} \left(\frac{1}{2} || \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n ||^2 - \sum_{n=1}^{N} \alpha_n \rho_n \right)$$

with $\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x}_n$ and $\beta_n = C - \alpha_n$

Problem 7

Let $(\mathbf{w}'_*, b'_*, \xi'_*)$ be the optimal answer to (P'_1) with

$$\begin{aligned} & y_n\left(\mathbf{w}'_* \mathbf{x}_n + b'_*\right) \geq 0.5 - \xi'_{*n} \\ & \xi'_{*n} \geq 0 \end{aligned}$$

Multiply the inequalities by 2, and let $\mathbf{w}_* = 2\mathbf{w}'_*, b_* = 2b'_*, \xi_* = 2\xi'_*$, we have

$$\begin{aligned} & y_n\left(\mathbf{w}_* \mathbf{x}_n + b_*\right) \geq 1 - \xi_{*n} \\ & \xi_{*n} \geq 0 \end{aligned}$$

Since $(\mathbf{w}'_*, b'_*, \xi'_*)$ is optimal for (P'_1), we have

$$\min_{\mathbf{w}, b, \xi} \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{n=1}^{N} \xi_n = \frac{1}{2} \mathbf{w}'^T_* \mathbf{w}'_* + C \sum_{n=1}^{N} \xi'_{*n}$$

Multiply both sides by 4, we have

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}(2\mathbf{w}^T)(2\mathbf{w}) + 2C\sum_{n=1}^{N}(2\xi_n) = \frac{1}{2}(2\mathbf{w}_*'^T)(2\mathbf{w}_*') + 2C\sum_{n=1}^{N}(2\xi_{*n}')$$
$$= \frac{1}{2}\mathbf{w}_*^T\mathbf{w}_* + 2C\sum_{n=1}^{N}\xi_{*n}$$

If we represent the left hand side by

$$\min_{\hat{\mathbf{w}},\hat{b},\hat{\xi}} \frac{1}{2}\hat{\mathbf{w}}^T\hat{\mathbf{w}} + \hat{C}\sum_{n=1}^{N}\hat{\xi}_n$$

where $\hat{C} = 2C$, then it is the optimization problem in (P_1) and $(\mathbf{w}_*, b_*, \xi_*) = (2\mathbf{w}_*', 2b_*', 2\xi_*')$ is the optimal answer which also satisfies the contraints in (P_1). So the optimal $\mathbf{w}$ and b for (P_1) is $2\mathbf{w}_*'$ and $2b_*'$, respectively.

Problem 8

In class, we have seen that the constraint on α in the soft margin SVM is

$$0 \le \alpha_n \le C \ for \ n = 1, 2, \ldots, N$$

while in the hard margin SVM, the constraint on α is

$$0 \le \alpha_n \ for \ n = 1, 2, \ldots, N$$

It is clear that if $C \ge \max_{1\le n\le N} \alpha_n^*$, then α^* satisfies the constraints in the soft margin SVM. And since the soft margin SVM and the hard margin SVM have the same optimization goal, α^* is also optimal for the soft margin SVM.

Problem 9

Let K' a $n \times n$ matrix such that $K'_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. To show K is a valid kernel function, we have to show that K' is symmetric and positive semidefinite. It suffices to show that $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}')$, which guarantees the symmetry and postive semidefiniteness of K'.

Lemma 1. *Kernel functions are closed under addition.*

Proof. Let $K'_a, K'_b \in \mathbb{R}^{n\times n}$ be two symmetric and positive semidefinite matrices corresponding to kernel functions K_a, K_b, respectively, in the manner described previously. For any $c \in \mathbb{R}^{n\times 1}$, we have $c^T K_a c \ge 0$ and $c^T K_b c \ge 0$, which leads to $c^T(K_a + K_b)c \ge 0$. And it is obvious that symmetry is closed under addition. □

Lemma 2. *Kernel functions are closed under product.*

Proof. Let K_a, K_b be two kernel functions such that $K_a(\mathbf{x}, \mathbf{x}') = \phi^a(\mathbf{x})^T\phi^a(\mathbf{x}')$ and $K^b(\mathbf{x}, \mathbf{x}') = \phi^b(\mathbf{x})^T\phi^b(\mathbf{x}')$, where $\phi^a(\mathbf{x}) \in \mathbb{R}^M$ and $\phi^b(\mathbf{x}) \in \mathbb{R}^N$, $N, M \in \mathbb{N}$. Denote $\phi_i(\mathbf{x})$ as the *i-th* element in $\phi(\mathbf{x})$, we have

$$\begin{aligned}
K_c(\mathbf{x}, \mathbf{x}') &= K_a(\mathbf{x}, \mathbf{x}') \cdot K_b(\mathbf{x}, \mathbf{x}') \\
&= \sum_{m=1}^{M}(\phi_m^a(\mathbf{x})\phi_m^a(\mathbf{x}')) \cdot \sum_{n=1}^{N}\left(\phi_n^b(\mathbf{x})\phi_n^b(\mathbf{x}')\right) \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N}\left(\phi_m^a(\mathbf{x})\phi_n^b(\mathbf{x})\right)\left(\phi_m^a(\mathbf{x}')\phi_n^b(\mathbf{x}')\right) \\
&= \sum_{m=1}^{M}\sum_{n=1}^{N}\phi_{mn}^c(\mathbf{x})\phi_{mn}^c(\mathbf{x}') \\
&= \phi^c(\mathbf{x})^T\phi^c(\mathbf{x}')
\end{aligned}$$

where $\phi^c(\mathbf{x}) \in \mathbb{R}^{MN}$ such that $\phi^c_{mn}(\mathbf{x}) = \phi^a_m(\mathbf{x})\phi^b_n(\mathbf{x})$. □

[a] Counter example: Consider $K'_1 = \begin{bmatrix} 0.5 & 0.1 \\ 0.1 & 0.5 \end{bmatrix}$, which corresponds to a valid kernel. However, $K' = \begin{bmatrix} 0.5 & 0.9 \\ 0.9 & 0.5 \end{bmatrix}$, which has eigenvalues 1.4 and -0.4, is not positive semidefinite.

[b] In this case, K' is a $n \times n$ matrix filled with ones. It is symmetric and has eigenvalues n and 0, which makes $K(\mathbf{x}, \mathbf{x}')$ a valid kernel function.

[c] We denote $K_1(\mathbf{x}, \mathbf{x}')$ as k for brevity.
Using the lemmas derived previously, $K_N(\mathbf{x}, \mathbf{x}') = 1 + k + k^2 + \cdots + k^N$ is a valid kernel function.
We have $K(\mathbf{x}, \mathbf{x}') = \lim_{N\to\infty} 1 + k + k^2 + \cdots + k^N = \frac{1-k^{N+1}}{1-k}$. Since $0 < k < 1$, we have $\lim_{N\to\infty} k^{N+1} = 0$, so $K(\mathbf{x}, \mathbf{x}') = (1 - K_1(\mathbf{x}, \mathbf{x}'))^{-1}$ is a valid kernel.

[d] Using the result in **[c]** and **Lemma 2**, $(1 - K_1(\mathbf{x}, \mathbf{x}'))^{-1} \cdot (1 - K_1(\mathbf{x}, \mathbf{x}'))^{-1} = (1 - K_1(\mathbf{x}, \mathbf{x}'))^{-2}$ is a valid kernel.

The answers are **[b], [c], [d]**

Problem 10

The original optimization problem can be expressed as

$$\min_{\alpha} \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m K(\mathbf{x}, \mathbf{x}') - \sum_{n=1}^{N} \alpha_n$$

subject to

$$\begin{aligned} &\sum_{n=1}^{N} y_n \alpha_n = 0 \\ &0 \le \alpha_n \le C, \text{ for } n = 1, 2, \ldots, N \end{aligned}$$

Now let $\hat{K}(\mathbf{x}, \mathbf{x}') = pK(\mathbf{x}, \mathbf{x}')$, and $\hat{C} = \frac{C}{p}$ for some $p > 0$, the new optimization problem is

$$\min_{\hat{\alpha}} \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \hat{\alpha}_n \hat{\alpha}_m y_n y_m \hat{K}(\mathbf{x}, \mathbf{x}') - \sum_{n=1}^{N} \hat{\alpha}_n$$

subject to

$$\begin{aligned} &\sum_{n=1}^{N} y_n \hat{\alpha}_n = 0 \\ &0 \le \hat{\alpha}_n \le \hat{C}, \text{ for } n = 1, 2, \ldots, N \end{aligned}$$

It is clear that if α^* is optimal for the original problem, $\frac{1}{p}\alpha^*$ satisfies the conditions of the new optimization problem and is also optimal.

Proof. Suppose $\hat{\alpha}^*$ is optimal for the new optimization problem and $\hat{\alpha}^* \neq \frac{1}{p}\alpha^*$, then $p\hat{\alpha^*}$ is valid and optimal for the orignal problem, contradiction. □

For brevity, everything without a hat ($\hat{.}$) refers to the original problem, otherwise it refers to the new problem. We have

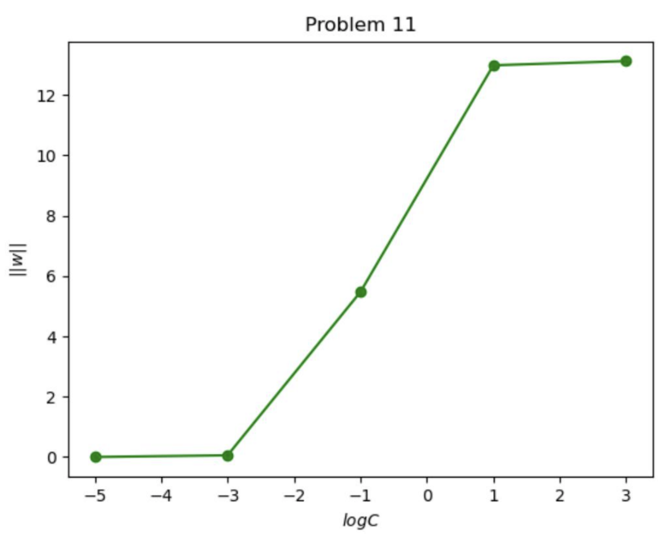
$$b = y_s - \sum_{n=1}^{N} \alpha_n^* y_n K(\mathbf{x}_n, \mathbf{x}_s) \text{ for } s \in \{\text{indices of SVs}\}$$
$$g_{\text{SVM}}(x) = sign\left(\left(\sum_{n=1}^{N} \alpha_n^* y_n K(\mathbf{x}_n, \mathbf{x})\right) + b\right)$$

and

$$\begin{aligned}
\hat{b} &= y_s - \sum_{n=1}^{N} \hat{\alpha}_n^* y_n \hat{K}(\mathbf{x}_n, \mathbf{x}_s) \text{ for } s \in \{\text{indices of SVs}\} \\
&= y_s - \sum_{n=1}^{N} \frac{1}{p} \alpha_n^* y_n p K(\mathbf{x}_n, \mathbf{x}_s) \text{ for } s \in \{\text{indices of SVs}\} \\
&= b \\
\hat{g}_{\text{SVM}}(x) &= sign\left(\left(\sum_{n=1}^{N} \hat{\alpha}_n^* y_n \hat{K}(\mathbf{x}_n, \mathbf{x})\right) + \hat{b}\right) \\
&= sign\left(\left(\sum_{n=1}^{N} \frac{1}{p} \alpha_n^* y_n p K(\mathbf{x}_n, \mathbf{x})\right) + b\right) \\
&= g_{\text{SVM}}(x)
\end{aligned}$$

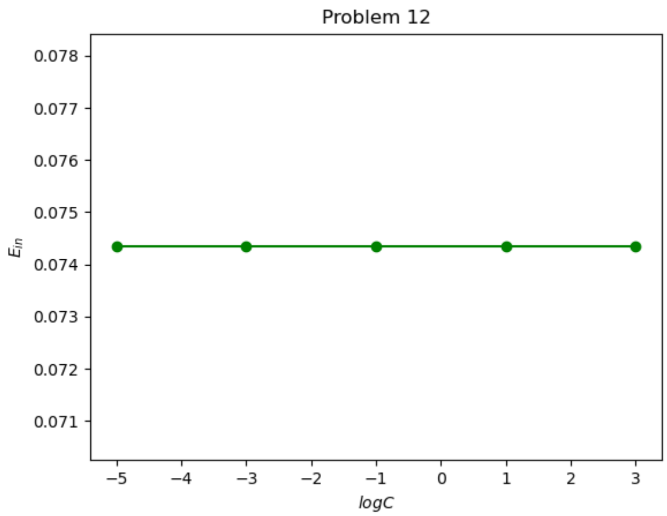
This shows the two SVMs are equivalent.

Problem 11



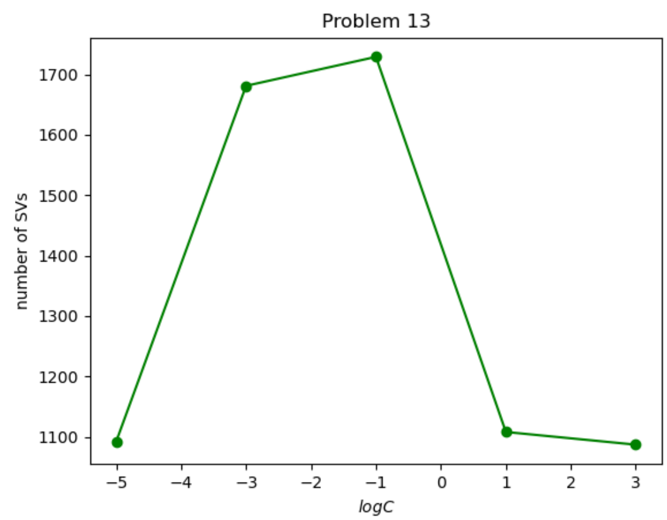
The bigger the C(punishment), the less it can tolerate violation. So the margin $\frac{1}{|\mathbf{w}|}$ gets smaller($\mathbf{w}$ gets bigger) as C grows larger.

Problem 12



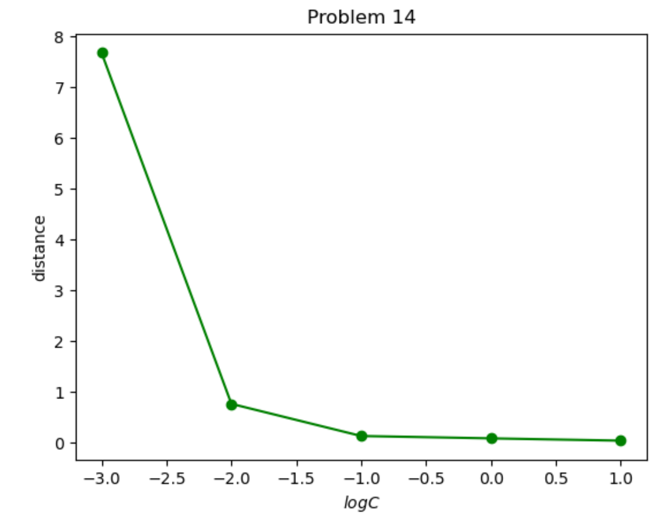
The E_{in} stays the same as C becomes larger. It turned out that the number 8 is too scarce, and the SVM found a boundary that treated every data as "Not 8", regardless of C. If we print the distribution, we could see that 8 is very scarce and is mixed with "Not 8". So the behavior of the SVM is reasonable.

Problem 13



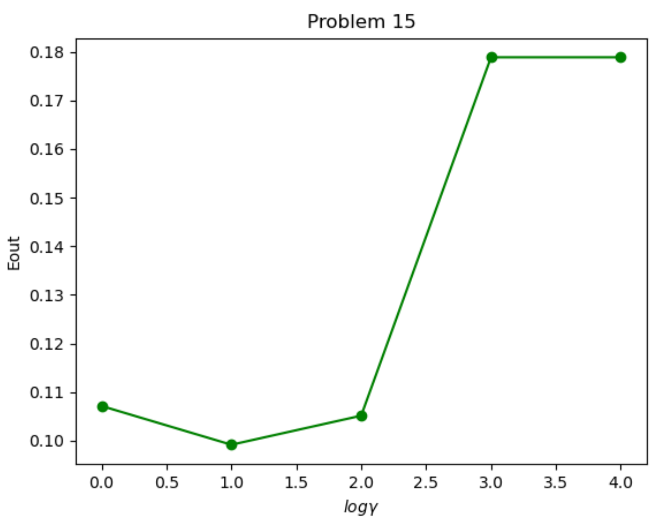
Theoretically, as C goes larger, the margin shrinks and thus the number of support vectors should be fewer. One possible explanation for the anomaly that the number of support vectors grows when C goes from 10^{-5} to 10^{-4} is that, when C is too small, the α is also small and the computer couldn't handle small floating points correctly and treat them as 0. So we missed some support vectors.

Problem 14



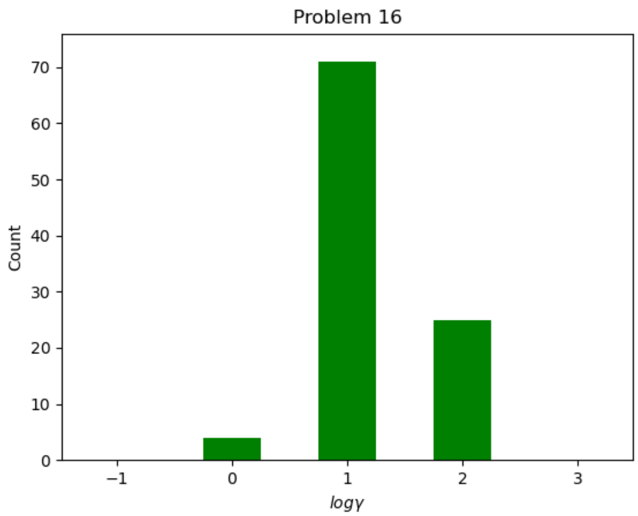
As C grows larger, the margin gets smaller, and thus the distance from a free SV to the hyperplane also gets smaller.

Problem 15



When $\gamma = 1$, E_{out} is the lowest. when γ is too big, i.g. $3, 4$, it is likely the model ovefits the training data and is bad at generalization.

Problem 16



Similar to the observation in the previous problem, $\gamma = 1$ has better performance overall. If γ is too small, the model may not be powerful enough to separate the points. On the other hand, it γ is too big, the model is likely to overfit and perform badly on validation set.

Problem 17

Let N be the number of vectors(data), we have

$$w_i = \sum_{n=1}^{N} \alpha_n y_n z_i$$

Since z_i is a constant feature, it can be put in front of the summation.

$$w_i = z_i \sum_{n=1}^{N} \alpha_n z_n$$

If w is optimal, we have

$$\sum_{n=1}^{n} \alpha_n y_n = 0$$

$$\Longrightarrow w_i = z_i \sum_{n=1}^{N} \alpha_n y_n = 0$$

This shows that w_i is always 0.

Problem 18

The standard hard-margin SVM dual is

$$\min_{\alpha} \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \mathbf{x}_n^T \mathbf{x}_m - \sum_{n=1}^{N} \alpha_n$$

subject to

$$\sum_{n=1}^{N} y_n\alpha_n = 0$$
$$\alpha_n \geq 0 \text{ for } n = 1, 2, \ldots, N$$

We introduce some Lagrange multipliers and define

$$D(\alpha, \lambda) = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} \alpha_n\alpha_m y_n y_m \mathbf{x}_n^T\mathbf{x}_m - \sum_{n=1}^{N}\alpha_n + (\lambda' - \lambda'')\left(\sum_{n=1}^{N} y_n\alpha_n\right) - \sum_{n=1}^{N}\lambda_n\alpha_n$$

And we solve for the optimization problem

$$\min_{\alpha}\max_{\text{all }\lambda\geq 0} D(\alpha, \lambda)$$

The Lagrange multipliers and the max function ensures the original constraints still hold. Using the strong duality property of the optimization problem, we have

$$\min_{\alpha}\max_{\text{all }\lambda\geq 0} D(\alpha, \lambda) = \max_{\text{all }\lambda\geq 0}\min_{\alpha} D(\alpha, \lambda)$$

And we solve for the latter. For optimality, we set the derivatives to 0.

$$\frac{\partial D(\alpha, \lambda)}{\partial \alpha_i} = \sum_{n=1}^{N}\alpha_n y_n y_i \mathbf{x}_n^T\mathbf{x}_i - 1 + (\lambda' - \lambda'')y_i - \lambda_i = 0$$
$$\Longrightarrow \sum_{n=1}^{N}\alpha_n y_n y_i \mathbf{x}_n^T\mathbf{x}_i = 1 - (\lambda' - \lambda'')y_i + \lambda_i$$

Now we let

$$Q = \begin{bmatrix} y_1y_1\mathbf{x}_1^T\mathbf{x}_1 & \ldots & y_1y_N\mathbf{x}_1^T\mathbf{x}_N \\ y_2y_1\mathbf{x}_2^T\mathbf{x}_1 & \ldots & y_2y_N\mathbf{x}_1^T\mathbf{x}_N \\ \vdots & \ddots & \vdots \\ y_Ny_1\mathbf{x}_N^T\mathbf{x}_1 & \ldots & y_Ny_N\mathbf{x}_N^T\mathbf{x}_N \end{bmatrix}$$
$$\alpha = \begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_N\end{bmatrix}, y = \begin{bmatrix}y_1\\ y_2\\ \vdots\\ y_N\end{bmatrix}, \lambda = \begin{bmatrix}\lambda_1\\ \lambda_2\\ \vdots\\ \lambda_N\end{bmatrix}, 1_N = \left.\begin{bmatrix}1\\ 1\\ \vdots\\ 1\end{bmatrix}\right\}N$$

and let $Q^\dagger$ be the pseudo inverse of Q, then we have

$$Q\alpha = 1_N - (\lambda' - \lambda'')y + \lambda \tag{1}$$
$$\alpha = Q^\dagger(1_N - (\lambda' - \lambda'')y + \lambda) \tag{2}$$
$$D(\alpha, \lambda) = \frac{1}{2}\alpha^T Q\alpha - 1_N^T\alpha + (\lambda' - \lambda'')\alpha^T y - \alpha^T\lambda \tag{3}$$

Using (1), (3) can be written as

$$D(\alpha, \lambda) = \frac{1}{2}\alpha^T(1_N - (\lambda' - \lambda'')y + \lambda) - 1_N^T\alpha + (\lambda' - \lambda'')\alpha^T y - \alpha^T\lambda$$
$$= -\frac{1}{2}\alpha^T(1_N - (\lambda' - \lambda'')y + \lambda)$$

and using (2)

$$-\frac{1}{2}\alpha^T(1_N - (\lambda' - \lambda'')y + \lambda) = -\frac{1}{2}(Q^\dagger(1_N - (\lambda' - \lambda'')y + \lambda))^T(1_N - (\lambda' - \lambda'')y + \lambda)$$
$$= -\frac{1}{2}(1_N - (\lambda' - \lambda'')y + \lambda)^T(Q^\dagger)^T(1_N - (\lambda' - \lambda'')y + \lambda)$$

Now the problem can be stated as

$$\begin{aligned}
&\max_{\text{all } \lambda \geq 0} -\frac{1}{2}(1_N - (\lambda' - \lambda'')y + \lambda)^T (Q^\dagger)^T (1_N - (\lambda' - \lambda'')y + \lambda) \\
\Longrightarrow &\min_{\text{all } \lambda \geq 0} \frac{1}{2}(1_N - (\lambda' - \lambda'')y + \lambda)^T (Q^\dagger)^T (1_N - (\lambda' - \lambda'')y + \lambda)
\end{aligned}$$

subject to

$$\lambda_i \geq 0 \text{ for } i = 1, 2, \ldots, N \text{ and } \lambda', \lambda'' \geq 0$$

which is a standard QP problem.