Warm-up

Q1

1. Recall that a function $f : \mathbb{R}^n \to \mathbb{R}$ is convex if for all $x, y \in \mathbb{R}^n$ and $\lambda \in [0,1], \lambda f(x) + (1-\lambda)f(y) \geq f(\lambda x + (1-\lambda)y)$. Using this definition, show that

   a. $f(x) = wf_1(x)$ is a convex function for $x \in \mathbb{R}^n$ whenever $f_1 : \mathbb{R}^n \to \mathbb{R}$ is a convex function and $w \geq 0$

$$\begin{aligned} f(\lambda x + (1-\lambda)y) &= wf_1(\lambda x + (1-\lambda)y) \\ &\leq \lambda(wf_1(x)) + (1-\lambda)(wf_1(y)) \\ &\leq w(\lambda f_1(x) + (1-\lambda)f_1(y)) \end{aligned}$$

   This holds because w is a non-negative scalar, that is, w will not flip the inequality sign.

   b. $f(x) = f_1(x) + f_2(x)$ is a convex function for $x \in \mathbb{R}^n$ whenever $f_1 : \mathbb{R}^n \to \mathbb{R}$ and $f_2 : \mathbb{R}^n \to \mathbb{R}$ are convex functions

$$\begin{aligned} f(\lambda x + (1-\lambda)y) &= (f_1 + f_2)(\lambda x + (1-\lambda)y) \\ &= f_1(\lambda x + (1-\lambda)y) + f_2(\lambda x + (1-\lambda)y) \\ &\leq \lambda f_1(x) + \lambda f_2(x) + (1-\lambda)f_1(y) + (1-\lambda)f_2(y) \\ &= \lambda(f_1 + f_2)(x) + (1-\lambda)(f_1 + f_2)(y) \\ &= \lambda f(x) + (1-\lambda)f(y) \end{aligned}$$

   c. $f(x) = \max\{f_1(x),\ f_2(x)\}$ is a convex function for $x \in \mathbb{R}^n$ whenever $f_1 : \mathbb{R}^n \to \mathbb{R}$ and $f_2 : \mathbb{R}^n \to \mathbb{R}$ are convex functions

$$\begin{aligned} f(\lambda x + (1-\lambda)y) &= max\{f_1(\lambda x + (1-\lambda)y),\ f_2(\lambda x + (1-\lambda)y)\} \\ &\leq max\{\lambda f_1(x) + (1-\lambda)f_1(y),\ \lambda f_2(x) + (1-\lambda)f_2(y)\} \\ &\leq max\{\lambda f_1(x),\ \lambda f_2(x)\} + max\{(1-\lambda)f_1(y),\ (1-\lambda)f_2(y)\} \\ &\leq \lambda max\{f_1(x),\ f_2(x)\} + (1-\lambda)max\{f_1(y),\ f_2(y)\} \\ &= \lambda f(x) + (1-\lambda)f(y) \end{aligned}$$

Q2

(a) By solving function $x^2 - 2x = |x|$, we have root $x = 0, 3$, therefore we can write $f(x)$ in another form,

$$f(x) = \begin{cases} x, 0 \leq x \leq 3 \\ x^2 - 2x, otherwise \end{cases}$$

It is obvious that function $f(x)$ is not differentiable at point 0 and 3. Thus, the subgradient at $x = 0$ is not unique, the minimum value is $(x^2 - 2x)' \mid_{x=0} = -2$, and the maximum value is 1. We can take any value in closed interval $[-2, 1]$ as the subgradient for $f(x)$ at $x = 0$. As for $x = -2$, the corresponding subgradient is $(x^2 - 2x)' \mid_{x=-2} = -6$.

(b) Similarly, we can write g(x) as

$$g(x) = \begin{cases} (x-2)^2, x \leq 1.5 \\ (x-1)^2, otherwise \end{cases}$$

When at point $x = 1.5$, $g(x)$ is not differentiable, the minimum subgradient is $((x-2)^2)' \mid_{x=1.5} = -1$ and the maximum subgradient is $((x-1)^2)' \mid_{x=1.5} = 1$. Hence, we can take any subgradient in interval $[-1, 1]$. For $x = 0$, the corresponding subgradient is $((x-2)^2)' \mid_{x=0} = -4$.

Problem 1: Perceptron Learning

Q1

(1) The number of iterations to find the perfect classifier: 46 or 47

(2) The values of w and b for the first three iteration:

$w^{(1)} = [1278.99646108, 460.06125801, -108.55851404, -1672.31572948]$ $b^{(1)} = -354.0$
$w^{(2)} = [1307.29472974, 432.74778799, -27.55191988, -1523.78895446]$ $b^{(2)} = -493.0$
$w^{(3)} = [1255.18981362, 425.50402882, 18.7965404, -1434.66754197]$ $b^{(3)} = -625.0$

(3) The final weights and bias:
$w = [685.79932892, 243.89947473, 8.24199193, -797.62505314]$ $b = -1485.0$

Q2

(1) The number of iterations to find the perfect classifier: 1091000

(2) The values of w and b for the first three iteration:
$w^{(1)} = [4.61754424, 2.46967938, 1.96766079, -1.81335551]$ $b^{(1)} = -1.0$
$w^{(2)} = [4.61754424, 2.46967938, 1.96766079, -1.81335551]$ $b^{(2)} = -1.0$
$w^{(3)} = [3.45322288, 0.16943482, 2.62801595, -4.64709851]$ $b^{(3)} = -2.0$

(3) The final weights and bias:
$w = [149.27714019, 52.53347317, 1.67167265, -172.89194014]$ $b = -322.0$

Q3

The observations can be divided into two categories. The first one is **fixed step size**(constant), and the other one is **varying step size**.

For any fixed step size such as $0.01, 0.1, 0.5, 1, 5, 10$, the total number of iterations for the algorithm to converge will not change. In other words, the rate of convergence is also a constant. This is because that (1) the gradient of w and b for the first iteration is a fixed value; (2) the initial value of w and b are 0. With these two facts, assume we have two fixed step size a and b, and we denote the w in the first iteration for step size a and b is $w_a^{(1)}$ and $w_b^{(1)}$, respectively. We will have

$$\frac{w_a^{(1)}}{w_b^{(1)}} = \frac{a}{b}$$

For example, when the step size is 0.5, we will obtain

$w^{(1)} = [639.49823054, 230.03062901, -54.27925702, -836.15786474]$

which is exactly half of the $w^{(1)}$ in Q1. This observation holds true for any $w_a^{(i)}$ and $w_b^{(i)}$. Therefore, the total number of iterations for the algorithm to converge will be a fix number.

When varying step size is used, consider general function $1/(a + bt^c)$, the detailed results are shown in Table 1.

Table 1: Number of iterations to converge for different varying step size

$\boldsymbol{\gamma_i(t)}$	$\boldsymbol{\gamma_i(1)}$	$\boldsymbol{\gamma_i(100)}$	**#Iteration to converge**
1	1	1	46
$\frac{1}{\sqrt{t}}$	1	0.1	188
$\frac{1}{10+t}$	0.091	0.0091	96
$\frac{1}{5+t}$	0.167	0.009524	388
$\frac{1}{3+t}$	0.25	0.009709	2172
$\frac{1}{3.5+0.5t}$	0.25	0.0187	179

From the results above, there are two observations. (1) The slower the step size decreases, the faster the algorithm converges. For example, consider the function $\gamma_1(t)$ and $\gamma_2(t)$, they have the same value when $t = 1$, which means that the value of w in the first iteration for these two step sizes are same. However, for $\gamma_1(t)$, it outputs a constant value, and the step size never decreases. As for $\gamma_2(t)$, the step size decreases faster when compared to $\gamma_1(t)$, and the value of step size is 0.1 when $t = 100$. Therefore, it needs more iterations to converge. What is more, the comparison between $\gamma_5(t)$ and $\gamma_6(t)$ again proves the observation. (2) If the step size decreases at same rate (i.e., have same b and c), the bigger the a is, the faster the algorithm converges. Consider function $\gamma_3(t)$, $\gamma_4(t)$ and $\gamma_5(t)$, we can see that $\gamma_3(t)$ converges fastest since it has the largest a, which proves the observation.

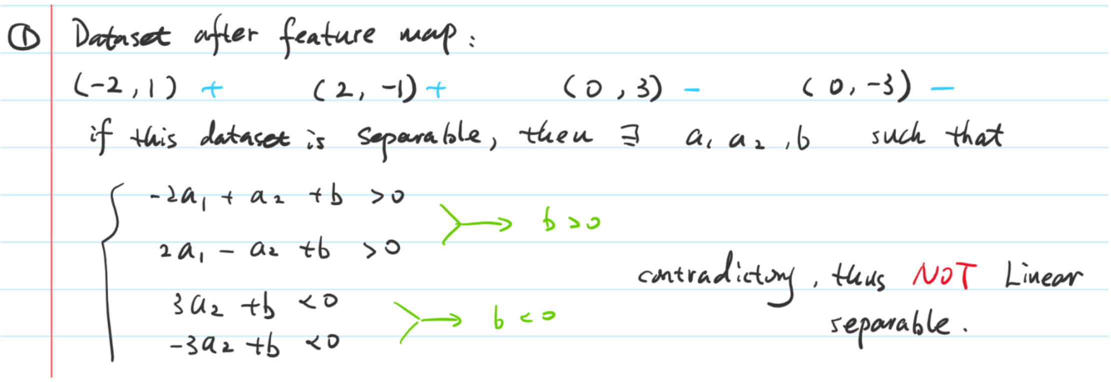
Q4

Generally, when the data is not separable, the algorithm cannot converge. For a two-dimensional data set contains both class labels, the minimum number of points such that the data not separable is 3. Specifically, when three points lie on the same line and middle one has different label, the data set will not be separable. For example, consider the data set D that contains the following three points:

$$x^{(1)} = (1, 1), y^{(1)} = 1 \qquad x^{(2)} = (2, 1), y^{(2)} = -1 \qquad x^{(3)} = (3, 1), y^{(3)} = 1$$

It is obvious that D is not separable, when apply the Perceptron to this dataset with step size $\gamma_t = 1$, the gradient can never be zero and w, b are updated in a periodic way. Specifically, for this example, the period is 2, which means $w^{(i)} = w^{(i\%2)}, b^{(i)} = b^{(i\%2)}$. Hence, the algorithm cannot converge.

Problem 2



① Dataset after feature map:

(-2, 1) + (2, -1) + (0, 3) − (0, -3) −

if this dataset is separable, then ∃ a_1, a_2, b such that

$$\left\{\begin{array}{ll} -2a_1 + a_2 + b > 0 & \\ 2a_1 - a_2 + b > 0 & \longrightarrow b > 0 \\ 3a_2 + b < 0 & \\ -3a_2 + b < 0 & \longrightarrow b < 0 \end{array}\right.$$

contradictory, thus NOT Linear separable.

(b)

$D' = D(\phi(\vec{x}), y) = \{([1,1,1],1), ([1,1,-1],-1), ([1,1,1],1), ([1,1,-1],-1)\}$
Let $\vec{x'} = \phi(\vec{x})$

If D' is linearly separable, then $\exists \vec{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$ and b where, $a_1, a_2, a_3, b \in R$ such that

$$\forall \vec{x'}, y \in D', y = \vec{a}^T x + b$$

Or,

$$a_1 x'_1 + a_2 x'_2 + a_3 x'_3 + b = y$$

$\Rightarrow$ The following system of equations must be consistent.

$$a_1 + a_2 + a_3 + b = 1 \tag{1}$$

$$a_1 + a_2 - a_3 + b = -1 \tag{2}$$

Subtract equations (1) and (2)

$$\Rightarrow a_3 = 1 \tag{3}$$

Substituting equation (3) in (1)

$$\Rightarrow a_1 + a_2 + b = 0 \tag{4}$$

This system of equations is consistent and has infinitely many solutions.
Any hyperplane of the form $\vec{a}^T \vec{x} + b$ where the constraints given by equations (3) and (4) are satisfied will be a linear separator on the dataset D'.

One such separator is $f(x'_1, x'_2, x'_3) = [1,1,1] \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} + (-2) = 0$

On D',

$$f(\vec{x'}^{(1)}) = f(\vec{x'}^{(3)}) = f(1,1,1) = 1 > 0$$
$$f(\vec{x'}^{(2)}) = f(\vec{x'}^{(4)}) = f(1,1,-1) = -1 < 0$$

Therefore, the hyperplane $f(x'_1, x'_2, x'_3) = [1,1,1] \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} + (-2) = 0$ separates the dataset $D(\phi(\vec{x}), y)$.

(c)

On applying the feature transformation

$$\phi(x_1, x_2) = \begin{bmatrix} exp(x_1) \\ exp(x_2) \end{bmatrix}$$

The transformed dataset looks as follows:

$D' = D(\phi(\vec{x}), y) = \{([e^{-1}, e^{-1}], 1), ([e^{1}, e^{-1}], -1), ([e^{1}, e^{1}], 1), ([e^{-1}, e^{1}], -1)\}$
Let $\vec{x'} = \phi(\vec{x})$
If D' is linearly separable, then $\exists \vec{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$ and b where, $a_1, a_2, b \in R$ such that

$$\forall \vec{x'}, y \in D', y = \vec{a}^T x + b$$

Or,

$$a_1 x'_1 + a_2 x'_2 + b = y$$

$\Rightarrow$ The following system of equations must be consistent.

$$e^{-1} a_1 + e^{-1} a_2 + b = 1 \tag{1}$$

$$e a_1 + e^{-1} a_2 + b = -1 \tag{2}$$

$$e a_1 + e a_2 + b = 1 \tag{3}$$

$$e^{-1} a_1 + e a_2 + b = -1 \tag{4}$$

We have 4 equations and 3 unknowns. Therefore, we will solve the first 3 equations and check if their solution (if any) is consistent with equation ((4))
From equation (1), we get

$$\Rightarrow a_1 + a_2 = e(1 - b) \tag{5}$$

From equation (3), we get

$$\Rightarrow a_1 + a_2 = (1 - b)e^{-1} \tag{6}$$

Therefore,

$$e(1 - b) = e^{-1}(1 - b) \Rightarrow b = 1 \tag{7}$$

$$\Rightarrow a_1 + a_2 = 0 \tag{8}$$

Substitute equation (7) into (2)

$$e a_1 + e^{-1} a_2 = -2 \Rightarrow e^2 a_1 + a_2 = -2e \tag{9}$$

Solving equations (8) and (9)

$$a_1 = \frac{-2e}{e^2 - 1}, a_2 = \frac{2e}{e^2 - 1} \tag{10}$$

Substitute the values of a_1, a_2, b into equation (4)

$$e^{-1} \frac{-2e}{e^2 - 1} + e \frac{2e}{e^2 - 1} + 1 = 3 \neq 1 \tag{11}$$

$\Rightarrow$ This system of equations is inconsistent and has no solution. Therefore, no linear separator exists for the dataset $D(\phi(\vec{x}), y)$.

(d)

On applying the feature transformation

$$\phi(x_1, x_2) = \begin{bmatrix} x_1 sin(x_2) \\ x_1 \end{bmatrix}$$

$D' = D(\phi(\vec{x}), y) = \{([-sin(-1), -1], 1), ([sin(-1), 1], -1), ([sin(1), 1], 1), ([-sin(1), -1], -1)\}$
Let $\vec{x'} = \phi(\vec{x})$
If D' is linearly separable, then $\exists \vec{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$ and b where, $a_1, a_2, b \in R$ such that

$$\forall \vec{x'}, y \in D', y = \vec{a}^T x + b$$

Or,

$$a_1 x'_1 + a_2 x'_2 + b = y$$

$\Rightarrow$ The following system of equations must be consistent.

$$-sin(-1)a_1 - a_2 + b = 1 \Rightarrow sin(1)a_1 - a_2 + b = 1 \tag{1}$$

$$sin(-1)a_1 + a_2 + b = -1 \Rightarrow -sin(1)a_1 + a_2 + b = -1 \tag{2}$$

$$sin(1)a_1 + a_2 + b = 1 \tag{3}$$

$$-sin(1)a_1 - a_2 + b = -1 \tag{4}$$

We have 4 equations and 3 unknowns. Therefore, we will solve the first 3 equations and check if their solution (if any) is consistent with equation ((4))
Add equations (1) and (2)

$$\Rightarrow b = 0 \tag{5}$$

Substitute equation (5) into (2) and (3) and add them

$$\Rightarrow a_2 = 0 \tag{6}$$

Substitute equations (5) and (6) into (3)

$$\Rightarrow a_1 = \frac{1}{sin(1)} \tag{7}$$

Check if the values of a_1, a_2, b are consistent with equation (4)

$$-sin(1)\frac{1}{sin(1)} - 0 + 0 = -1 = -1$$

Therefore, these equations are consistent.
One such linear separator is $f(\vec{x'}) = [\frac{1}{sin(1)}, 0]\vec{x'} = 0$
On D',

$$f(\vec{x'}^{(1)}) = f(-sin(-1), -1) = f(sin(1), -1) = \frac{1}{sin(1)} sin(1) = 1 > 0$$

$$f(\vec{x'}^{(2)}) = f(sin(-1), 1) = f(-sin(1), 1) = \frac{1}{sin(1)}(-1)sin(1) = -1 < 0$$

$$f(\vec{x'}^{(3)}) = f(sin(1), 1) = \frac{1}{sin(1)} sin(1) = 1 > 0$$

$$f(\vec{x'}^{(4)}) = f(-sin(1), -1) = \frac{1}{sin(1)}(-1)sin(1) = -1 < 0$$

Therefore, the hyperplane $f(x'_1, x'_2) = [\frac{1}{sin(1)}, 0] \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix} = 0$ separates the dataset $D(\phi(\vec{x}), y)$.

2.

Dataset:

$$D(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, y)$$

Linear regression on two-dimensional data:

$$f(x_1, x_2) = a_0 + a_{10}x_1 + a_{11}x_2$$

Quadratic regression on two-dimensional data:

$$f(x_1, x_2) = a_0 + a_{10}x_1 + a_{11}x_2 + a_{20}{x_1}^2 + a_{21}x_1x_2 + a_{22}{x_2}^2$$

Polynomial regression with a polynomial of degree k on two-dimensional data:

$$f(x_1, x_2) = a_0 + \sum_{i=1}^{k}\sum_{j=0}^{i} a_{ij}{x_1}^{i-j}{x_2}^j$$

It should be noted that though the data has been transformed to a k-degree polynomial feature space, the model stays linear.

$$\phi(\overrightarrow{x_1, x_2})^T = [1, x_1, x_2, ..., {x_1}^k, ..., {x_2}^k]$$

There are $(i{+}1)$ elements in the feature vector with degree i : $[{x_1}^i, {x_1}^{i-1}x_2, ..., x_1{x_2}^{i-1}, {x_2}^i]$
Size of the feature space $(size_{fs}) = \sum_{i=0}^{k}(i+1) = \underline{\frac{(k+1)(k+2)}{2}}$

$$f(x_1, x_2) = \vec{a}^T\phi(\overrightarrow{x_1, x_2})$$

where,

$$\vec{a}^T = [a_0a_{10}a_{11}, ..., a_{k0}, ..., a_{kk}]$$

Assuming that there are total M data points, the loss function can be defined as follows if we use average squared loss function

$$L(f) = \frac{1}{M}\sum_{m=1}^{M}(f(\vec{x}^{(m)}) - y^{(m)})^2$$

$$L(f) = \frac{1}{M}\sum_{m=1}^{M}((\underline{a_0} + \sum_{i=1}^{k}\sum_{j=0}^{i} \overset{\text{unknown Parameters}}{\underline{a_{ij}}}({x_1}^{i-j}{x_2}^j)^{(m)}) - y^{(m)})^2$$

On using standard gradient descent to minimize the loss:

$$\nabla L(f)_{a_0} = \frac{2}{M}\sum_{m=1}^{M}((a_0 + \sum_{i=1}^{k}\sum_{j=0}^{i} a_{ij}({x_1}^{i-j}{x_2}^j)^{(m)}) - y^{(m)})$$

$$\nabla L(f)_{a_{ij}} = \frac{2}{M}\sum_{m=1}^{M}((a_0 + \sum_{i=1}^{k}\sum_{j=0}^{i} a_{ij}({x_1}^{i-j}{x_2}^j)^{(m)}) - y^{(m)})({x_1}^{i-j}{x_2}^j)^{(m)}$$

The coefficients would be updated as follows:

$$a_0 = a_0 - \gamma_t\nabla L(f)_{a_0}$$

$$a_{ij} = a_ij - \gamma_t\nabla L(f)_{a_{ij}}$$

Assuming that all the coefficients a_{ij} are updated in parallel, the complexity of a single iteration of this algorithm can be evaluated as follows:
Complexity = O(number of data points x size of feature space) = $O(M \times size_{fs})$

Problem 3

Q1

$$L(f) = \frac{1}{M}\sum_m (f(x_m) - y_m) = \frac{1}{M}\sum_m (e^{ax_m+b} - y_m)^2$$

Q2

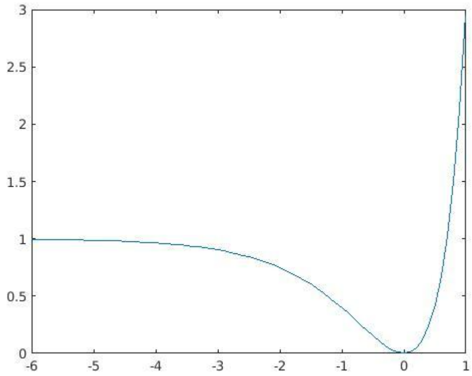
$$a_{t+1} = a_t - \gamma_t \nabla f(a_t)$$
$$b_{t+1} = b_t - \gamma_t \nabla f(b_t)$$

Where the partial derivatives with respect to a and b of the squared loss function are the following.

$$\frac{\partial L}{\partial a} = \frac{1}{M}\sum_m 2 \cdot (e^{ax_m+b} - y_m) \cdot e^{ax_m+b} \cdot x_m$$
$$\frac{\partial L}{\partial b} = \frac{1}{M}\sum_m 2 \cdot (e^{ax_m+b} - y_m) \cdot e^{ax_m+b}$$

Q3

The loss function is not convex because $(e^{ax^m+b} - y^m)^2$ is not guaranteed to be convex. For example, consider $x^m = 1, y^m = 1, b = 0$, we have function $(e^a - 1)^2$, the graph of this function is like



You can see the function is not convex.

Problem 4: Support Vector Machines

Assume the function of hyperplane is $w^T x + b = 0$. Our objective is to maximize the margin, which is $1/||w||$, such that all the data points lie outside the margin, i.e.,

$$\min_{w,b} \frac{1}{2}||w||^2$$
$$s.t.$$
$$y^{(i)}(w^T x^{(i)} + b) \geq 1, \forall i$$

However, if we directly apply quadratic programming solver on this problem, the solver will tell us there is no solution, which means the data is not separable in the original 4-dimensional space.

So I begin to try non-linear separators. Just like P2:Q2, we can utilize feature vectors that map original samples to a higher dimensional space and then find linear separators in that space. This is equivalent to finding a non-linear separator in the original space. When considering non-linear separator with degree k, we assume the corresponding feature vector is $\phi_k(x)$. Then our problem would be

$$\min_{w,b} \frac{1}{2}||w||^2$$
$$s.t.$$
$$y^{(i)}(w^T \phi_k(x^{(i)}) + b) \geq 1, \forall i$$

First, we first consider curves of degree 2 and the corresponding $\phi_k(x)$ is

$$\phi_2(x) = [x_1^2, x_2^2, x_3^2, x_4^2, x_1x_2, x_1x_3, x_1x_4, x_2x_3, x_2x_4, x_3x_4, x_1, x_2, x_3, x_4]^T$$

In this case, the quadratic programming solver outputs the optimal solution, which means the data is separable under feature vector $\phi_2(x)$. The corresponding weight and bias obtained are

```
Weight and bias is:
 [[ 153.3164314     0.78666497    3.4075616   -86.93400487   -8.84890925
    12.85362414   19.85657039    0.82010415   19.95226068   37.33534355
   -27.22938608  -10.9268032    -0.55728305 -100.67209996]]
 91.72666463354375

Margin is 0.004740337891703417

Sample 7 is support vector [0.01852987 0.50782148 0.71364513 0.69774797]
Sample 43 is support vector [0.67760052 0.01731556 0.22660751 0.91446604]
Sample 346 is support vector [0.56116788 0.94932549 0.95929542 0.9899408 ]
Sample 436 is support vector [0.523322   0.39248601 0.06702511 0.77851889]
Sample 521 is support vector [0.02269783 0.29143716 0.35408261 0.64963031]
Sample 525 is support vector [0.19385208 0.83338556 0.74510427 0.74385454]
Sample 571 is support vector [0.77484911 0.50902121 0.0257518  0.97552753]
Sample 574 is support vector [0.0016966  0.71757684 0.34507596 0.65638682]
Sample 575 is support vector [0.16933499 0.62115084 0.42461128 0.67708347]
Sample 736 is support vector [0.50743509 0.14759421 0.92070003 0.92946127]
Sample 835 is support vector [0.22847848 0.0126919  0.12022385 0.63297825]
Sample 873 is support vector [0.38033462 0.37208327 0.42427464 0.74312689]
Sample 897 is support vector [0.13807235 0.65965397 0.19403925 0.62648398]
Sample 969 is support vector [0.60270148 0.62545054 0.01968289 0.8197326 ]
```

Note that the answer is not unique, any answer that are correct will worth full credit.