EE559 - Homework #5 - Lei Lei

1. (a) Write the criterion function J_{MSE} for a 2-class MSE classifier. Then add an L2 regularizer term (similar to the L2 regularizer term for Ridge Regression). Give any restrictions on $\underline{b}$.

   (b) Solve for the MSE solution using a pseudoinverse technique.

1.(a) $J_{MSE}(\underline{w}) = \frac{1}{N}\sum_{i=1}^{N} \epsilon_i^2 = \frac{1}{N}\sum_{i=1}^{N} [\hat{y}(x_i) - y_i]^2$

$\because \hat{y}(x_i) = \underline{w}^T x_i$

$\therefore J_{MSE}(\underline{w}) = \frac{1}{N}\sum_{i=1}^{N} [\underline{w}^T x_i - y_i]^2$

For a 2-class problem, we can reflect the data pts.

$\therefore J_{MSE}(\underline{w}) = \frac{1}{N}\sum_{n=1}^{N} [g(z_n x_n) - b_n]^2$

$= \frac{1}{N}\sum_{n=1}^{N} [\underline{w}^T z_n x_n - b_n]^2 \quad , b_n > 0$

$= \frac{1}{N} \| \underline{\underline{X}}_r \underline{w} - \underline{b} \|_2^2$

$= \frac{1}{N} (\underline{\underline{X}}_r \underline{w} - \underline{b})^T (\underline{\underline{X}}_r \underline{w} - \underline{b}) \quad , \underline{b} > 0$

Add an L2 regularizer term:

$J_{MSE}(\underline{w}) = \frac{1}{N} \| \underline{\underline{X}}_r \underline{w} - \underline{b} \|_2^2 + \lambda \| \underline{w} \|_2^2$

(b) $J_{MSE}(\underline{w}) = \frac{1}{N} (\underline{\underline{X}}_r \underline{w} - \underline{b})^T (\underline{\underline{X}}_r \underline{w} - \underline{b}) + \lambda \| \underline{w} \|_2^2$

$N J_{MSE}(\underline{w}) = \underline{w}^T \underline{\underline{X}}_r^T \underline{\underline{X}}_r \underline{w} - 2\underline{w}^T \underline{\underline{X}}_r^T \underline{b} + \underline{b}^T \underline{b} + N\lambda \underline{w}^T \underline{w}$

$N \nabla_{\underline{w}} J_{MSE}(\underline{w}) = 2\underline{\underline{X}}_r^T \underline{\underline{X}}_r \underline{w} - 2\underline{\underline{X}}_r^T \underline{b} + 2N\lambda \underline{w}$

Let equation = 0.

$\therefore 2\underline{\underline{X}}_r^T \underline{\underline{X}}_r \underline{w} - 2\underline{\underline{X}}_r^T \underline{b} + 2N\lambda \underline{w} = \underline{0}$

$(\underline{\underline{X}}_r^T \underline{\underline{X}}_r + N\lambda \underline{\underline{I}}) \underline{w} = \underline{\underline{X}}_r^T \underline{b}$

$\therefore \hat{\underline{w}} = (\underline{\underline{X}}_r^T \underline{\underline{X}}_r + N\lambda \underline{\underline{I}})^{-1} \underline{\underline{X}}_r^T \underline{b}$

Q2 is in the last few pages.

3. (a) You are given the following criterion function for a 2-class classification problem:

$$J(\underline{w}) = -\sum_{n=1}^{N} [\![\underline{w}^T z_n \, \underline{x}_n \le 0]\!] \, \underline{w}^T z_n \, \underline{x}_n$$

in which augmented notation is used.

Prove that $J(\underline{w})$ is convex.

Hints:

(i) How can $[\![a \le 0]\!]\, a$ be written using the max[.] function? ($[\![..]\!]$ denotes the indicator function.)

(ii) You may find Discussion 4 notes on convexity helpful.

(b) Suppose you make the classifier nonlinear by using a nonlinear transformation of the feature space first:

$$\underline{x} \rightarrow \underline{\varphi}(\underline{x})$$

so that the criterion function is:

$$J(\underline{w}') = -\sum_{n=1}^{N} \left[\left[\underline{w}'^T z_n \, \underline{\phi}(\underline{x}_n) \le 0\right]\right] \underline{w}'^T z_n \, \underline{\phi}(\underline{x}_n) .$$

Is $J(\underline{w}')$ convex? Prove your answer.

3. (a) $[[a \le 0]]\, a = \begin{cases} a & \text{when } a \le 0 \\ 0 & \text{when } a > 0 \end{cases}$

$\therefore [[a \le 0]]\, a = -\max(0, -a)$

$\therefore J(\underline{w}) = -\sum_{n=1}^{N} [[\underline{w}^T z_n \underline{x}_n \le 0]]\, \underline{w}^T z_n \underline{x}_n$

$= -\sum_{n=1}^{N} -\max(0, -\underline{w}^T z_n \underline{x}_n)$

$= \sum_{n=1}^{N} \max(0, -\underline{w}^T z_n \underline{x}_n)$

$\because g(\underline{w}) = -\underline{w}^T z_n \underline{x}_n$ is an affine function of $\underline{w}$

$\therefore g(\underline{w})$ is convex

Also $\because h(a) = \max(0, a)$ is non-decreasing & convex

$\therefore f(x) = h(g(x))$ is convex

Also $\because f(w_1), f(w_2)$ are convex

$\therefore \Sigma f(w) = J(\underline{w})$ is convex.

(b) $\underline{x} \to \underline{\Phi}(\underline{x})$

$$J(\underline{w}') = -\sum_{n=1}^{N} [[\, \underline{w}'^T z_n \underline{\Phi}(\underline{x}_n) \le 0 \,]]\ \underline{w}'^T z_n \underline{\Phi}(\underline{x}_n)$$

$$= \sum_{n=1}^{N} \max(0, -\underline{w}'^T z_n \underline{\Phi}(\underline{x}_n))$$

Let $g(\underline{w}') = -\underline{w}'^T z_n \underline{\Phi}(\underline{x}_n)$

$\forall \underline{w}_1', \underline{w}_2' \in \text{dom}$, $\theta \in (0,1)$

$\theta g(\underline{w}_1') + (1-\theta) g(\underline{w}_2') = -\theta \underline{w}_1'^T z_n \underline{\Phi}(\underline{x}_n) - (1-\theta) \underline{w}_2'^T z_n \underline{\Phi}(\underline{x}_n)$

$g(\theta \underline{w}_1' + (1-\theta)\underline{w}_2') = -[\theta \underline{w}_1'^T + (1-\theta)\underline{w}_2'^T] z_n \underline{\Phi}(\underline{x}_n)$

$\therefore \theta g(\underline{w}_1') + (1-\theta) g(\underline{w}_2') = g(\theta \underline{w}_1' + (1-\theta)\underline{w}_2')$

$\therefore g(\underline{w}')$ is convex

Also $\because h(a) = \max(0, a)$ is non-decreasing & convex

$\therefore h(g(\underline{w}'))$ is convex

$\therefore \sum h(g(\underline{w}')) = J(\underline{w}')$ is convex

4. In this problem you will solve a constrained optimization problem by hand. You will minimize a function $f(\underline{w})$ with respect to $\underline{w}$, subject to a set of inequality constraints that depend on a set of data points $\underline{u}_i$, $i = 1,2,\cdots,N$. (In part (c) you will use $N = 2$; before then keep N as a variable.) You will then use this result to find a decision boundary and regions for a 2-class classifier based on the 2 given data points.

You will use Lagrangian optimization. Nonaugmented notation will be used throughout this problem.

Consider that we want to minimize the function $f(\underline{w}) = \frac{1}{2}\|\underline{w}\|_2^2$ subject to the constraints $z_i\left(\underline{w}^T\underline{u}_i + w_0\right) \geq 1$ for all $i = 1, 2, \cdots, N$, in which $\underline{u}_i$ is the i^{th} training sample in expanded feature space, and as usual $z_i = \begin{cases} +1, & \underline{u}_i \in S_1 \\ -1, & \underline{u}_i \in S_2 \end{cases}$.

Tip: Write clearly so that μ_i and $\underline{u}_i$ are always distinguishable.

4. (a) Let $\underline{w}^+$ equal to the augmented weights

Let $\underline{u}_i^+$ equal to the augmented data pts.

Then $z_i(\underline{w}^T\underline{u}_i + w_0) = z_i\underline{w}^{+T}\underline{u}_i^+ \geq 1$

As we all know, if the augmented data is correctly classified.

it should satisfy $z_i\underline{w}^{+T}\underline{u}_i^+ > 0$

$\because$ $\underline{w}$ satisfy the constraints $z_i(\underline{w}^T\underline{u}_i + w_0) \geq 1 > 0$

then the training samples will be correctly classified.

(b) (i) $L(\underline{w}, w_0, \underline{\mu}) = \frac{1}{2}\|\underline{w}\|^2 - \underline{\mu}\sum_{i=1}^{N}\left[z_i(\underline{w}^T\underline{u}_i + w_0) - 1\right]$

KKT:

$$\begin{cases} z_i(\underline{w}^T\underline{u}_i + w_0) - 1 \geq 0, & \forall i \\ \mu_i \geq 0, & \forall i \\ \mu_i\left[z_i(\underline{w}^T\underline{u}_i + w_0) - 1\right] = 0, & \forall i \end{cases}$$

(ii) $\nabla_{\underline{w}} L = \frac{1}{2} \times 2\,\underline{w} - \sum_{i=1}^{N}\mu_i\left[z_i(\underline{w}^T\underline{u}_i + w_0) - 1\right]$

$= \underline{w} - \sum_{i=1}^{N}\mu_i z_i \underline{u}_i = 0$

$\therefore$ $\underline{w}^* = \sum_{i=1}^{N}\mu_i z_i \underline{u}_i$

$$\frac{\partial L}{\partial w_0} = -\sum_{i=1}^{N} \mu_i z_i = 0$$

$$\therefore \sum_{i=1}^{N} \mu_i z_i = 0 .$$

(iii) $$L(\underline{w}, w_0, \underline{\mu}) = \frac{1}{2}\|\underline{w}\|^2 - \sum_{i=1}^{N}\left[\mu_i z_i (\underline{w}^T \underline{u}_i + w_0) - \mu_i\right]$$

$$= \frac{1}{2}\left(\sum_{i=1}^{N} \mu_i z_i \underline{u}_i\right)^2 - \sum_{i=1}^{N}\left[\mu_i z_i \cdot \sum_{j=1}^{N} \mu_j z_j \underline{u}_j \cdot \underline{u}_i + \mu_i z_i w_0 - \mu_i\right]$$

$$= \frac{1}{2}\left(\sum_{i=1}^{N} \mu_i z_i \underline{u}_i\right)^2 - \sum_{i=1}^{N}\sum_{j=1}^{N} \mu_i \mu_j z_i z_j \underline{u}_i^T \underline{u}_j + \cancel{\sum_{i=1}^{N} \mu_i z_i w_0}^{\,0} + \sum_{i=1}^{N} \mu_i$$

$$= \frac{1}{2}\left(\sum_{i=1}^{N} \mu_i z_i \underline{u}_i\right)^T \left(\sum_{j=1}^{N} \mu_j z_j \underline{u}_j\right) - \sum_{i=1}^{N}\sum_{j=1}^{N} \mu_i \mu_j z_i z_j \underline{u}_i^T \underline{u}_j + \sum_{i=1}^{N} \mu_i$$

$$= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \mu_i \mu_j z_i z_j \underline{u}_i^T \underline{u}_j - \sum_{i=1}^{N}\sum_{j=1}^{N} \mu_i \mu_j z_i z_j \underline{u}_i^T \underline{u}_j + \sum_{i=1}^{N} \mu_i$$

$$\therefore L_D(\underline{\mu}) = -\frac{1}{2}\left[\sum_{i=1}^{N}\sum_{j=1}^{N} \mu_i \mu_j z_i z_j \underline{u}_i^T \underline{u}_j\right] + \sum_{i=1}^{N} \mu_i$$

$$\begin{cases} \sum_{i=1}^{N} \mu_i z_i = 0 \\ w^* = \sum_{i=1}^{N} \mu_i z_i \underline{u}_i \\ \\ z_i(\underline{w}^T \underline{u}_i + w_0) - 1 \geq 0 \\ \\ \mu_i \geq 0 \\ \mu_i\left[z_i(\underline{w}^T \underline{u}_i + w_0) - 1\right] = 0 \end{cases}$$

(c) $L_D'(\underline{\mu}, \lambda) = \sum_{i=1}^{N} \mu_i - \frac{1}{2}\left[\sum_{i=1}^{N}\sum_{j=1}^{N} \mu_i \mu_j z_i z_j \underline{u}_i^T \underline{u}_j\right] + \lambda\left(\sum_{i=1}^{N} z_i \mu_i\right)$

$\because \sum_{i=1}^{N} \mu_i z_i = 0$ & $\underline{u}_1 \in S_1 \Rightarrow z_1 = 1$ $\underline{u}_2 \in S_2 \Rightarrow z_2 = -1$

$\therefore \mu_1 z_1 + \mu_2 z_2 = \mu_1 - \mu_2 = 0$

$\therefore \mu_1 = \mu_2 = \mu$

$\therefore L_D'(\underline{\mu}, \lambda) = 2\mu - \frac{1}{2}\left[\mu^2 z_1 z_1 \underline{u}_1^T \underline{u}_1 + \mu^2 z_1 z_2 \underline{u}_1^T \underline{u}_2 + \mu^2 z_2 z_1 \underline{u}_2^T \underline{u}_1 + \mu^2 z_2 z_2 \underline{u}_2^T \underline{u}_2\right] + 0$

$= 2\mu - \frac{1}{2}\mu^2\left[1 + (-1)\cdot 0 + (-1)\cdot 0 + 1\right]$

$= 2\mu - \frac{1}{2}\mu^2 \cdot 2 = 2\mu - \mu^2$

$\because \frac{\partial L_D'}{\partial \mu} = 2 - 2\mu = 0$

$\therefore \mu = 1$

$\underline{w}^* = \sum_{i=1}^{N} \mu_i z_i \underline{u}_i = \mu \cdot 1 \cdot \begin{bmatrix} -1 \\ 0 \end{bmatrix} + \mu \cdot (-1) \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$= \mu \begin{bmatrix} -1 \\ 0 \end{bmatrix} - \mu \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \mu \cdot \begin{bmatrix} -1 \\ -1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$

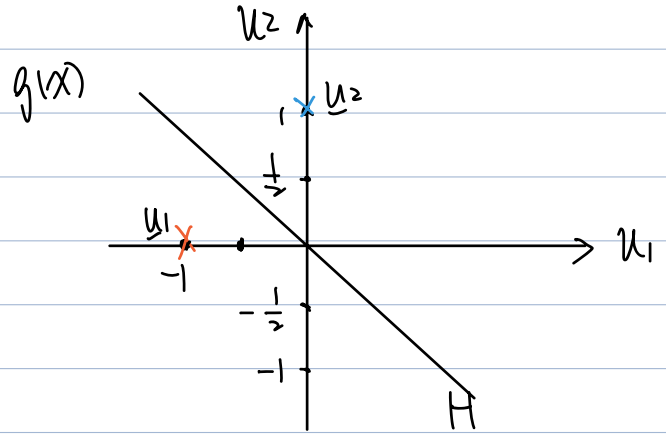
$\because \mu_i \left[z_i (\underline{w}^{*T} \underline{u}_i + w_0) - 1\right] = 0$

$\therefore z_1 \left(\begin{bmatrix} -1 \\ -1 \end{bmatrix}^T \begin{bmatrix} -1 \\ 0 \end{bmatrix} + w_0\right) - 1 = 1 + 0 + w_0 = 1$

$\therefore w_0^* = 0$

$\therefore g(x) = \underline{w}^{*T} \underline{u} + w_0^* = \begin{bmatrix} -1 \\ -1 \end{bmatrix}^T \underline{u} = \begin{bmatrix} -1 \\ -1 \end{bmatrix}^T \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$

$= -u_1 - u_2 - 1 = 0 \qquad u_2 = -u_1$



(a) $|d(H \to \underline{u}_1)| = \frac{g(\underline{u}_1)}{\|\underline{w}^*\|} = \frac{1}{\sqrt{2}}$

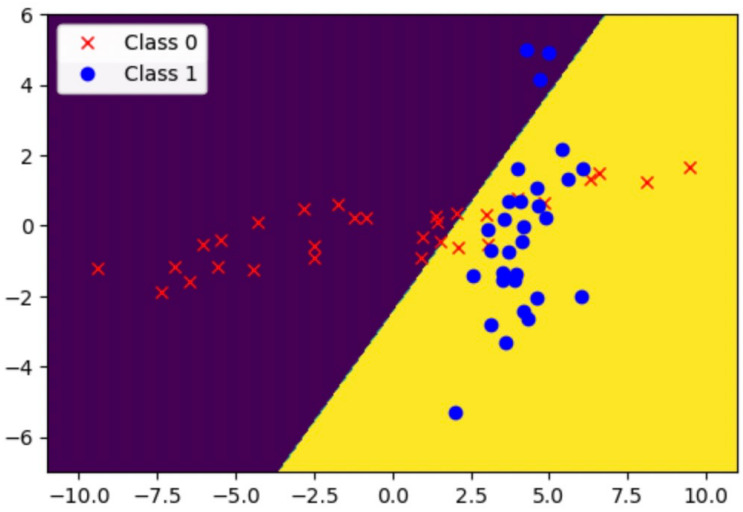
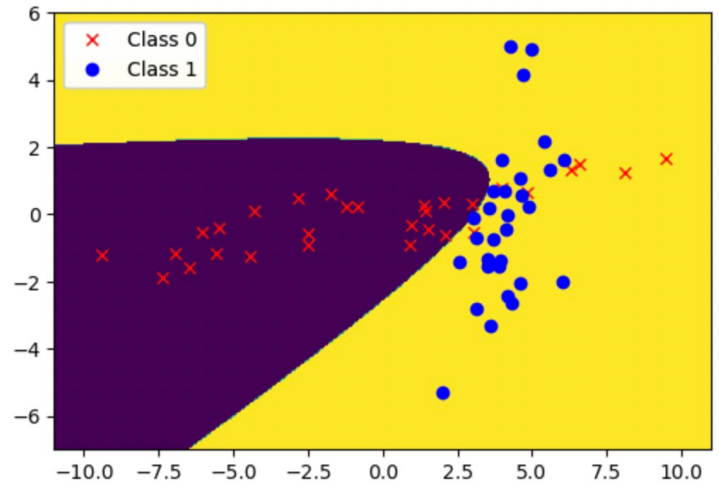
$|d(H \to \underline{u}_2| = \frac{|g(\underline{u}_2)|}{\|\underline{w}^*\|} = \frac{1}{\sqrt{2}}$

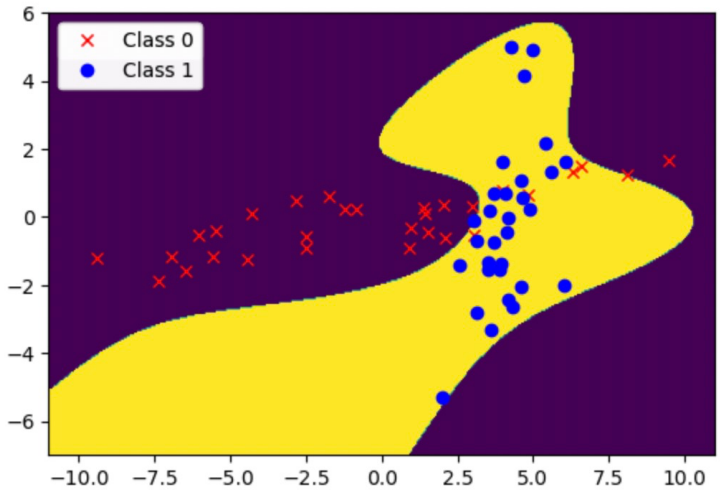
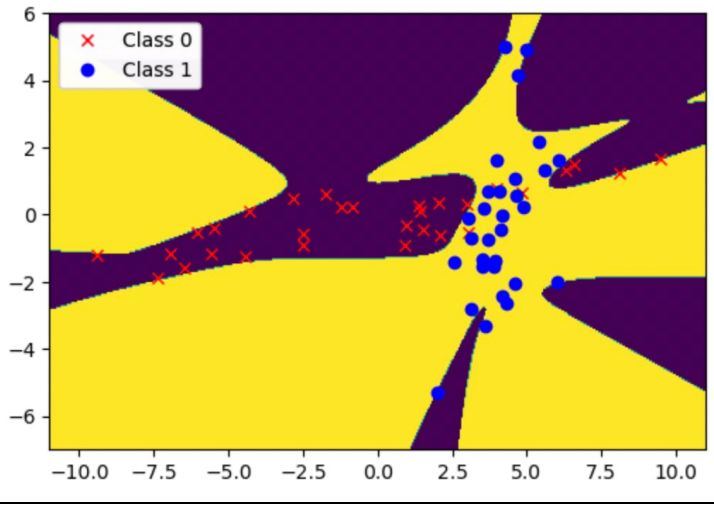
No, the decision boundary H is orthogonal to the line connecting $\underline{u}_1$ and $\underline{u}_2$ and cut it equally.

2.(a) (i) (ii)

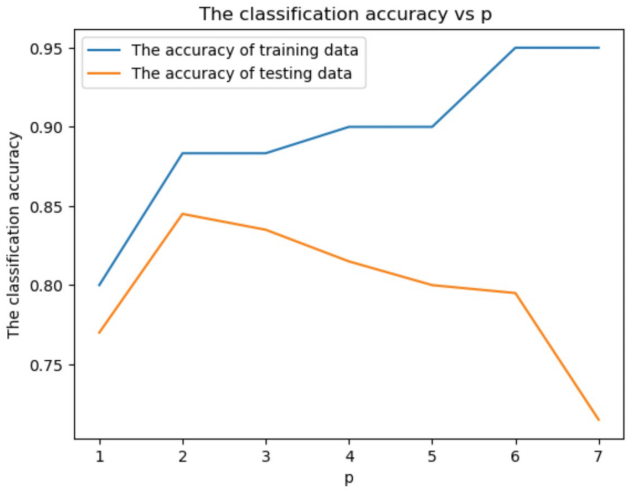
Report the classification accuracy on the **training and test** sets for each value of p.

Plot the **training** data points and the decision regions for p = 1,2,4,7.

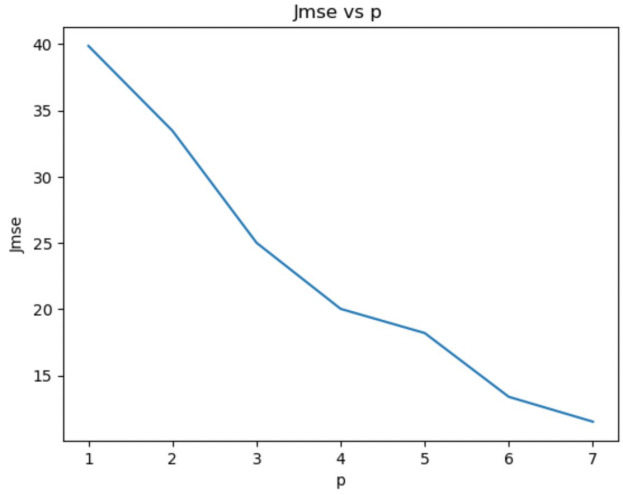
P	Training Accuracy	Testing Accuracy	Plot
1	0.8	0.77	
2	0.8833333 333333333	0.845	
3	0.8833333 333333333	0.835	

4	0.9	0.815	
5	0.9	0.8	
6	0.95	0.795	
7	0.95	0.715	

（iii）Plot the train and test accuracy vs. p on a single plot for all values of p.



(iv) Plot J_{MSE} vs. p for all values of p.



(b) Compare and comment on the results in part (a). In particular, how does the train and test accuracy vary as p increases. Also, how do the decision boundaries appear? Do you see any effect of overfitting?

According to the results, we can find that the train accuracy and test accuracy raised when the p increasing from 1 to 2. The train accuracy keep raise when the p increasing from 2 to 7 and reached 0.95, while the test accuracy keeps going down.

The decision boundary is a linear line when p = 1, and it becomes a curve while p >= 2. When p = 7, The decision boundary is more meandering and only surrounding the training data points.

In my opinion, when p is too big, such that p > 5, there is effect of overfitting. Because the training accuracy is high but the test accuracy is lower than usual.

(c) How many d.o.f. and constraints are there (for each p) and how do they relate to the obtained results?

For p = 1, d.o.f = 3, constraints = 60.

For p = 2, d.o.f = 6， constraints = 60.

For p = 3, d.o.f = 10， constraints = 60.

For p = 4, d.o.f = 15， constraints = 60.

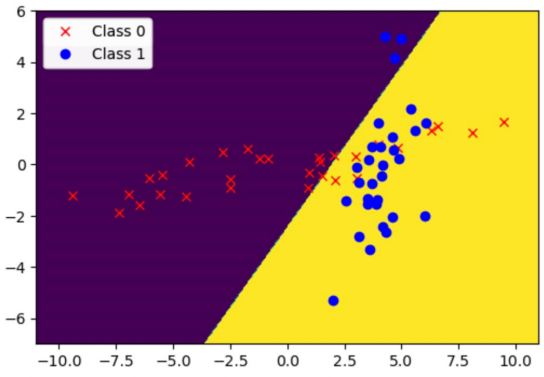
For p = 5, d.o.f = 21， constraints = 60.

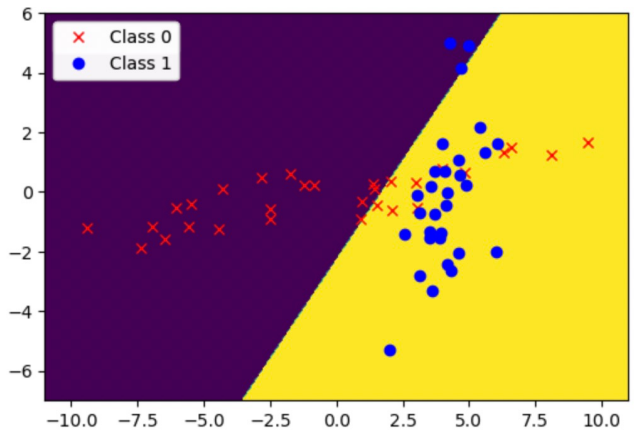
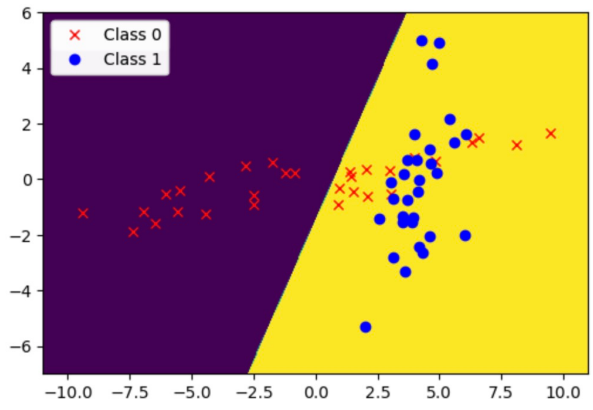
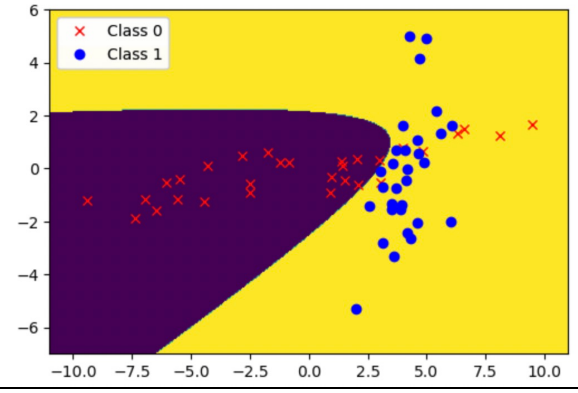
For p = 6, d.o.f = 28， constraints = 60.

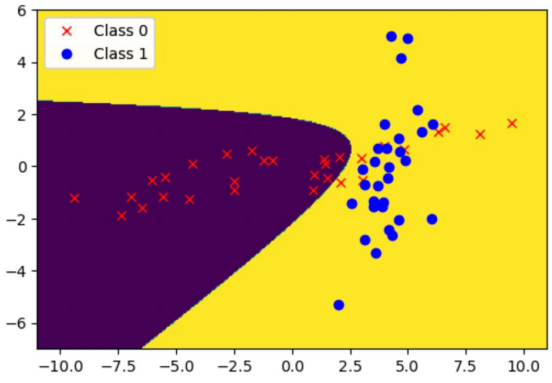
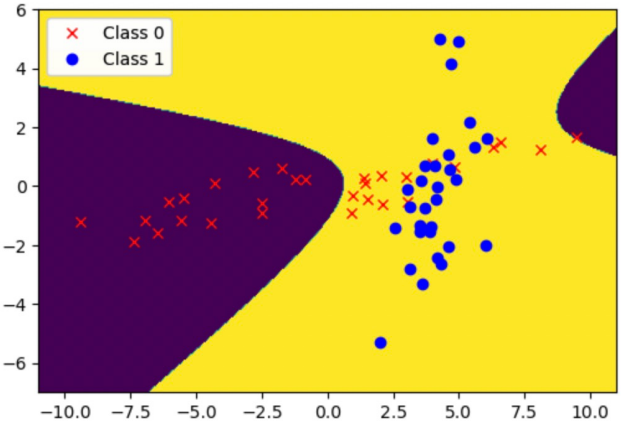
For p = 7, d.o.f = 36， constraints = 60.

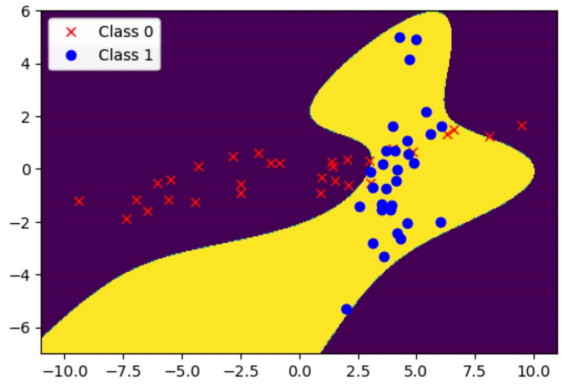
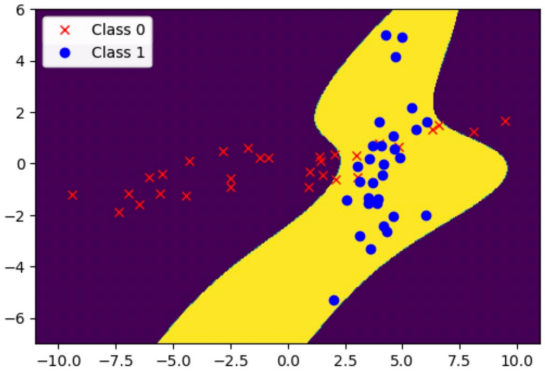
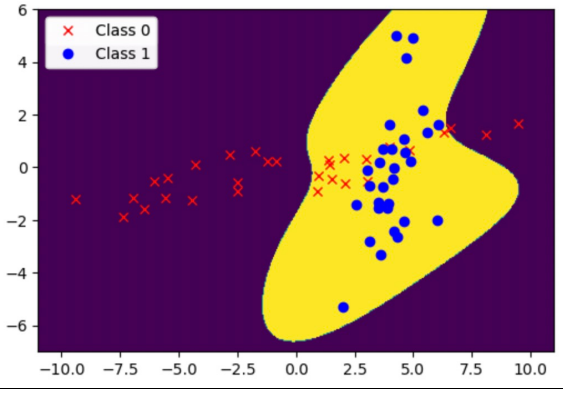
Due to the inequality for the constraints, Nc > (3-10) * d.o.f. , when the accuracy remains the same while the d.o.f. increasing, the model perform well and started overfitting when the d.o.f is too large.

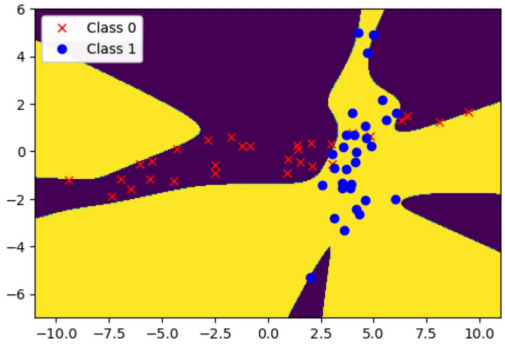
(d)

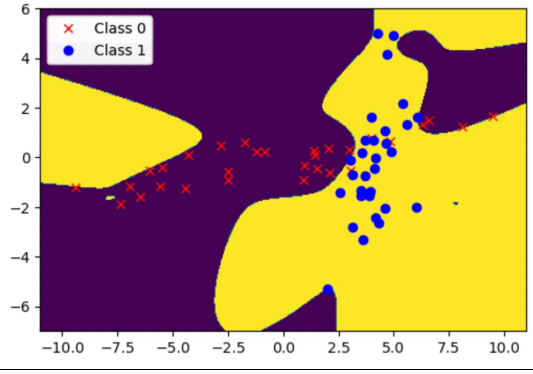
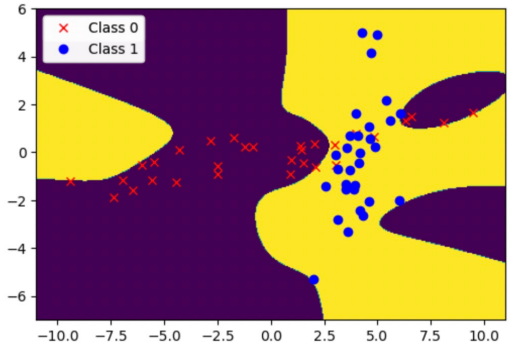
P	lambda	Training Accuracy	Testing Accuracy	Plot
1	0.3	0.8	0.77	
	1	0.8	0.78	
	3	0.8	0.78	

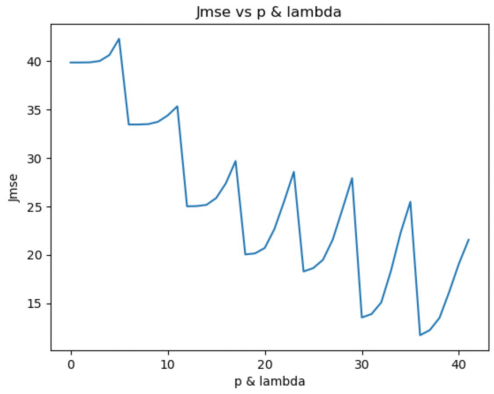
	10	0.7666666666666667	0.79	
	30	0.7666666666666667	0.8	
	100	0.75	0.765	
2	0.3	0.8833333333333333	0.85	
	1	0.8833333333333333	0.85	
	3	0.8833333333333333	0.845	

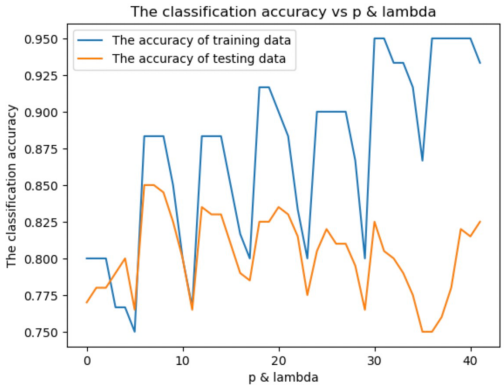
	10	0.85	0.825	
	30	0.8	0.8	
	100	0.7666666666666667	0.765	
3	0.3	0.8833333333333333	0.835	
	1	0.8833333333333333	0.83	
	3	0.8833333333333333	0.83	
	10	0.85	0.81	
	30	0.8166666666666667	0.79	
	100	0.8	0.785	
4	0.3	0.9166666666666666	0.825	

	1	0.916666666666666	0.825	
	3	0.9	0.835	
	10	0.883333333333333	0.83	
	30	0.833333333333334	0.815	
	100	0.8	0.775	
5	0.3	0.9	0.805	
	1	0.9	0.82	
	3	0.9	0.81	
	10	0.9	0.81	

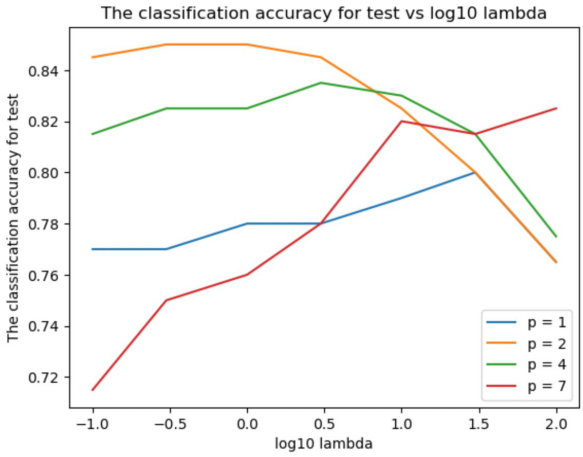
	30	0.8666666666666667	0.795	
	100	0.8	0.765	
6	0.3	0.95	0.825	
	1	0.95	0.805	
	3	0.9333333333333333	0.8	
	10	0.9333333333333333	0.79	
	30	0.9166666666666666	0.775	
	100	0.8666666666666667	0.75	
7	0.3	0.95	0.75	
	1	0.95	0.76	
	3	0.95	0.78	

	10	0.95	0.82	
	30	0.95	0.815	
	100	0.933333333333333	0.825	





(e) Additionally, plot test accuracy vs. log (λ), for p = 1,2,4,7 on a single plot. Use log base 10. (Also consider λ = 0 case, which will be your results from part (a).)



(f) Compare and comment on the results in part (d) and how they relate to results from part (a). Do you see any effect of regularization? Explain briefly.

The classification accuracy increased compared to the results from part (a) especially for the test case classification accuracy. The regularization reduced the effect of overfitting for large p. And the larger lambda, the much effect shows.

(g) Study the sklearn **LinearRegression / Ridge** class and explain how to obtain the trained weight vector. The weight vector can be used to write the equation of the decision boundary / the decision rule. Give the decision rule for $p = 2$ in part (a).

The LinearRegression method of sklearn uses the Least Square method. So, the algorithm updates the weight to find the least square error and if X is the feature matrix and w is the final weight vector , Xw will be the prediction of the LinearRegression model.

The weight vector is [-0.35082056 0.10705408 0.02444723 0.00225307 -0.04160807 0.0613047].

So, the decision boundary is $0.10705408x_1 + 0.02444723x_2 + 0.00225307x_1^2 + 0.0022530x_1x_2 - 0.04160807x_2^2 - 0.3582056 = 0$

If the inequation $0.10705408x_1 + 0.02444723x_2 + 0.00225307x_1^2 + 0.0022530x_1x_2 - 0.04160807x_2^2 - 0.3582056 < 0$, then this data point is assigned to class 1(target = -1), else the data point is assigned to class 0(target = 1).