HW2 Sol p1

a)

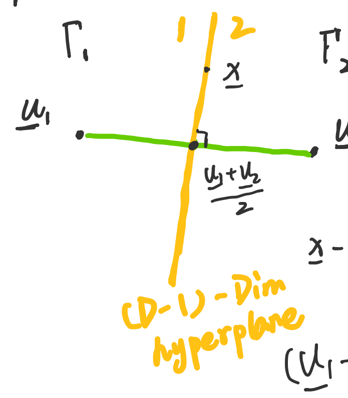
$$g(\underline{x}) = -\|\underline{x}-\underline{u}_1\|_2^2 + \|\underline{x}-\underline{u}_2\|_2^2$$
$$= -(\underline{x}-\underline{u}_1)^T(\underline{x}-\underline{u}_1) + (\underline{x}-\underline{u}_2)^T(\underline{x}-\underline{u}_2)$$
$$= -(\underline{x}^T\underline{x} - 2\underline{u}_1^T\underline{x} + \underline{u}_1^T\underline{u}_1)$$
$$+ (\underline{x}^T\underline{x} - 2\underline{u}_2^T\underline{x} + \underline{u}_2^T\underline{u}_2)$$
$$= 2(\underline{u}_1-\underline{u}_2)^T\underline{x} - (\underline{u}_1^T\underline{u}_1 - \underline{u}_2^T\underline{u}_2)$$

Decision boundary: $g(\underline{x}) = 0$

Decision rule: $g(\underline{x}) \underset{\Gamma_2}{\overset{\Gamma_1}{\gtrless}} 0$ (If the sign of $g(x)$ is different, then $g(\underline{x}) \underset{\Gamma_1}{\overset{\Gamma_2}{\gtrless}} 0$)

a)

Method 2



$\underline{x}$ is a point on the decision boundary

$\underline{x} - \frac{\underline{u}_1+\underline{u}_2}{2} \perp \underline{u}_1 - \underline{u}_2$

$\Downarrow$

$(\underline{u}_1-\underline{u}_2)^T(\underline{x} - \frac{\underline{u}_1+\underline{u}_2}{2}) = 0$

$$g(\underline{x}) = (\underline{u}_1-\underline{u}_2)^T(\underline{x} - \frac{\underline{u}_1+\underline{u}_2}{2})$$
$$= -\underline{u}_2^T\underline{x} + \underline{u}_1^T\underline{x} + \underline{u}_2^T\frac{\underline{u}_1+\underline{u}_2}{2} - \underline{u}_1^T\frac{\underline{u}_1+\underline{u}_2}{2}$$
$$= (\underline{u}_1-\underline{u}_2)^T\underline{x} + \frac{1}{2}(\underline{u}_2-\underline{u}_1)^T(\underline{u}_1+\underline{u}_2)$$
$$= (\underline{u}_1-\underline{u}_2)^T\underline{x} + \frac{1}{2}(\underline{u}_2^T\underline{u}_2 - \underline{u}_1^T\underline{u}_1)$$

b) Yes, it is linear. Because from a), we know $g(x)$ can be expressed as $g(\underline{x}) = \underline{w}^T\underline{x} + w_0$ where

$\underline{w} = 2(\underline{u}_1-\underline{u}_2)$, $w_0 = -(\underline{u}_1^T\underline{u}_1 - \underline{u}_2^T\underline{u}_2)$

c) /

d) $g_m(\underline{x}) = -\|\underline{x}-\underline{u}_m\|_2^2$ (means $g_m(\underline{x}) \uparrow$, closer to $\underline{u}_m$)

$$= -(\underline{x}-\underline{u}_m)^T(\underline{x}-\underline{u}_m)$$
$$= -(\underline{x}^T\underline{x} - 2\underline{u}_m^T\underline{x} + \underline{u}_m^T\underline{u}_m)$$

Since all $g_m(x)$ have the term $\underline{x}^T\underline{x}$, which is a constant of m, we can simplify $g_m(x)$ by dropping it.

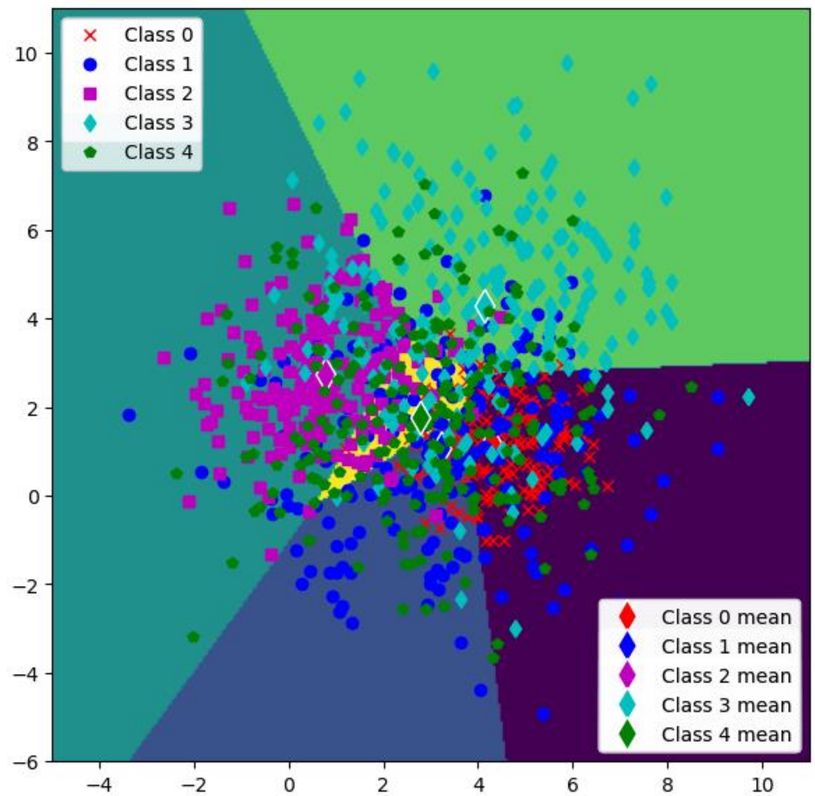
So, $g_m(\underline{x}) = 2\underline{u}_m^T\underline{x} - \underline{u}_m^T\underline{u}_m$

e). Yes, it is linear. It can be expressed as $g_m(\underline{x}) = \underline{w}^T\underline{x} + w_0$ where $\underline{w} = 2\underline{u}_m$ and $w_0 = \underline{u}_m^T\underline{u}_m$

f) Yes. Because the decision rule chooses the class m s.t. $g\_m(x)$ is the maximal.

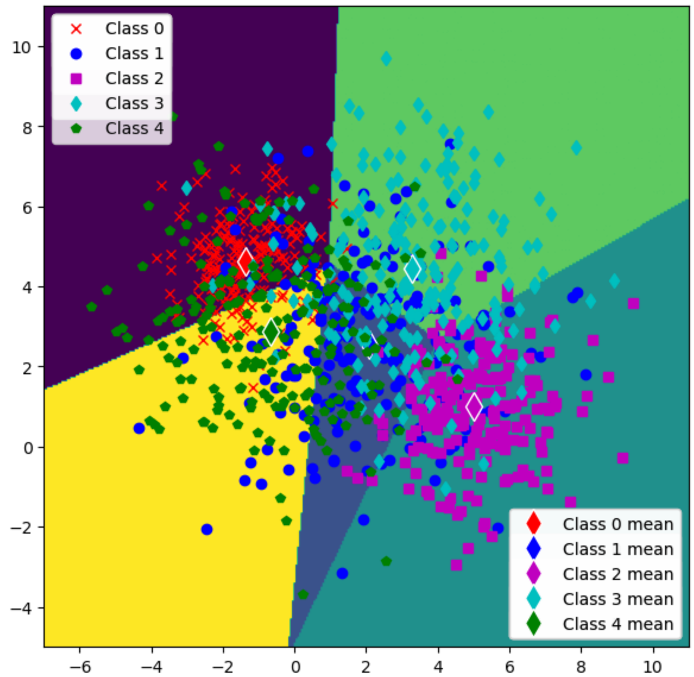
Problem 2)

a) The classification results using all D = 7 features are:
    a. Train: 0.8523
    b. Test: 0.8244
b) The classification results using X1 and X2 features are:
    a. Train: 0.5085
    b. Test: 0.48



The decision boundaries look consistent with the class means (there is large overlap between the data points of different classes).

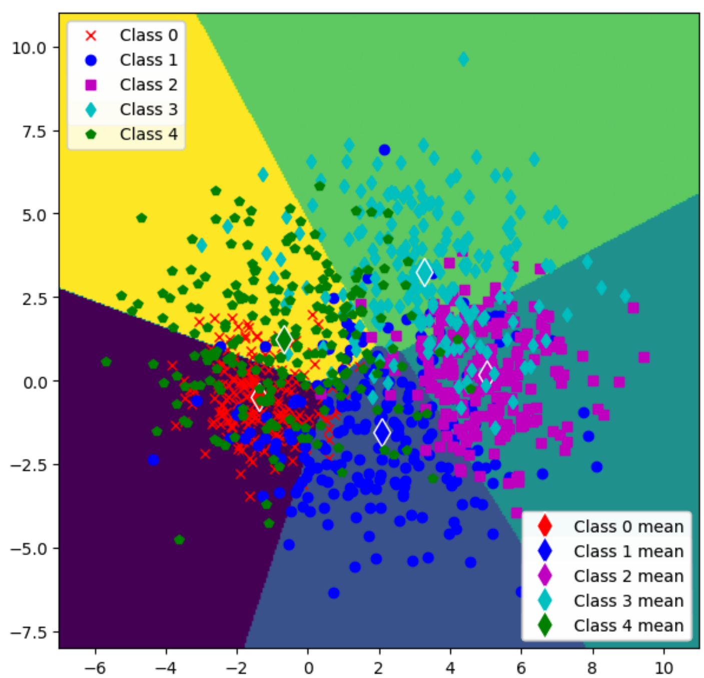
c) The classification results using X3 and X4 features are:
    a. Train: 0.6095
    b. Test: 0.6044



The decision boundaries look consistent with the class means. Compared to (c) it is easier to distinguish the data points of each class (less overlap).

d) The classification results using X3 and X7 features are:

   a. Train: 0.6761
   b. Test: 0.6377



The decision boundaries look consistent with the class means. Compared to (c) & (d) the data points of each class lie mainly inside their respective region.

e) The best training and test accuracy is given using X3 and X7 features (problem (d)). Using all the features results in higher accuracy.

f) There are no intermediate regions and all decision regions are convex.

Sample code for problem 2:

```
class NearestMeans:
    def __init__(self) :
        self.no_of_classes_ = None #number of classes
        self.no_of_features_ = None # number of features
        self.classes_ = None # class labels
        self.means_ = []

    def fit(self,X_data_,y_label_):
        'compute the mean of each class'
        assert X_data_.shape[0] == y_label_.shape[0], 'len(X_data) != len(y_label)'
        y_label_ = y_label_.astype(int)
        self.classes_ = np.unique(y_label_).astype(int)
        self.no_of_classes_ = int(len(self.classes_))
        self.no_of_features_ = X_data_.shape[1]
        for i_class in range(self.no_of_classes_):
            self.means_.append(X_data_[y_label_ == i_class].mean(axis = 0))
        self.means_ = np.array(self.means_)

    def predict(self,X_data_):
        'predict class of each data point'

        dist_mat_ = []
        for i_feat in self.classes_:
            cur_dist_ = np.linalg.norm(X_data_-self.means_[i_feat,:], axis=1,ord = 2)
            dist_mat_.append(cur_dist_)
        dist_mat_ = np.array(dist_mat_)

        return np.argmin(dist_mat_,axis = 0)
```

```
def accuracy_fcn(y_actual_,y_pred_):
    assert y_actual_.shape[0] == y_pred_.shape[0], 'len(y_actual) != len(y_pred)'
    return sum(y_pred_ == y_actual_)/y_actual_.shape[0]
```

```
# predict the output for train set
y_pred_train = nm_clas.predict(X_train)
# predict the output for test set
y_pred_test = nm_clas.predict(X_test,)
print('Train accuracy',accuracy_fcn(y_train,y_pred_train))
print('Test accuracy',accuracy_fcn(y_test,y_pred_test))
```

```
Train accuracy 0.8523809523809524
Test accuracy 0.8244444444444444
```

```
plotDecBoundaries_2(X_train_reduced, y_train, clf_red.means_, fsize=(7,7),legend_on =  True)
```

Problem 3)

a) Yes. $p(m) = O(m)$. If $a = 1$, it becomes $0 \leq 10m - 50 \leq m$ therefore $10m - 50 \leq m$ only holds when $m \leq \frac{50}{9}$. Therefore, a value should be larger than 1. If $a = 11$, the equation $0 \leq 10m - 50 \leq 11m$ holds $\forall m \geq 5$ Thus for $a = 11, m_0 = 5$.

b) Yes. $p(m) = \Omega(m)$. If $b = 1$, it becomes $0 \leq m \leq 10m - 50$ therefore $m \leq 10m - 50$ only holds when $m \geq 6$. Therefore b can be 1 and $m_1 = 6$.

c) Yes. $p(m) = \Theta(m)$. The result follows from (a) and (b). $p(m) = O(m)$ and $p(m) = \Omega(m)$ thus $p(m) = \Theta(m)$.

Problem 4)

(a) Since $p_1(m) = O(q_1(m))$, there exist positive constants c_1 and m_1, such that

$$0 \leq p_1(m) \leq c_1 q_1(m) \quad \forall m \geq m_1$$

Since $p_2(m) = O(q_2(m))$, there exist positive constants c_2 and m_2, such that

$$0 \leq p_2(m) \leq c_2 q_2(m) \quad \forall m \geq m_2$$

Since $p_3(m) = O(q_3(m))$, there exist positive constants c_3 and m_3, such that

$$0 \leq p_3(m) \leq c_3 q_3(m) \quad \forall m \geq m_3$$

If we sum the inequalities, the following equation holds.

$$0 \leq p_1(m) + p_2(m) + p_3(m) \leq c_1 q_1(m) + c_2 q_2(m) + c_3 q_3(m) \quad \forall m \geq \texttt{max}(m_1, m_2, m_3)$$

Let $c_{max} = \texttt{max}(c_1, c_2, c_3)$ then

$$q_1(m) + q_2(m) + q_3(m) \leq c_1 q_1(m) + c_2 q_2(m) + c_3 q_3(m) \leq c_{max}\big(q_1(m) + q_2(m) + q_3(m)\big)$$

If we set $m_T = \texttt{max}(m_1, m_2, m_3)$, we can say that there exist $a = c_{max}$ and $m \geq m_T$ such that

$$0 \leq q_1(m) + q_2(m) + q_3(m) \leq c_{max}\big(q_1(m) + q_2(m) + q_3(m)\big) \quad \forall m \geq m_T$$

The above equation satisfies the definition of $p(m) = O\big(q_1(m) + q_2(m) + q_3(m)\big)$.

(b) Yes. It holds for $\Omega()$ too.

Since $p_1(m) = \Omega(q_1(m))$, there exist positive constants c_1 and m_1, such that

$$0 \leq c_1 q_1(m) \leq p_1(m) \quad \forall m \geq m_1$$

Since $p_2(m) = \Omega(q_2(m))$, there exist positive constants c_2 and m_2, such that

$$0 \leq c_2 q_2(m) \leq p_2(m) \quad \forall m \geq m_2$$

Since $p_3(m) = \Omega(q_3(m))$, there exist positive constants c_3 and m_3, such that

$$0 \leq c_3 q_3(m) \leq p_3(m) \quad \forall m \geq m_3$$

If we integrate the inequality, the following equation holds.

$$0 \leq c_1 q_1(m) + c_2 q_2(m) + c_3 q_3(m) \leq p_1(m) + p_2(m) + p_3(m) \quad \forall m \geq \texttt{max}(m_1, m_2, m_3)$$

Let $c_{min} = \texttt{min}(c_1, c_2, c_3)$ then

$$c_{min}\big(q_1(m) + q_2(m) + q_3(m)\big) \leq c_1 q_1(m) + c_2 q_2(m) + c_3 q_3(m) \leq q_1(m) + q_2(m) + q_3(m)$$

If we set $m_T = \texttt{max}(m_1, m_2, m_3)$, we can say that there exist $a = c_{min}$ and $m \geq m_T$ such that

$$0 \leq c_{min}\big(q_1(m) + q_2(m) + q_3(m)\big) \leq q_1(m) + q_2(m) + q_3(m) \quad \forall m \geq m_T$$

The above equation satisfies the definition of $p(m) = \Omega\big(q_1(m) + q_2(m) + q_3(m)\big)$.

Problem 5)

a) We have for the upper bound:
   a. $m^2 \, log_2 m = \; O(m^2 \, log_2 m)$
   b. $10\left(\frac{2^m}{\log_2 m}\right) = \; O\left(\frac{2^m}{\log_2 m}\right)$
   c. $0.1(2^{m-5}) = O(2^m)$

   From problem 4 we can conclude $m^2 \, log_2 m + \; 10\left(\frac{2^m}{\log_2 m}\right) + \; 0.1(2^{m-5}) = O\left(m^2 \, log_2 m + \frac{2^m}{\log_2 m} + \; 2^m\right) = O(\, 2^m)$

b) We have for the lower bound:
   a. $m^2 \, log_2 m = \; \Omega(m^2 \, log_2 m)$
   b. $10\left(\frac{2^m}{\log_2 m}\right) = \; \Omega\left(\frac{2^m}{\log_2 m}\right)$
   c. $0.1(2^{m-5}) = \Omega(2^m)$

   From problem 4 we can conclude $m^2 \, log_2 m + \; 10\left(\frac{2^m}{\log_2 m}\right) + \; 0.1(2^{m-5}) = \Omega\left(m^2 \, log_2 m + \frac{2^m}{\log_2 m} + \; 2^m\right) = \Omega(\, 2^m)$