**CS446: Machine Learning** **Fall 2015**

# Problem Set 1

Jifu Zhao *Handed In: September* 14 2015

1. **Answer to problem 1**

   **a. Algorithm**

   Suppose that we have **m** training samples and there is **n** variables. And also, suppose that:

$$(x_1 = 1 \wedge x_5 = 0 \wedge x_7 = 1) \equiv (x_1 \wedge \neg x_5 \wedge x_7)$$

---

**Algorithm:** Finding accurate conjunctions

---

1: **temp** := $x_1 \wedge \neg x_1 \wedge x_2 \wedge \neg x_2 \wedge ..... \wedge x_{n-1} \wedge \neg x_{n-1} \wedge x_n \wedge \neg x_n$

2: **for** i := 1:m
    **if** *instance*($i$) is positive(+) **then**
        **remove** any item in temp that is inconsistent with *instance*($i$)
        (for example, if in *instance*($i$), $x_j = 1$, then remove $\neg x_j$ from temp, else if $x_j = 0$, then remove $x_j$)
    **end if**
**end for**

3: **for** i := 1:n
    **check** if there is $x_i \wedge \neg x_i$ exists
    **if** exists
        **remove** both $x_i$ and $\neg x_i$ from temp
    **end if**
**end for**

4: **for** i := 1:m
    **check** whether temp is consistent with *instance*($i$)
    **if** inconsistent
        **return:** Not found
        **stop**
    **end if**
**end for**

---

**Algorithm Explanation**:

According to **Find-S Algorithm** (Tom Mitchell, *Machinelearning*, 1997), in the proposed algorithm, we first generate the most specific condition: $x_1 \wedge \neg x_1 \wedge x_2 \wedge \neg x_2 \wedge ..... \wedge x_{n-1} \wedge \neg x_{n-1} \wedge x_n \wedge \neg x_n$, which is assigned to **temp**. Then, in **step 2**, compare this most specific case with the instances whose label is positive(+). In this procedure, we will remove any sub-term from **temp** that is inconsistent with the *instance*$(i)$. So, after step 2, in **temp**, only those that are consistent with all training samples will be left. In **step 3**, we consider the situation that both $x_i$ and $\neg x_i$ appear simultaneous in **temp**, then remove them from **temp**. Finally, in **step 4**, we compare the **temp** with all samples to make sure that our final conclusion is right.

**b. Proof**

1: In **step 4**, through comparing the proposed conjunction with all instances, we can make sure that if there is one conjunction generated through our algorithm, it will be consistent with all instances.

2: In more details, through **step 2** and **step 3**, we can make sure that the remaining temp will only keep the term that is consistent with all positive(+) instances.

3: Through **step 4**, we can make sure that our algorithm will be consistent with all negative(-) instances. Or there will be no result generated.

So, in conclusion, if there is one conjunction generated as the result, it will be consistent will all training samples.

**c. Complexity analysis**

1: In **step 2**, there will be **m** iteration. And in every iteration, we need to maximumly compare **n** terms, so the maximum comparison will be $m * n$.

2: In **step 3**, we will do the check for maximumly **n** times. So the maximum procedure will be $n$.

3: In **step 4**, there will be maximumly **m** procedure. So the maximum complexity will be $m$.

4: So, in conclusion, the final complexity will be $m * n + m + n$. It can also be represented by $O(mn)$.

**d. Ability analysis**

1: If the label got from the algorithm is **positive(+)**, it means that the correct label for this new example will be positive(+). The reason is clear. In our algorithm, all terms that is inconsistent with all the training examples will be removed. If we have more information, we will only remove more terms

from our final conjunction. But this will not affect the fact that the label for new example is positive(+). So, no matter how much more information is included, the label for new example will always be positive(+).

2: If the label for the new example is **negative(-)**, we cannot determine whether or not its real label is negative(-) or positive(+). The reason is also clear. When the label for the new example is negative(-), if we have more information, we will remove more terms from the conjunction we got. So, after removal, it is possible that we can get a positive(+) label for the new example. So, if the label for the new example is negative(-), we cannot determine easily. We need more information to get a more reliable conclusion.

2. **Answer to problem 2**

**a. Derivation**

In algebra, the hyperplane $\vec{w}^T\vec{x}+\theta=0$ means that the vector $\vec{w}$ is perpendicular to the hyperplane, and the $\theta$ means the shift from the hyperplane that goes across the zero point. So, in order to calculate the distance between the point $\vec{x_0}$ and the hyperplane $\vec{w}^T\vec{x}+\theta=0$, we first find one point on the hyperplane, which is $\vec{x}$, the vector formed by $\vec{x}$ and $\vec{x_0}$ will be $\vec{x_0}-\vec{x}$. Next, we only need to calculate the projection of $\vec{x_0}-\vec{x}$ on $\vec{w}$. It is defined as $\frac{|\vec{w}^T(\vec{x_0}-\vec{x})|}{||\vec{w}||}$. Since $\vec{w}^T\vec{x}+\theta=0$, so $\vec{w}^T\vec{x}=-\theta$. So $\frac{|\vec{w}^T(\vec{x_0}-\vec{x})|}{||\vec{w}||}$ will change into $\frac{|\vec{w}^T\vec{x_0}+\theta|}{||\vec{w}||}$. So the distance between $\vec{x_0}$ and the hyperplane $\vec{w}^T\vec{x}+\theta=0$ is:

$$d=\frac{|\vec{w}^T\vec{x_0}+\theta|}{||\vec{w}||}$$

**b. Distance between two hyper-planes**

The two hyperplanes, $\vec{w}^T\vec{x}+\theta_1=0$ and $\vec{w}^T\vec{x}+\theta_2=0$. They are parallel since they have the same $\vec{w}^T$. To calculate the distance between these two hyperplanes, we only need to find two point from each of these two hyperplanes, and calculate the projection of the vector formed by these two points on $\vec{w}$. Suppose we choose $\vec{(x_1)}$ from the first hyperplane, so $\vec{w}^T\vec{x_1}+\theta_1=0$. And choose $\vec{(x_2)}$ from the first hyperplane, so $\vec{w}^T\vec{x_2}+\theta_2=0$. The vector formed by $\vec{(x_1)}$ and $\vec{(x_2)}$ will be $\vec{(x_1-x_2)}$. So the projection of $\vec{(x_1-x_2)}$ on $\vec{w}$ will be $\frac{|\vec{w}^T(\vec{x_1}-\vec{x_2})|}{||\vec{w}||}$. Since $\vec{w}^T\vec{x_1}+\theta_1=0$ and $\vec{w}^T\vec{x_2}+\theta_2=0$, so $\vec{w}^T\vec{x_1}=-\theta_1$ and $\vec{w}^T\vec{x_2}=-\theta_2$. So the distance between these two hyperplanes will be:

$$d=\frac{|\theta_1-\theta_2|}{||\vec{w}||}$$

3. **Answer to problem 3**

## a. Linear separability

### a.1 Linear separability

**1, First, prove that if $D$ is linearly separable, then there exists a hyperplane that satisfies condition (3) with $\delta = 0$.**

If $D$ is linearly separable, then according to the definition,

$$y_i = \begin{cases} 1 & \text{if } \vec{w}^T\vec{x_i} + \theta \geq 0 \\ -1 & \text{if } \vec{w}^T\vec{x_i} + \theta < 0. \end{cases} \tag{1}$$

For the case that $\vec{w}^T\vec{x_i} + \theta \geq 0$, we can find an example $\vec{x_m}$ that is *closest* to the hyperplane among all positive examples. Set it to be $\vec{w}^T\vec{x_m} + \theta = A \geq 0$. Also, for the cases of $\vec{w}^T\vec{x_j} + \theta < 0$, we can also find a negative example $\vec{x_n}$ that is closest to the hyperplane among negative examples. Set it to be $\vec{w}^T\vec{x_n} + \theta = B < 0$. So, we have $A \geq 0 > B$.

Now, let's define $\alpha$ that satisfy $A - \alpha \geq 0 > B - \alpha$. So, now we have $\vec{w}^T\vec{x_i} + \theta - \alpha = A - \alpha \geq 0$ and $\vec{w}^T\vec{x_n} + \theta - \alpha = B - \alpha < 0$.

More specifically, let's find an $\alpha$ that makes the hyperplane $\vec{w}^T\vec{x} + \theta - \alpha = 0$ have the same distance for $x\vec{m}$ and $x\vec{n}$. So, we have

$$\frac{|\vec{w}^T\vec{x_m} + \theta - \alpha|}{||\vec{w}||} = \frac{|\vec{w}^T\vec{x_n} + \theta - \alpha|}{||\vec{w}||}.$$

So, we can get that $|\vec{w}^T\vec{x_m} + \theta - \alpha| = |\vec{w}^T\vec{x_n} + \theta - \alpha|$, then we have $\vec{w}^T\vec{x_m} + \theta - \alpha = -(\vec{w}^T\vec{x_n} + \theta - \alpha)$, which means that $A - \alpha = -B + \alpha$. So, $\alpha = \frac{A+B}{2}$. So, the hyperplane $\vec{w}^T\vec{x} + \theta - \alpha = 0$ will separate the set $D$.

So, when we choose $\vec{x_m}$ and $\vec{x_n}$, we have $\vec{w}^T\vec{x_m} + \theta - \alpha = \frac{A-B}{2} \geq 0$ and we also have $\vec{w}^T\vec{x_n} + \theta - \alpha = -\frac{A-B}{2} < 0$. If we multiply $\beta = \frac{2}{A-B}$ on both side, we can get $\beta \cdot \vec{w}^T\vec{x_m} + \frac{\theta-\alpha}{\beta} = 1$ and $\beta \cdot \vec{w}^T\vec{x_n} + \frac{\theta-\alpha}{\beta} = -1$.

Now, let $\vec{w'}^T = \beta \cdot \vec{w}^T$, and $\theta' = \frac{\theta-\alpha}{\beta}$, we will have that $y_i(\vec{w'}^T\vec{x_i} + \theta') = 1 - \delta$ to separate the set $D$, and $\delta = 0$.

So, from condition (1), we can get condition (3).

**2, Secondly, let's prove that from (3), we can get (1).**

From condition (3) $y_i(\vec{w}^T\vec{x_i} + \theta) \geq 1 - \delta$, when $\delta = 0$, we will have $y_i(\vec{w}^T\vec{x_i} + \theta) \geq 1$. It is very obvious that when $\vec{w}^T\vec{x_i} + \theta \geq 1$, we will have $\vec{w}^T\vec{x_i} + \theta \geq 0$. And when $\vec{w}^T\vec{x_i} + \theta < -1$, we will have $\vec{w}^T\vec{x_j} + \theta < 0$.

So, from condition(3), we can easily prove the correctness of condition(1).

**In conclusion**, the data set $D$ is linearly separable if and only if there exists a hyperplane that satisfies condition(3) with $\delta = 0$.

**3, If there exists a hyperplane that satisfies condition(3) with $\delta > 0$, our conclusion will be determined by the value of $\delta$.**

If $\delta < 1$, we can conclude that $y_i(\vec{w}^T\vec{x_i} + \theta) > 0$, which is consistent with the condition(1). So, **when $0 < \delta < 1$, we can conclude that the set $D$ is separable**.

But when $\delta \geq 1$, we will have $y_i(\vec{w}^T\vec{x_i} + \theta) \geq 1 - \delta$. However, we can only know that $1 - \delta \leq 0$, we cannot determine whether or not $y_i(\vec{w}^T\vec{x_i} + \theta) > 0$. So, **when $\delta \geq 1$, we cannot determine whether or not the set $D$ is separable**.

### a.2 Trivial optimal solution

When $y_i(\vec{w}^T\vec{x_i} + \theta) \geq -\delta$, if $\delta \geq 0$, the optimal solution for $\delta$ will be $\delta = 0$. When $\delta = 0$, we will have $y_i(\vec{w}^T\vec{x_i} + \theta) \geq 0$, so the optimal solution for this will be: $\vec{w} = \vec{0}$ and $\theta = 0$. This solution have no meaning for this question.

So, if we use the format that $y_i(\vec{w}^T\vec{x_i} + \theta) \geq -\delta$, we can only get the optimal solution that $\delta = 0$, $\vec{w} = \vec{0}$ and $\theta = 0$, which have no meaning for us. But if we choose the format in (2) to (4), our optimal solution will not be $\vec{w} = \vec{0}$ and $\theta = 0$. So, (2) to (4) will help us actually solve this problem.

### a.3 Optimal solutions

When $\vec{x_1}^T = \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix}$, $\vec{x_2}^T = \begin{bmatrix} -1 & -1 & \dots & -1 \end{bmatrix}$ and $y_1 = 1$, $y_2 = -1$. Apply (3) to $\vec{x_1}$, $\vec{x_2}$, $y_1$ and $y_2$. We can get that $w_1 + w_2 + \dots w_n + \theta \geq 1 - \delta$ and $w_1 + w_2 + \dots w_n - \theta \geq 1 - \delta$. Considering the condition that $\delta = 0$, we will have that $w_1 + w_2 + \dots w_n \geq 1 - \theta$ and $w_1 + w_2 + \dots w_n \geq 1 + \theta$.

So, the final conclusion will be: $w_1 + w_2 + \dots w_n \geq 1 - |\theta|$.

## b. Linear Programming

### b.1 Rewrite the linear program

from (2) to (4), we can get

$$\min_{\delta,\vec{w},\theta} \quad \delta \tag{2}$$

$$\text{subject to} \quad y_i\vec{w}^T\vec{x_i} + y_i\theta \geq 1 - \delta \tag{3}$$

$$\delta \geq 0 \tag{4}$$

$$\min_{\delta,\vec{w},\theta} \quad \delta \tag{5}$$

$$\text{subject to} \quad y_i\vec{w}^T\vec{x_i} + y_i\theta + \delta \geq 1 \tag{6}$$

$$\delta \geq 0 \tag{7}$$

Write (5) to (7) into matrix, we can get:

$$z(\vec{t}) = \delta = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \vec{w} \\ \theta \\ \delta \end{bmatrix} \tag{8}$$

$$\text{subject to} \quad \begin{bmatrix} y_1\vec{x_1}^T & y_1 & 1 \\ y_2\vec{x_2}^T & y_2 & 1 \\ \dots & \dots & \dots \\ y_m\vec{x_m}^T & y_m & 1 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \vec{w} \\ \theta \\ \delta \end{bmatrix} \geq \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \\ 0 \end{bmatrix} \tag{9}$$

So, choose $\vec{c}^T = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$, $\vec{t} = \begin{bmatrix} \vec{w} \\ \theta \\ \delta \end{bmatrix}$, $A = \begin{bmatrix} y_1\vec{x_1}^T & y_1 & 1 \\ y_2\vec{x_2}^T & y_2 & 1 \\ \dots & \dots & \dots \\ y_m\vec{x_m}^T & y_m & 1 \\ 0 & 0 & 1 \end{bmatrix}$ and $b = \begin{bmatrix} 1 \\ 1 \\ \dots \\ 1 \\ 0 \end{bmatrix}$.

So, the question turns to be a linear question, which is

$$z(\vec{t}) = \vec{c}^T\vec{t}$$

$$\text{subject to} \quad A\vec{t} \geq \vec{b}$$

We can solve this problem through linear programming.

The Matlab code for **findLinearDiscriminant.m** is shown as follows.

---

```
function [w, theta, delta] = findLinearDiscriminant(data)
%% setup linear program
[m, np1] = size(data);
n = np1-1;

% write your code here

A = zeros(m+1, n+2);
for j = 1:m
    A(j, 1:n) = data(j, n+1).*data(j, 1:n);
    A(j, n+1) = data(j, n+1);
```

```
    A(j, n+2) = 1;
end
A(m+1, n+2) = 1;

b = zeros(m+1, 1);
b(1:m, 1) = 1;

c = zeros(n+2, 1);
c(n+2, 1) = 1;

%% solve the linear program
%adjust for matlab input: A*x <= b
[t, z] = linprog(c, -A, -b);

%% obtain w,theta,delta from t vector
w = t(1:n);
theta = t(n+1);
delta = t(n+2);

end
```

### b.2 Learning Conjunctions as an LP

The manually generated data set in **hw1sample2d.txt** is as follows:

$$\begin{array}{ccc} 1 & 1 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \\ 0 & 0 & -1 \end{array}$$

The Matlab code for **plot2dSeparator.m** is shown as follows.

```
function plot2dSeparator(w, theta)

n = length(w);
if n ~= 2
    disp('only 2d data supported.')
else
    x = -0.1:0.01:1.1;
    y = -w(1) * x / w(2) - theta / w(2);
    plot(x, y, 'LineWidth',2);
end
```

The output of **plot2dSeparator.m** and **plot2dData.m** is shown in figure 1. The red points means the label is positive(+) and the green points means that the label is negative(-). The blue line successfully separate those two labels data.

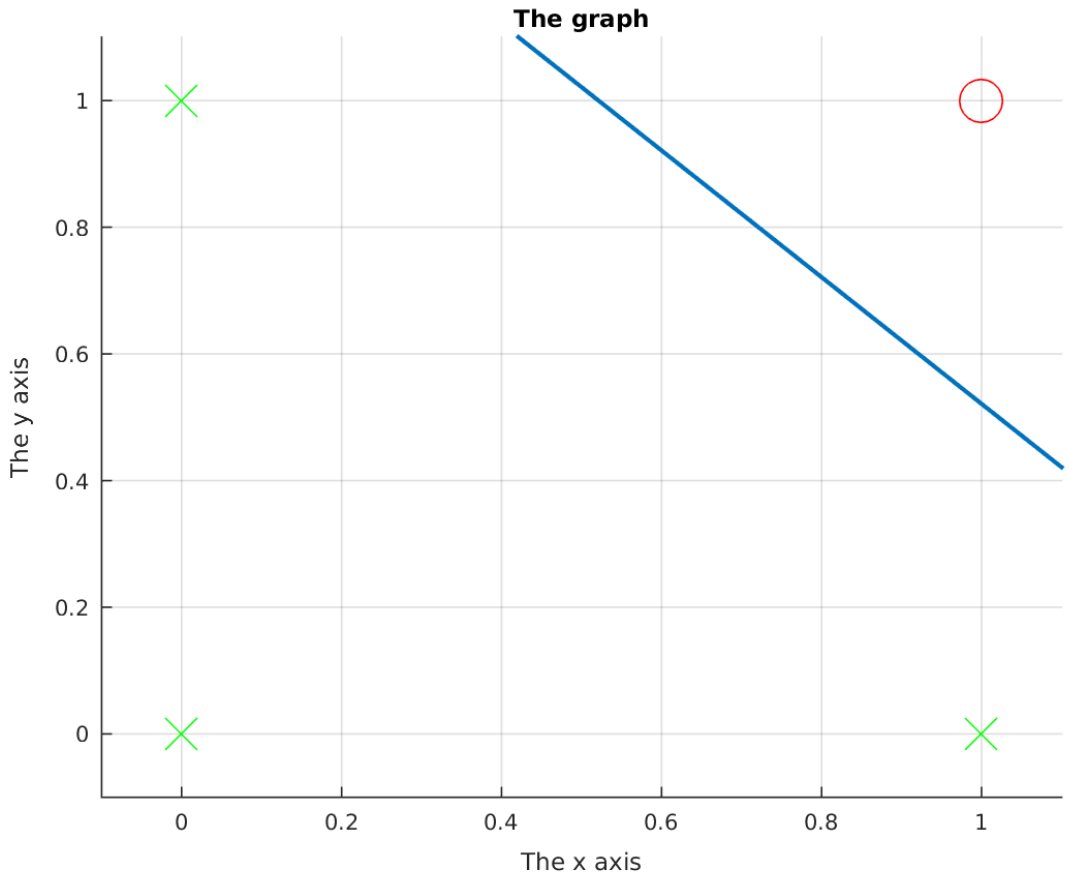


Figure 1: 2D data and separator

The output for **hw1conjunction.txt** are show below:

$$\begin{bmatrix} w_1 \\ w2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \\ w_9 \\ w_{10} \end{bmatrix} = \begin{bmatrix} 2.910374 \\ -2.049762 \\ 0.177469 \\ 190.519580 \\ 0.139878 \\ -3.100985 \\ -2.953437 \\ -193.277767 \\ 1.167917 \\ -8.894156 \end{bmatrix}$$

And the corresponding $\theta$ and $\delta$ are:

$$\theta = -90.211531 \text{ and } \delta = -0.000000$$

---

And the expression for final discriminant function are:

$$\vec{w}^T\vec{x} - 90.211531 = 1$$

Which can be written as:

$$\vec{w}^T\vec{x} - 91.211531 = 0$$

Corresponding to the output, we can get the expression for the conjunction:

$$\vec{x_4} \wedge \neg\vec{x_8}$$

The value of $\delta$ is very close to 0, which is consistent with our previous theory. This means that the data set can be linearly separated.

### b.3 Learning Badges

The Matlab code for **computeLabel.m** is shown as follows.

---

```
function y = computeLabel(x, w, theta)

y0 = w' * x + theta;

if y0 >= 0
    y = 1;
else
    y = -1;
end

end
```

Based on this code, run the learnBadges.m, we can get the result shown in figure 2.

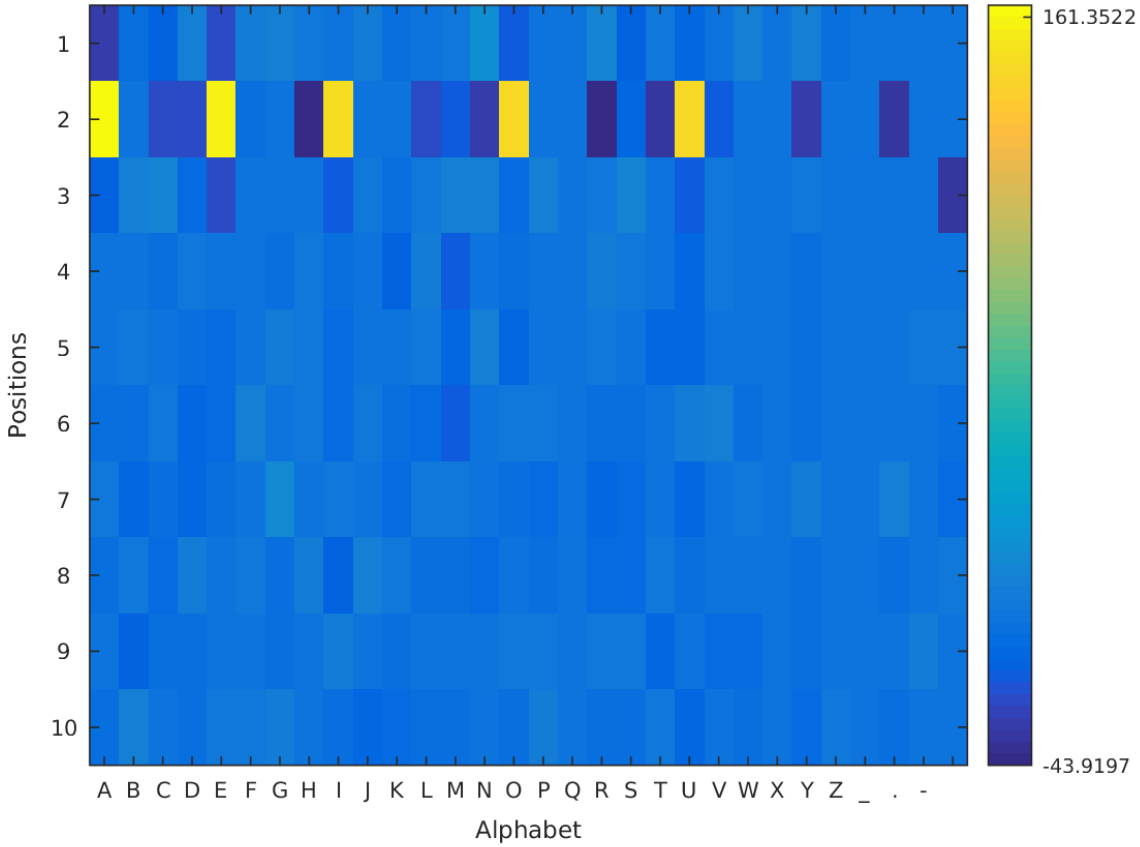


Figure 2: Result for Badges game 1

The corresponding value are:

$$\delta = 1.2612e - 19$$
$$accuracyInTrain = 1$$
$$accuracyInTest = 1$$

Now, change the value for **alphabet** and **positions** to get other results. After some changes, I find that when:

$$alphabet =' ABCDEFGHIJKLMNOPQRSTU'$$
$$positions = 3 : 10$$

The output will be:

$$\delta = 2.2524e - 11$$
$$accuracyInTrain = 1$$
$$accuracyInTest = 0.7234$$

And the corresponding figure is shown in figure 3.

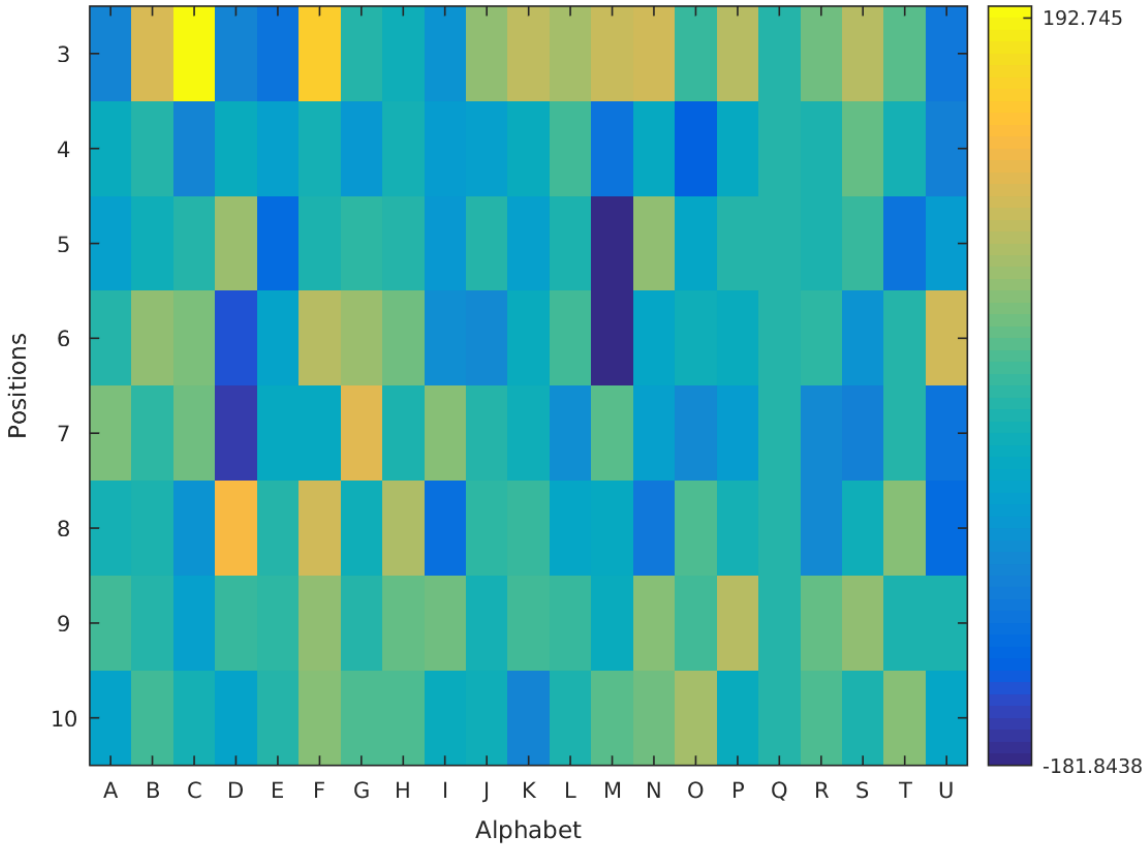


Figure 3: Result for Badges game 2

Compared with figure 2, it is easy to find that in figure 3, the relation of alphabet and position is not very clear. But in figure 2, the relation of alphabet and position is very clear.

### b.4 Learning Multiple Hyperplanes

When $\vec{w}$ is known, the question can be simplified.

from (2) to (4), we can get

$$\min_{\delta,\vec{w},\theta} \quad \delta \tag{10}$$

$$\text{subject to} \quad y_i\theta + \delta \geq 1 - y_i\vec{w}^T\vec{x_i} \tag{11}$$

$$\delta \geq 0 \tag{12}$$

Write (10) to (12) into matrix, we can get:

$$z(\vec{t}) = \delta = \begin{bmatrix}0 & 1\end{bmatrix} * \begin{bmatrix}\theta \\ \delta\end{bmatrix} \tag{13}$$

$$\text{subject to} \quad \begin{bmatrix} y_1 & 1 \\ y_2 & 1 \\ \dots & \dots \\ y_m & 1 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix}\theta \\ \delta\end{bmatrix} \geq \begin{bmatrix} 1 - y_1\vec{w}^t\vec{x_1} \\ 1 - y_1\vec{w}^t\vec{x_2} \\ \dots \\ 1 - y_1\vec{w}^t\vec{x_m} \\ 0 \end{bmatrix} \tag{14}$$

So, choose $\vec{c}^T = \begin{bmatrix}0 & 1\end{bmatrix}$, $\vec{t} = \begin{bmatrix}\theta \\ \delta\end{bmatrix}$, $A = \begin{bmatrix} y_1 & 1 \\ y_2 & 1 \\ \dots & \dots \\ y_m & 1 \\ 0 & 1 \end{bmatrix}$ and $b = \begin{bmatrix} 1 - y_1\vec{w}^t\vec{x_1} \\ 1 - y_1\vec{w}^t\vec{x_2} \\ \dots \\ 1 - y_1\vec{w}^t\vec{x_m} \\ 0 \end{bmatrix}$.

So, the question turns to be a linear question, which is

$$z(\vec{t}) = \vec{c}^{\,T}\vec{t}$$

$$\text{subject to} \quad A\vec{t} \geq \vec{b}$$

We can solve this problem through linear programming.

The Matlab code for **findLinearThreshold.m** is shown below:

---

```
function [theta,delta] = findLinearThreshold(data,w)
%% setup linear program
[m, np1] = size(data);
n = np1 - 1;

% write your code here

A = zeros(m+1 ,n);
A(1:m, 1) = data(1:m, np1);
A(m+1,1) = 0;
A(:, 2) = 1;
```

```
b = zeros(m+1, 1);
b(1:m, 1) = data(1:m, 1:n) * w;
for i = 1:m
    b(i) = 1 - data(i, np1) * b(i);
end
b(m+1, 1) = 0;

c = [0; 1];

%% solve the linear program
%adjust for matlab input: A*x <= b

[t, z] = linprog(c, -A, -b);

%% obtain w,theta,delta from t vector

theta = t(1);
delta = t(2);

end
```

---

The plots for different separators are shown in figure 4.

Through calculation, the weight is $\vec{w}^T = [170.009960, 294.223222]$ and the corresponding values are $\theta = -243.262997$ and $\delta = 0.000000$.

Using $\vec{w}^T = [100.0, 300.0]$, the corresponding values are $\theta = -218.994588$ and $\delta = 0.000000$.

Using $\vec{w}^T = [130.0, 100.0]$, the corresponding values are $\theta = -123.611614$ and $\delta = 0.000000$.

Using $\vec{w}^T = [500.0, 100.0]$, the corresponding values are $\theta = -344.335000$ and $\delta = 92.965000$.

Compared those 4 plots(on clockwise direction, the plots are 1, 2, 3, and 4), it is easy to find that line 1(red), line 2(blue) and line 3(brown) all successfully separate the data set. They should all be seen as right solution. But the line 4(purple) is not a good one. For line 1, 2 and 3, their $\delta$ are 0.000000, 0.0000000 and 0.000000. For line 4, its $\delta$ is 92.965000. When the value of $\delta$ is too far from 0, the hyperplane got is not very accurate.

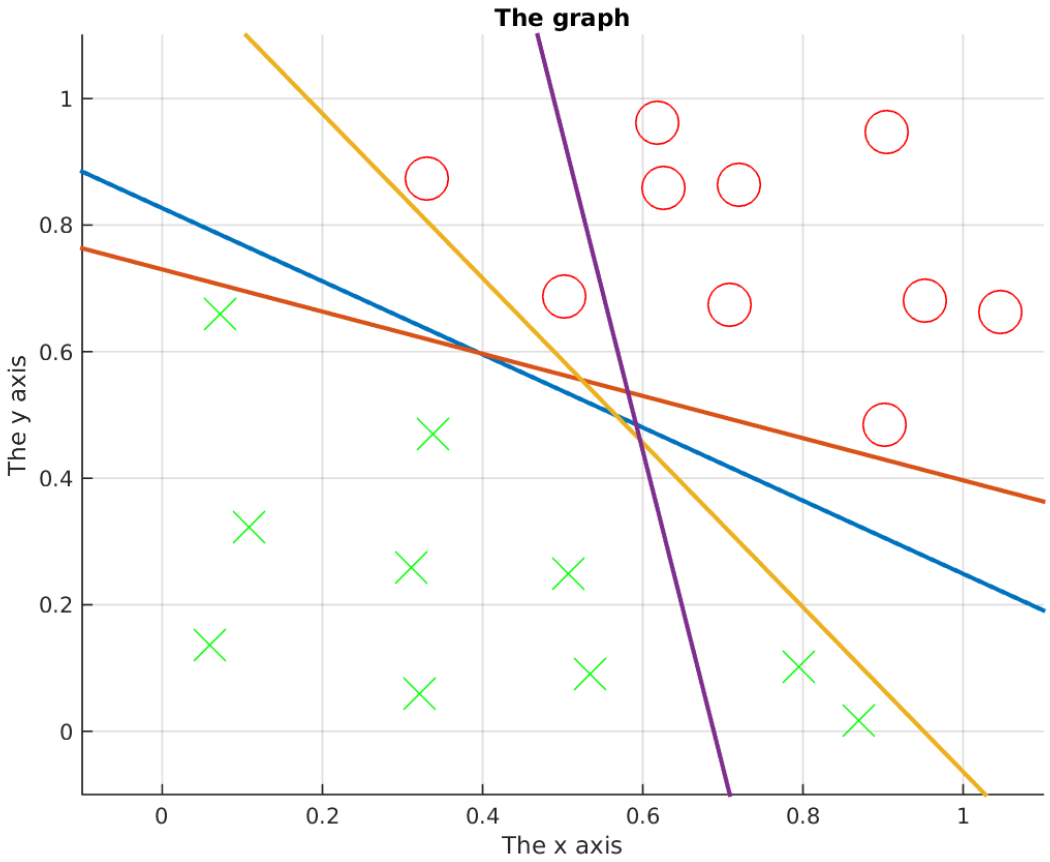


Figure 4: Multiple hyperplane separator

For line 1, 2 and 3, although they are all right, line 2(blue) seems to have the smallest average distance from all points. From this point, it is a better solution.

The solution to this question is clear not unique. But when we try to minimum the $\delta$, there should be only one optimal solution. But that solution is only for the condition that minimum $\delta$.