Solution Report

ML- CS 5780

Time of submission 01:15 PM 8th October '13

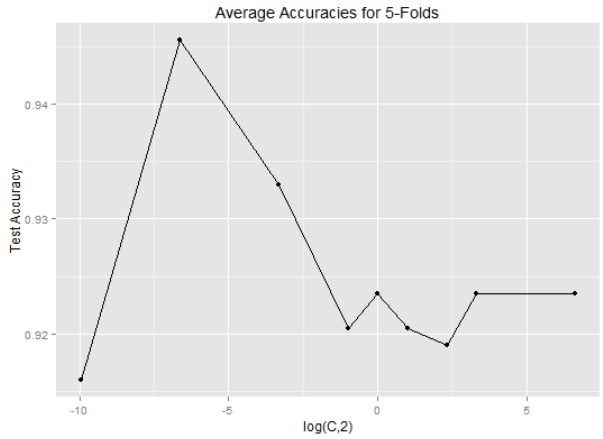
Assignment 3

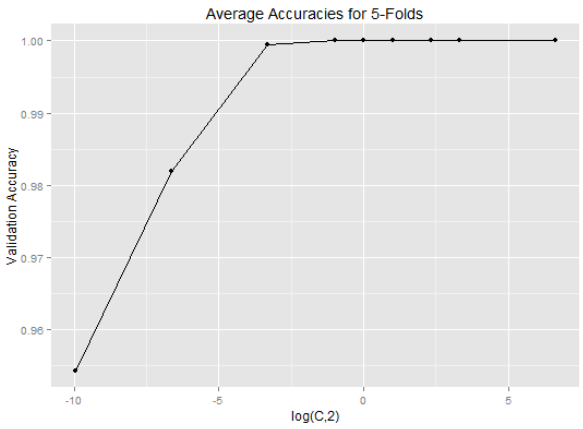
Net IDs- FT98 and KC647

Solution to Question 2

a)

The graphs are:





The Table containing these values is-

C	Log(C,2)	Avg Train Acc.	Avg Validation Acc.
0.001	-9.96578	0.95426	0.916
0.01	-6.64386	0.982	0.9455
0.1	-3.32193	0.99952	0.933
0.5	-1	1	0.9205
1	0	1	0.9235
2	1	1	0.9205
5	2.321928	1	0.919
10	3.321928	1	0.9235
100	6.643856	1	0.9235

The value of C for which we obtain maximum accuracy is C=0.01.

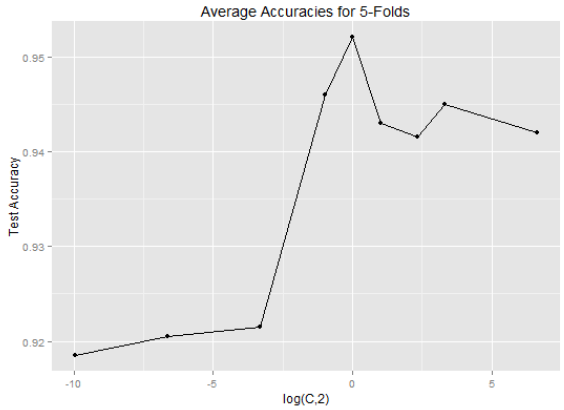
b) After running for C=0.01, we obtain following accuracies-

For Test Dataset= 93.10%

For Train Dataset= 100%

c)

After normalizing the results, we obtain the following results-



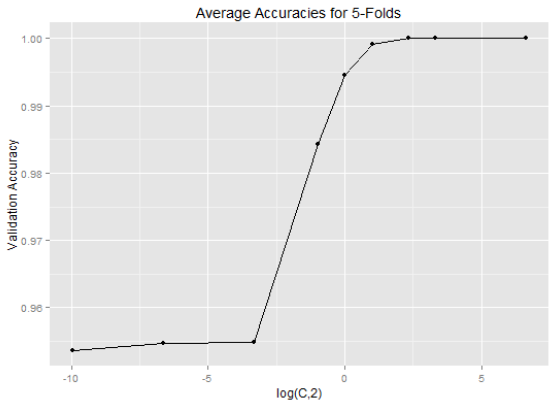


Table containing values of the plots are-

C	Log(C,2)	Avg Train Acc	Avg Validation Acc
0.001	-9.96578	0.953625	0.9185
0.01	-6.64386	0.95475	0.9205
0.1	-3.32193	0.954875	0.9215
0.5	-1	0.98425	0.946
1	0	0.9945	0.952
2	1	0.999125	0.943
5	2.321928	1	0.9415
10	3.321928	1	0.945
100	6.643856	1	0.942

Maximum accuracy over validation set is obtained for c=1

Train accuracy is 99.50% for complete data for c=1.

Test accuracy is 95.80% for complete data for c=1.

d) Test accuracy increases with normalization to 95% from 93.10% when the data was normalized.

This is because in our case of simple Euclidean distance is the classifying criterion that gives some of the features more importance than the others if the data is not normalized.

Thus for small features, even small change in the feature values can bring significant unrequired change in the model when the data is not normalized.

After normalization each of the points fall within the same ranges, hence no differential weightage even for small changes in values.

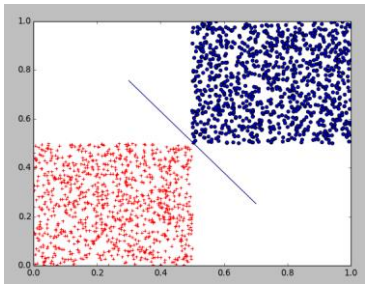
We have to re-estimate the optimal value of C, as it is the tuning parameter. Different C's gives different results over different type of datasets. This is very data specific. So, we can find the optimum value of C that divides the classes perfectly. We can find the data properties by checking out accuracies on the validation dataset. Size of the margin can be very small for skewed datasets, to give best accuracies. Once, we find the data properties by finding the optimum value of C, the model will provide better simulation of the real world simulation from which the data was generated.

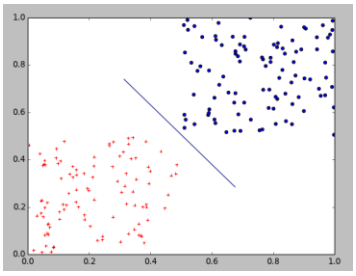
Solution to 3

a) The concept of class weights can be attached based on the ratio of number of negative is to number of positive examples.
Since P(Y=1) = 0.07, the number of positive class is very less, we can achieve an accuracy of more than 90% if we classify everything as negative. This we can have accuracy of around 93%. This can be achieved with extremely high value of j in the svmlearn. But the major problem with this rule is that it is highly insensitive to the positive classes and will classify it as negative majority of the times.

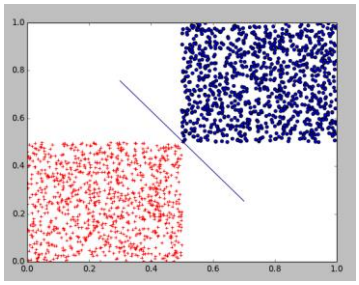
b) The graph of resulting dataset on both train and test as follows-

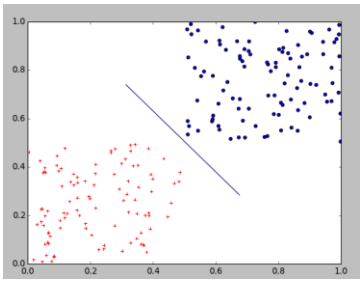
- For C=1 and nneg= 100

Test: 

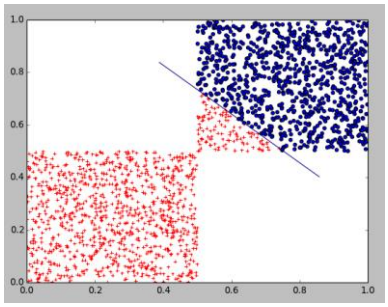
Train: 

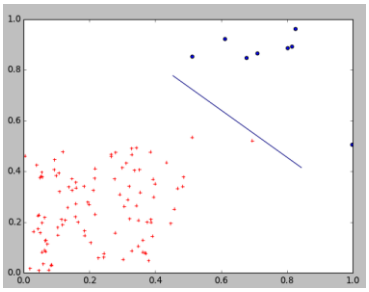
- For C=1 and nneg=50

Test: 

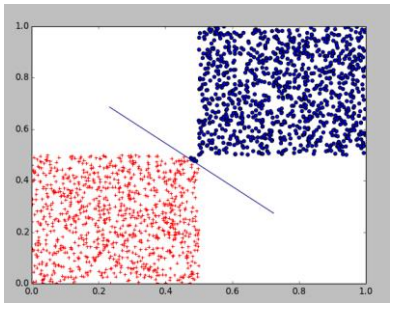
Train: 

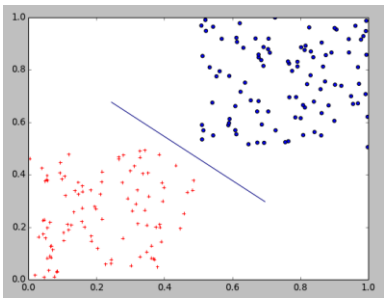
- For C=1 and nneg=10

Test: 

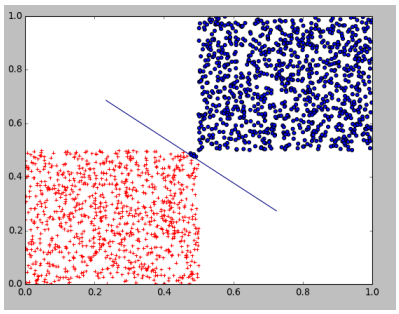
Train: 

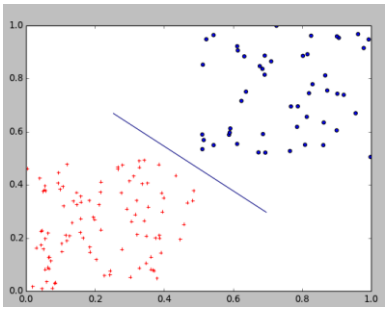
- For C=1000 and nneg=100

Test: 

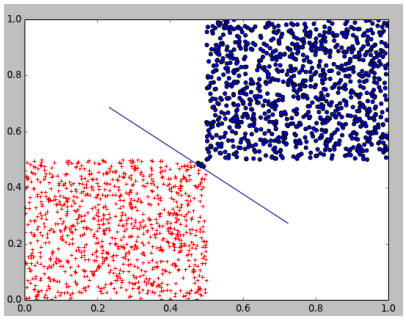
Train: 

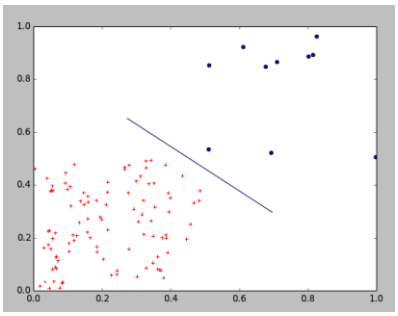
- For C=1000 and nneg=50

Test: 

Train: 

- For C=1000 and nneg=10

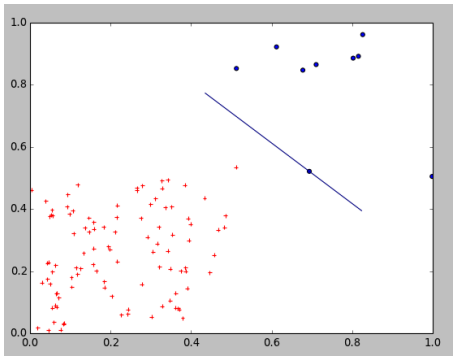
Test: 

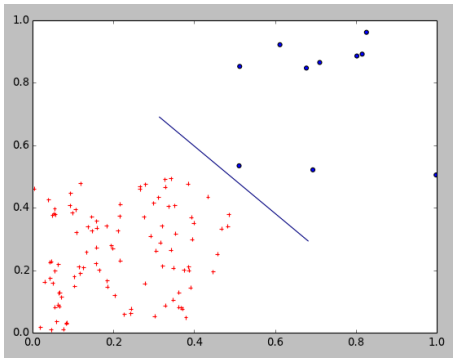
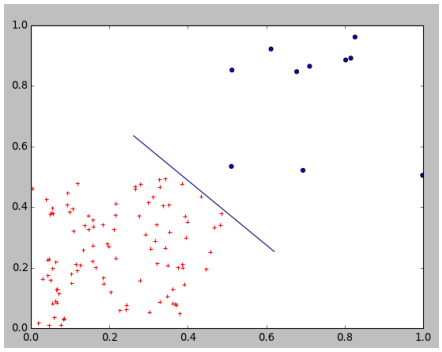
Train: 

c) For C=1, the location of hyperplane is sensitive to position of the data points (the negative data points location) as value of c (margin) is low. Therefore when the class size gets skewed, hyperplane shifts towards negative points.
For C=1000, the location of hyperplane is not that sensitive to position of the data points as value of c (margin) is very high. So, hyperplane position is typically consistent inspite of the class size getting skewed.
Soft margin classifier decreases the penalty on the some of the erroneous points when they fall within the margin range. It allows for the possibility of violating the inequality constraints..

d) Quantitative argument:
The j is a more accurate measure as it counts in for the weight to be given to each class independent of the class type. If the class size is skewed, and there is no other information provided, then it makes intuitive sense for differential penalty for each of the class. The class which has more number of instances will be penalized less when its instances cross over the margin. Whereas the class which has lesser number of instances will be penalized more when it data points try to cross over the margin. This penalization can be effectively done as the ratio of number of instances in either of the classes. In the end our optimization objective is to find the best fitting support vectors that separates out the classes in such a way that it maximizes the margin.

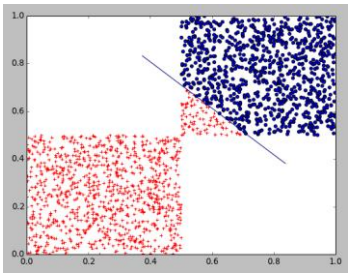
e) For S_10 and C=1, the value of each of the plots are given as below-

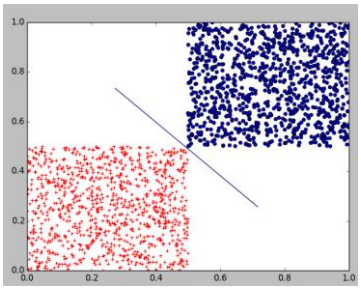
- For j=0.5 

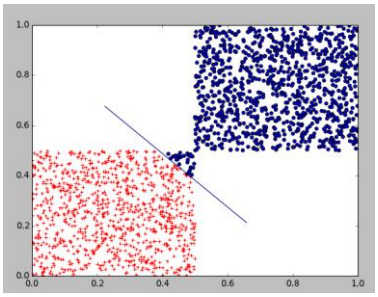
- For j=0.1 
- For j=0.05 

As we decrease the value of j, the hyperplane gets displaced and start shifting towards positive class from the negative class. Since j defined as the cost-factor, by which errors on positive instances outweight errors on negative instances, when we reduce the value of j errors on positive instances have lesser weight in the main SVM running function. This happens because differential penalty we apply on negative class due to change in j (i.e. the class size). When j decreases, the instances from the negative class cannot cross-over easily.

f) Overall accuracy in case of models from part 3e) when run on test dataset is as follows:

- For j=0.5, test data plot is 
  The accuracy of the model is – 95.6 %
  Number of False Positive is 87 and number of False Negative is 0.

- For j=0.1, test data plot is 
  The accuracy of the model is – 100%.
  Number of False Positive is 0 and number of False Negative is 0.

- For j=0.05, test data plot 
  The accuracy of the model is – 98.55 %.
  Number of False Positive is 0 and number of False Negative is 29.

When j is decreased, the number of False positive decreases to zero, while the number of False negative increases. This is because the when j is less, the instances of negative class are penalized strongly when they try to cross over the margin.

Solution to Question 1

Answer 1)

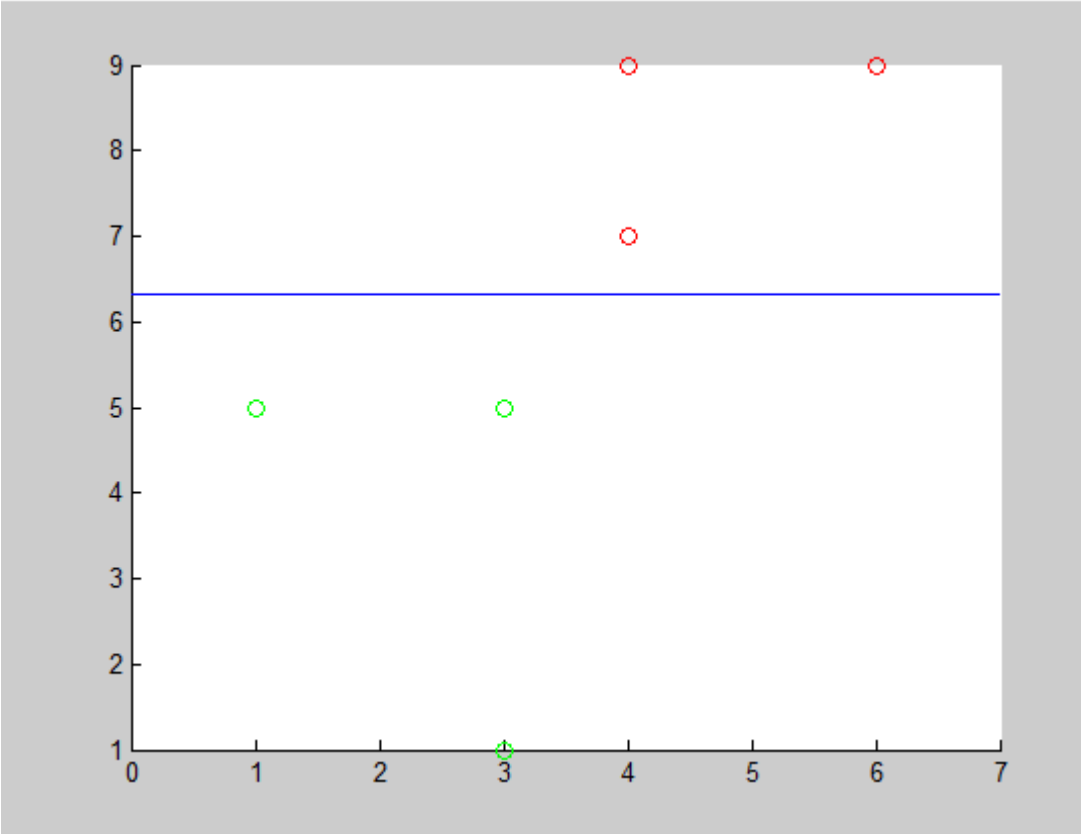
W_{biased}=<0, 3,-19>

No, this resulting hyper-plane does not maximise the margin between 2 classes does not.

The hyperplane equation is **3*y-19=0**

Its margin is 2/3 units from point (7,4)

The plot can be showed as follows:



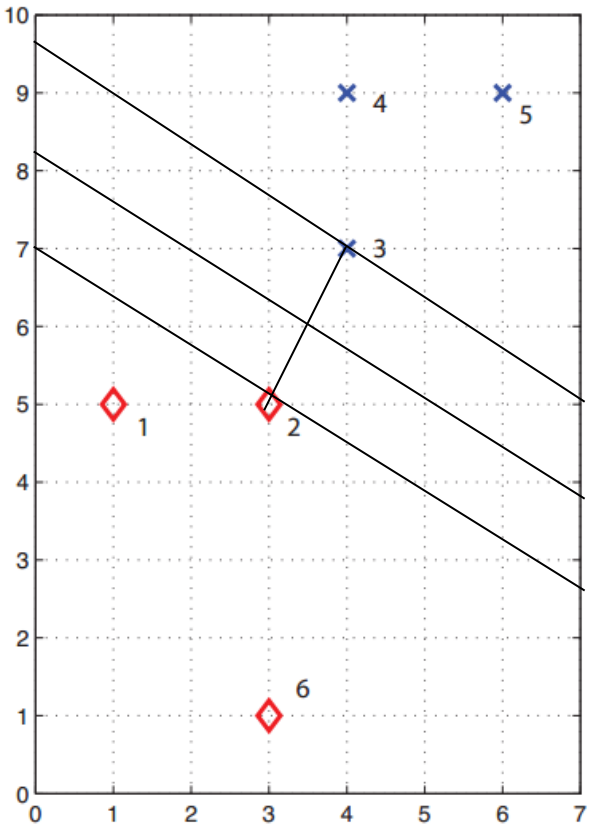
The code in order to generate this plot is :

```
x=[ 1,5,1;
    3,5,1;
    4,7,1;
    4,9,1;
    6,9,1;
    3,1,1
   ];
y=[-1;-1;1;1;1;-1];
w=zeros(50,3);
k=1;
flag=0;

while(flag==0)
    j=1;
    for i=1:6
        if(y(i)*(sum(w(k,:).*x(i,:)))<=0)
            j=j+1;
            w(k+1,:)=w(k,:)+y(i)*x(i,:);
            k=k+1;
        end
    end
    if(j==1)
        break;
    end

end
Yaxis=1:10;
scatter(x(:,1),x(:,2));
Xaxis=1:10;
Yaxis=(w(k,1)*Xaxis+w(k,3))/w(k,2);
plot(Xaxis,Yaxis);
```

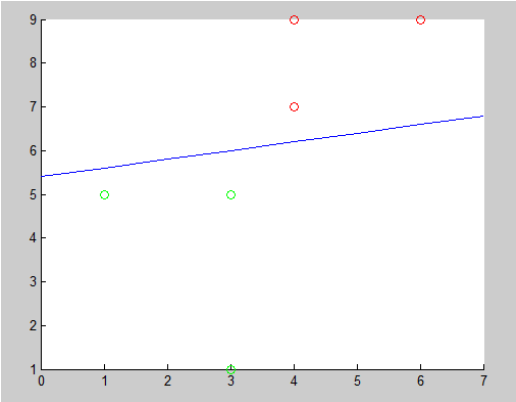
1)b)



1)c)

β	w_{opt}, b_{opt}			M
0.5	-1	5	-27	80

The plot is as follows:

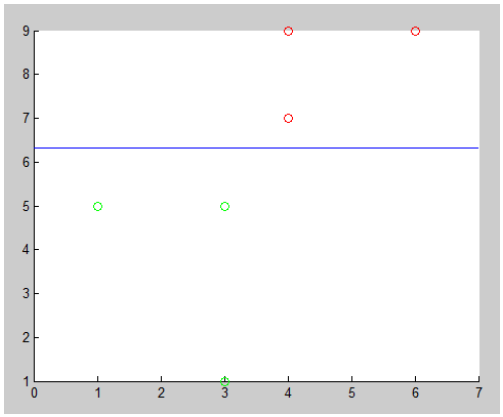


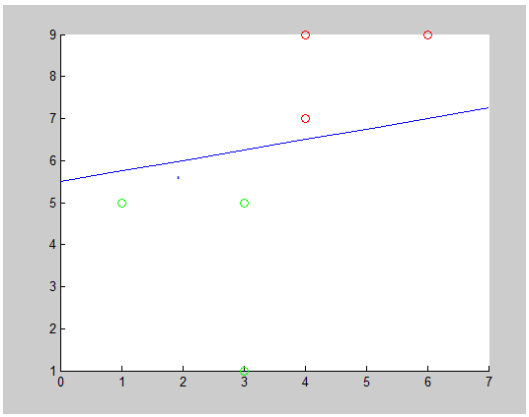
We observe that total number of mistakes which happen before the algorithm reaches convergence increases.

1)d)

β	w_{opt}, b_{opt}			M
0.1	0	3	-19	54
0.2	0	3	-19	54
0.3	0	3	-19	54
0.4	-1	4	-22	65
0.5	-1	5	-27	80
0.6	4	4	-36	109
0.7	1	5	-35	104
0.8	2	5	-37	112
0.9	2	5	-37	112
0.95	2	5	-37	112

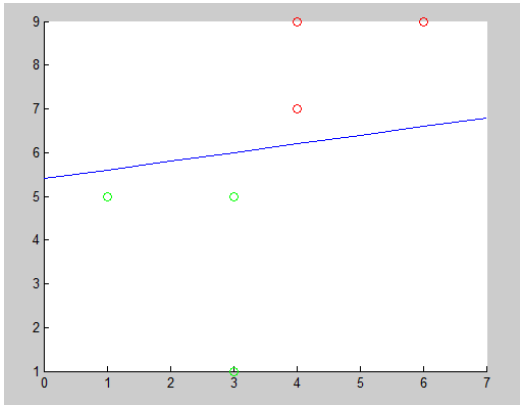
β=0.1,0.2,0.3 β=0.4

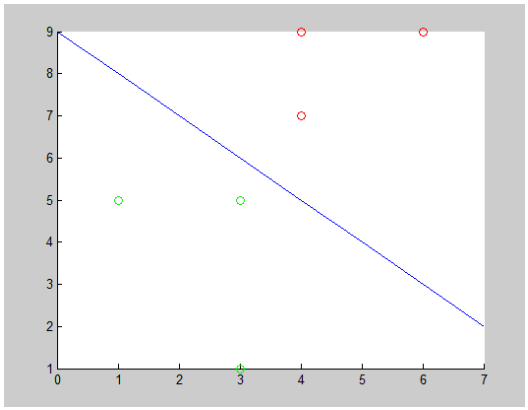




β =0.5,

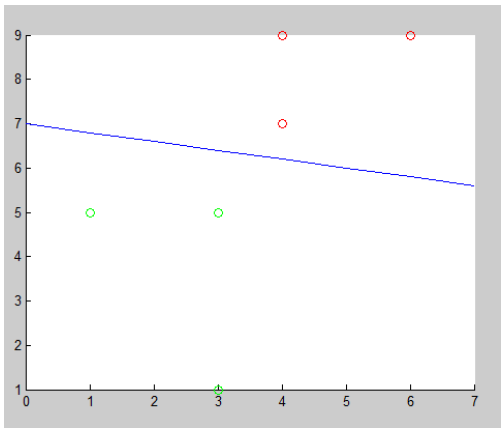
β= 0.6,

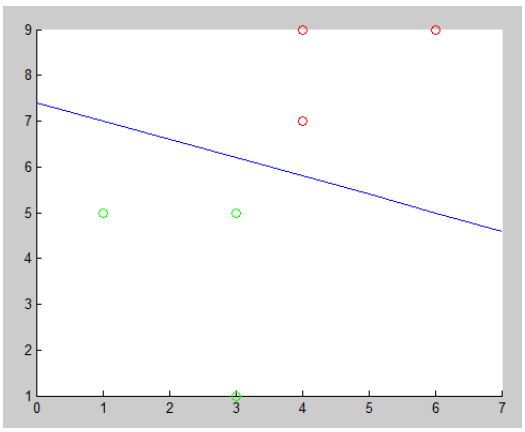




β =0.7,

β =0.8,0.9, 0.95





The code is:

```
x=[ 1,5,1;
    3,5,1;
    4,7,1;
    4,9,1;
    6,9,1;
    3,1,1
   ];
y=[-1;-1;1;1;1;-1];
w=zeros(50,3);
normk=zeros(50,1);
normk(1,:)=1;
k=1;
flag=0;

gama= 5/sqrt(20);
beta=0.5;
BetaGama= gama*beta;
for m=1:100
    j=1;
    for i=1:6
        Check(k)=y(i)*(sum(w(k,:).*x(i,:)))/normk(k);
        if(y(i)*(sum(w(k,:).*x(i,:))/normk(k))<=BetaGama)
            j=j+1;
            w(k+1,:)=w(k,:)+y(i)*x(i,:);
            normk(k+1)=sqrt(sum((w(k+1,1:2)).^2,2));
            k=k+1;
        end
    end
    if(j==1)
        break;
    end

end

Yaxis=1:10;
scatter(x([3,4,5],1),x([3,4,5],2),'r');
hold on;
scatter(x([1,2,6],1),x([1,2,6],2),'g');
hold on;
Xaxis=0:7;
Yaxis=(-w(k,1)*Xaxis-w(k,3))/w(k,2);
plot(Xaxis,Yaxis);
```

e) We observe that as the as we increase the value of β the number of mistakes which can be incurred increases. This is because with higher β the chances of a test example to lie with the same region increases but this also helps in maximising the marging of the optimal hyperplane that we achieve eventually . This can be easily observed by seeing that

β	w_{opt}, b_{opt}			Margin
0.1	0	3	-19	0.66

0.95	2	5	-37	6/sqrt(29)=1.11

1) (i) In this modified perceptron algorithm we predict the mistake bound using

$$y_i \frac{(w^T x_i + b)}{\|w\|} < \frac{\gamma}{2}.$$

- $\|w_{t+1}\|^2 \leq \|w_t\|^2 + 1$ (given).

Adding $\|w_t\|^2 - 2|w_{t+1}||w_t|$ on both sides.

$$\rightarrow \|w_{t+1}\|^2 + \|w_t\|^2 - 2\|w_{t+1}\|\|w_t\| \leq 1 + 2\|w_t\|^2 - 2|w_{t+1}||w_t|$$

$$\Rightarrow (|w_{t+1}| - |w_t|)^2 \leq 1 + 2|w_t|^2 - 2|w_{t+1}||w_t|$$

Now $(|w_{t+1}| - |w_t|)^2 \geq 0$.

$$\Rightarrow 0 < 1 + 2|w_t|^2 - 2|w_{t+1}||w_t|$$

$$\Rightarrow 2|w_t||w_{t+1}| \leq 1 + 2|w_t|^2$$

Dividing both sides by $2|w_t|$

$$\Rightarrow |w_{t+1}| \leq \frac{1 + 2|w_t|^2}{2|w_t|}$$

$$\rightarrow \boxed{|w_{t+1}| \leq \frac{1}{2|w_t|} + |w_t|}$$

Hence proved.

To prove:

$$|w_{t+1}| < |w_t| + \frac{1}{2|w_t|} + \frac{\gamma}{2}.$$

Proof

$$(w_{t+1})^2 = (w_t + y_i x_i)^2$$

$$= |w_t|^2 + 2 w_t y_i x_i + (y_i)^2 |x_i|^2$$

Given $\rightarrow (y_i)^2 = 1$ as $y_i = (-1, 1)$

$\rightarrow x_i . x_i = |x_i|^2$

$\rightarrow y_i (w . x_i) \leq \frac{\gamma}{2}$ (as it is a mistake)

$$\therefore |w_{t+1}|^2 < |w_t|^2 + \frac{\gamma}{2} + 1$$

$$\Rightarrow |w_{t+1}|^2 - |w_t|^2 < \frac{\gamma}{2} + 1.$$

Adding $-2 w_{k+1} . w_k + 2 |w_k|^2$

$$\Rightarrow |w_{t+1}|^2 - |w_t|^2 - 2 w_{k+1} . w_k + 2 |w_k|^2 < \frac{\gamma}{2} + 1 - 2 w_{k+1} . w_k + 2 |w_k|^2$$

$$\Rightarrow (|w_{t+1}| - |w_t|)^2 < \frac{\gamma}{2} + 1 - 2 w_{k+1} . w_k + 2 |w_k|$$

Since $(|w_{t+1}| - |w_t|)^2 \geq 0$

$$\Rightarrow \quad 0 \leq \frac{\gamma}{2} + 1 - 2\, w_{k+1}\, w_k + 2|w_k|^2$$

$$\Rightarrow \quad 2\, w_{k+1}\, w_k \leq \frac{\gamma}{2} + 1 + 2|w_k|^2$$

Dividing both sides by $2|w_k|$.

$$w_{k+1} \leq \frac{\gamma}{2} + \frac{1}{2|w_k|} + |w_k|$$

(c) To Prove $|w_{k+1}| \leq |w_k| + \frac{3\gamma}{4}$.

If we assume $|w_k| > \frac{2}{\gamma}$

$$\therefore \quad w_{k+1} \leq \frac{\gamma}{2} + \frac{1}{2|w_k|} + |w_k| \quad ,$$

Assume $1/|w_k| < \frac{\gamma}{2}$

$$\Rightarrow \quad w_{k+1} \leq \frac{\gamma}{2} + \frac{\gamma}{4} + |w_k|$$

$$\boxed{|w_{k+1}| \leq \frac{3\gamma}{4} + |w_k|}$$

γ^2 |.

(d) To prove $M \leq 8/\gamma^2$

After M updates.

$$|w_{M+1}| \leq \frac{2}{\gamma} + \frac{3M\gamma}{4}$$

since.

$$|w_{k+1}| \leq \frac{3\gamma}{4} + |w_k|$$

for $k = 1$

$$|w_2| \leq \frac{3\gamma}{4} + |w_1|$$

$$|w_2| \leq \frac{3\gamma}{4} + \frac{2}{\gamma} \qquad (|w_1| \geq \frac{2}{\gamma})$$

$$|w_3| \leq \frac{3\gamma}{4} + |w_2|$$

$$|w_3| \leq \frac{3\gamma}{4} + \frac{3\gamma}{4} + \frac{2}{\gamma}$$

$$|w_3| \leq \left(\frac{3\gamma}{4}\right)(2) + \frac{2}{\gamma}$$

By induction

$$|w_{M+1}| \leq \left(\frac{3\gamma}{4}\right) M + \frac{2}{\gamma}. \qquad -(1)$$

Since

$$w_{t+1} \cdot w_{opt} > w_t \cdot w_{opt} + \gamma .$$

Then if $t = 1$ $\quad |w_1| = 0$.

$\rightarrow w_2 \cdot w_{opt} > \gamma$

$\rightarrow w_3 \cdot w_{opt} > w_2 \cdot w_{opt} + \gamma$

$w_3 \cdot w_{opt} > 2\gamma .$

By induction

$$|w_{M+1}| \cdot |w_{opt}| > M\gamma . \qquad -(2) .$$

Solving using (1) and (2).

After M updates, we get

$$M\gamma \leq \frac{2}{\gamma} + 3\frac{M\gamma}{4}$$

$$\Rightarrow \quad \frac{M\gamma}{4} \leq \frac{2}{\gamma}$$

$$\Rightarrow \quad \boxed{M \leq \frac{8}{\gamma^2}} .$$

Ans 1) b).

Let the optimal hyperplane be represented as follows: $< w_{opt}, b_{opt} >$

The support vectors will be (3, 5) and (4, 7). They will form the margin.

For (3, 5) and (4, 7) the functional margin should be 1.

i.e. $y_i (w_{opt} x_i + b) = 1$

for (3, 5) $y_i = 1$

$\Rightarrow 1 (< w_1 \; w_2 > < 3 \; 5 > + b) = 1$

$\Rightarrow 3 w_1 + 5 w_2 + b = 1 \quad — (1)$.

for (4, 7) $y_i = -1$.

$\Rightarrow -1 (< w_1 \; w_2 > < 4, 7 > + b) = 1$

$\Rightarrow -4 w_1 - 7 w_2 - b = 1 \quad — (2)$

Adding 1 and 2.

$- w_1 - 2 w_2 = 2$

$w_1 + 2 w_2 = -2$. $\quad — (3)$

To maximise the margin, we minimise $\|\vec{w}\|$

$\therefore$ minimise $\|\vec{w}\|$

$$\|\vec{w}\|^2 = w_1^2 + w_2^2$$

Substituting from (3):

$$\begin{aligned}\|\vec{w}\|^2 &= (-2 - 2w_2)^2 + w_2^2 \\ &= 4 + 4w_2^2 + 8w_2 + w_2^2 \\ &= 5w_2^2 + 8w_2 + 4\end{aligned}$$

In order to minimise, we find the derivative.

$$\frac{\partial \|w\|^2}{\partial w_2} = 10w_2 + 8 = 0$$

$$\boxed{w_2 = \frac{-8}{10} = -\frac{4}{5}.} \quad (4)$$

From (3)

$$w_1 = -2 + \frac{2 \times 4}{5}$$

$$\boxed{w_1 = -\frac{2}{5}.} \quad (5)$$

Substituting from (4) and (5) in (1).

$$\boxed{b = \frac{31}{5}.}$$

∴ Hyperplane is

$$\frac{2}{5}x + \frac{4}{5}y - \frac{31}{5} = 0$$

Geometric margin

$$\gamma_{geo} = \frac{\gamma_{func}}{\|w\|}$$

$$= \frac{1}{\|w\|}$$

$$= \frac{1}{\sqrt{\frac{4}{25} + \frac{16}{25}}}$$

$$= \frac{5}{\sqrt{20}}$$

Dual variables

$$w = \sum \alpha_i y_i x_i$$

only for (3,5) and (4,7) we will get non zero α.

Let α_1 be α of (3, 5)
α_2 be α of (4, 7)

$$\alpha_1 [3 \quad 5] + \alpha_2 [4 \quad 7] = [2 \quad 4]$$

We will get 2 equations

$$3\alpha_1 + 4\alpha_2 = 2$$
$$5\alpha_1 + 7\alpha_2 = 4.$$

On solving

$$5(3\alpha_1 + 4\alpha_2 = 2)$$
$$3(5\alpha_1 + 7\alpha_2 = 4).$$

$$15\alpha_1 + 20\alpha_2 = 10$$
$$15\alpha_1 + 21\alpha_2 = 12$$

$$\boxed{\alpha_2 = 2} \text{ for } (4, 7)$$

$$3\alpha_1 + 8 = 2.$$
$$\alpha_1 = -\frac{6}{3} = -2.$$

$$\boxed{\alpha_1 = -2.} \text{ for } (3, 5).$$