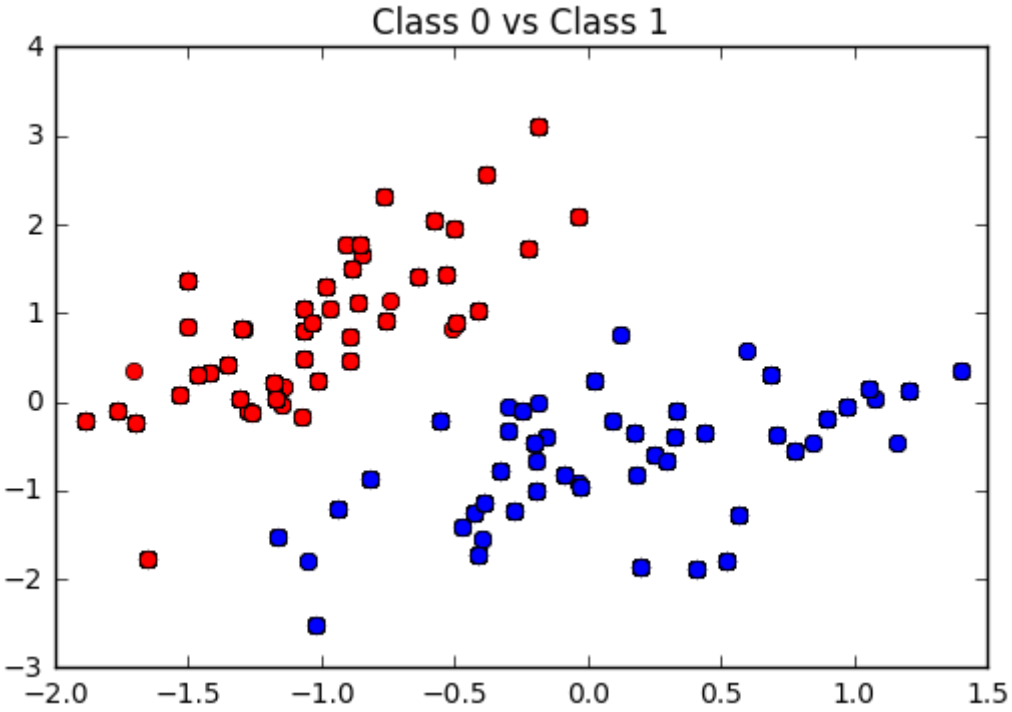
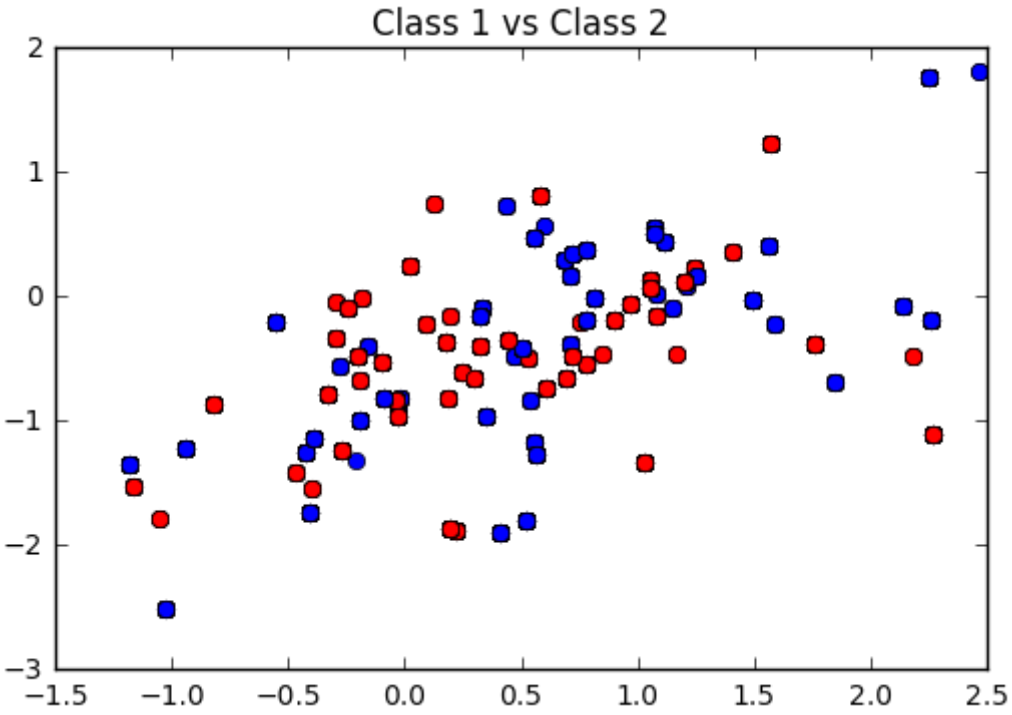
In [1]:

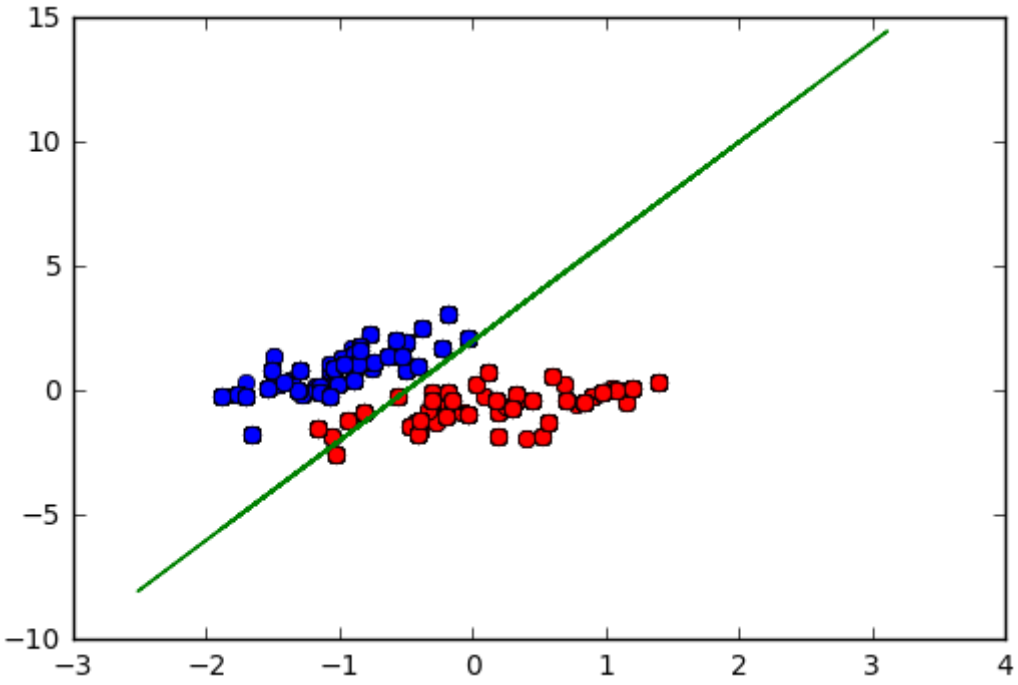
```
import numpy as np
import matplotlib.pyplot as plt
import mltools as ml
import mltools.logistic2 as lc2
import importlib

iris = np.genfromtxt("data/iris.txt",delimiter=None)
X, Y = iris[:,0:2], iris[:,-1] # get first two features & target
X,Y = ml.shuffleData(X,Y) # reorder randomly (important later)
X,_ = ml.transforms.rescale(X) # works much better on rescaled data
XA, YA = X[Y<2,:], Y[Y<2] # get class 0 vs 1
XB, YB = X[Y>0,:], Y[Y>0]


#print(len(XA[:,0]))
#print(len(YA))
#print(YB)
plt.clf()
for i in range(len(YA)):
    if YA[i]== 0:
        plt.plot(XA[i:,0],XA[i:,1], 'ro',color='red')
    else:
        plt.plot(XA[i:, 0], XA[i:, 1], 'ro', color='blue')
plt.title("Class 0 vs Class 1")
plt.show()

plt.clf()
for i in range(len(YA)):
    if YA[i] == 1:
        plt.plot(XB[i:, 0], XB[i:, 1], 'ro', color='red')
    else:
        plt.plot(XB[i:, 0], XB[i:, 1], 'ro', color='blue')
plt.title("Class 1 vs Class 2")
plt.show()
```

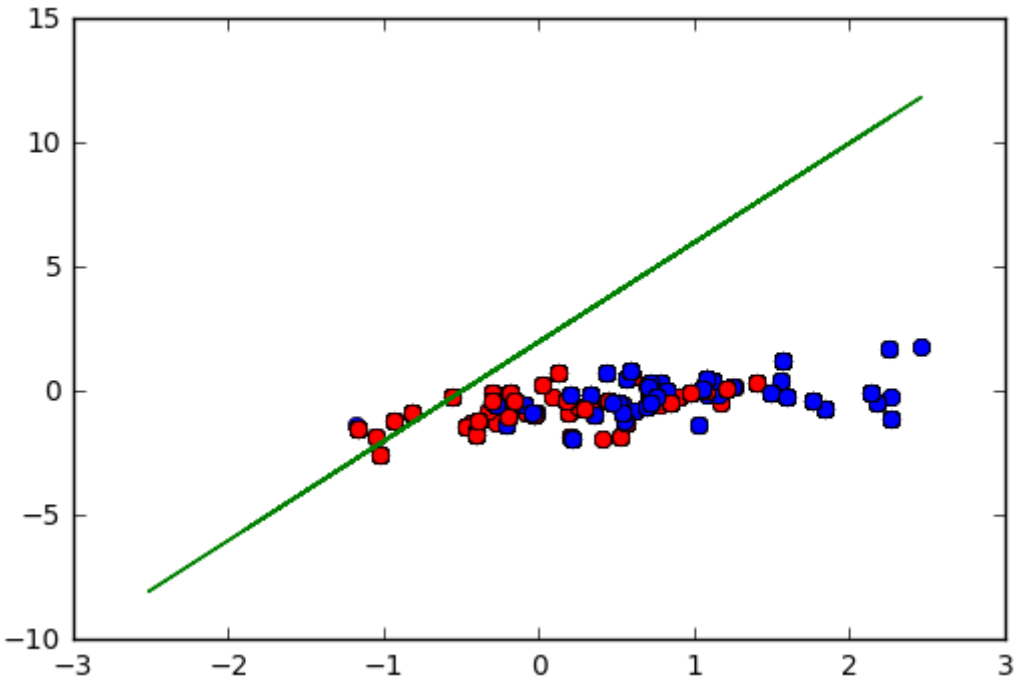

Class 0 vs Class 1
4
3
2
1
0
−1
−2
−3
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


Class 1 vs Class 2
2
1
0
−1
−2
−3
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0
2.5

In [67]:

```
reload(lc2)
learner = lc2.logisticClassify2()
learner.classes = np.unique(YA)
theta0 = 0.5
theta1 = 1
theta2 = -0.25
YAhat = [None] * len(XA)
YBhat = [None] * len(XB)
wts = [theta0, theta1, theta2]
learner.theta = wts
learner.plotBoundary(XA, YA)
plt.show()
learner.plotBoundary(XB, YB)
plt.show()
```





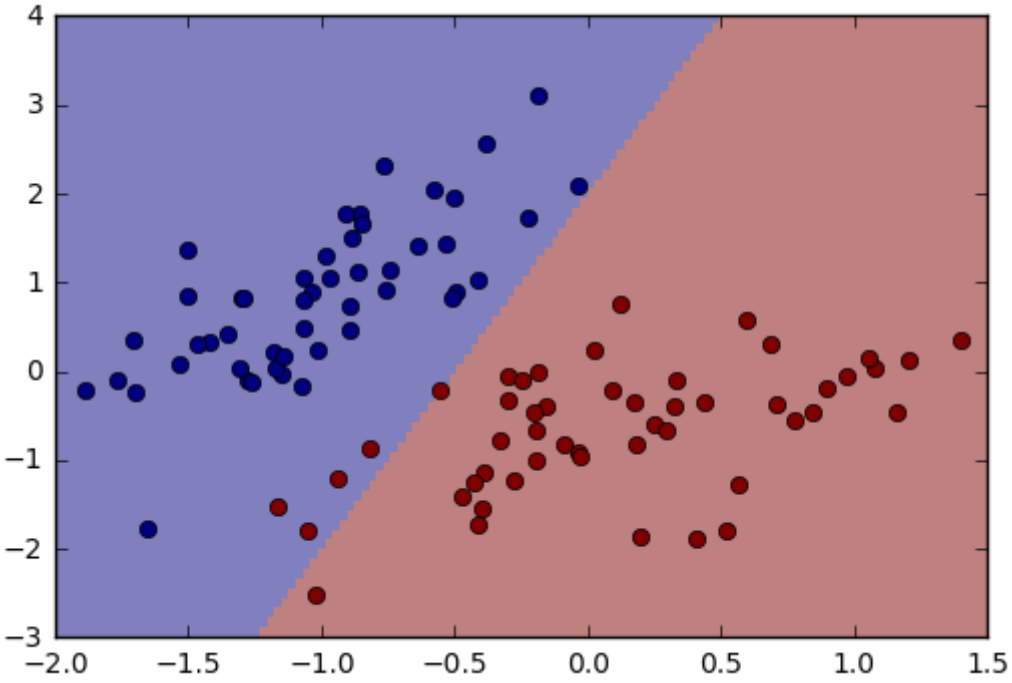
In [68]:

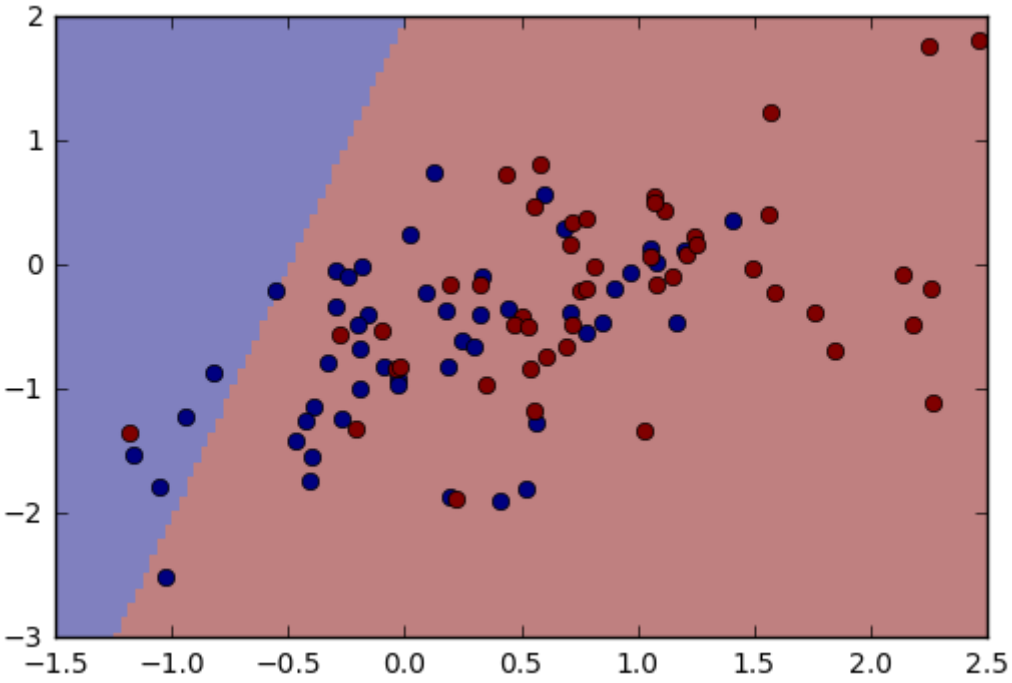
```
YAhat = learner.predict(XA)
YBhat = learner.predict(XB)
Aerr = 0
Berr = 0

for i in range(len(XA)):
    Aerr += ((YA[i] - YAhat[i])**2)
Aerr /= len(XA);
for i in range(len(XB)):
    Berr += ((YB[i] - YBhat[i])**2)
Berr /= len(XB);

print(Aerr)
print(Berr)
```

```
0.0505050505051
0.575757575758
```

In [34]:

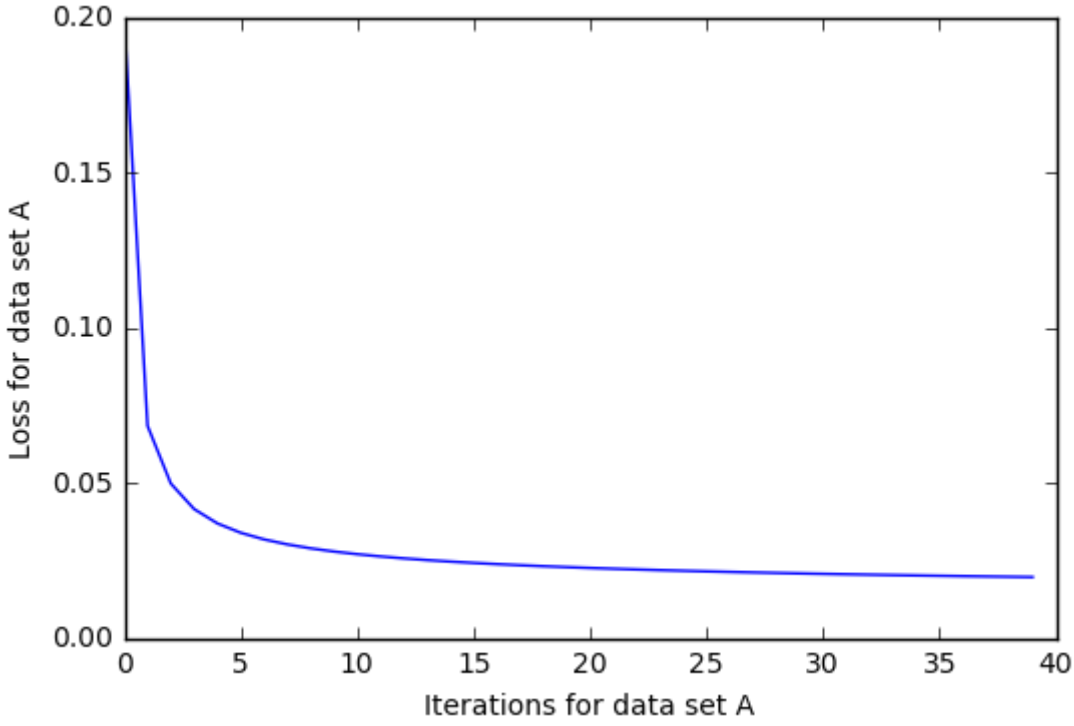
```
#plt.clf()
ml.plot.plotClassify2D(learner,XA,YA)
plt.show()
#learner.plotBoundary(XA,YA)
ml.plot.plotClassify2D(learner,XB,YB)
plt.show()
#learner.plotBoundary(XB,YBhat)
```

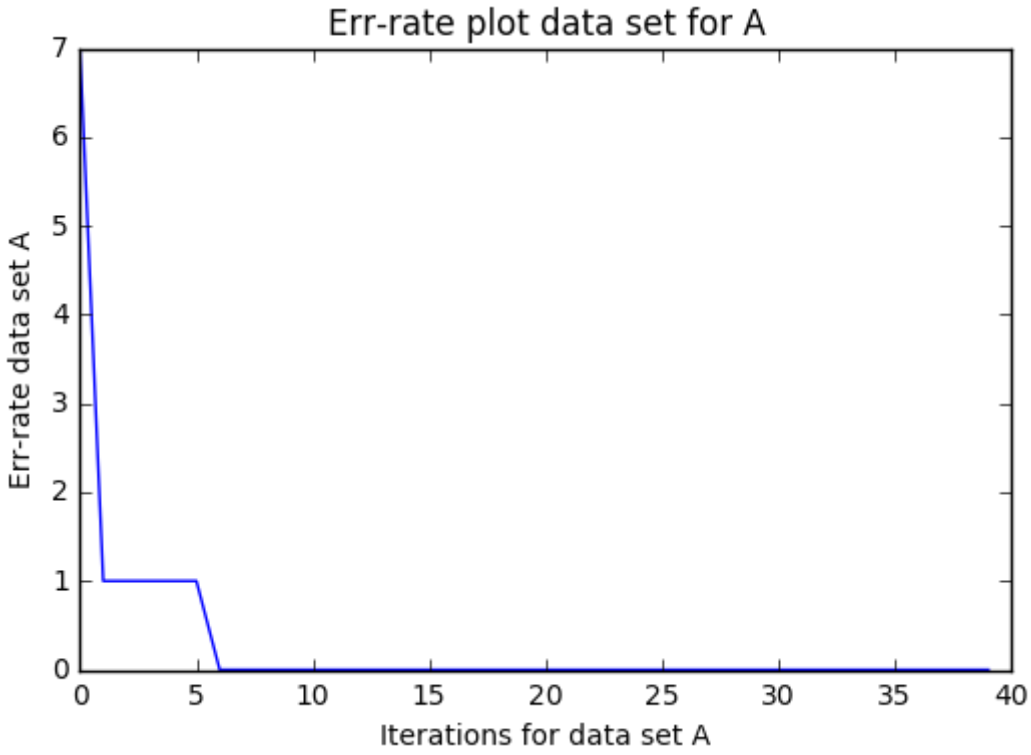


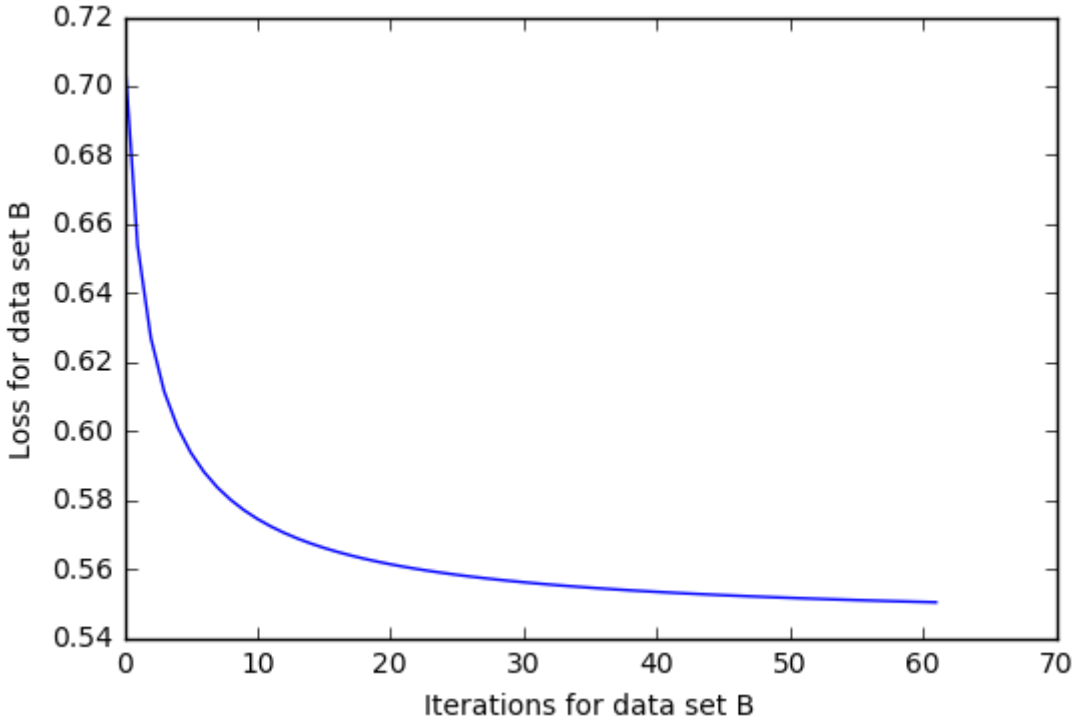


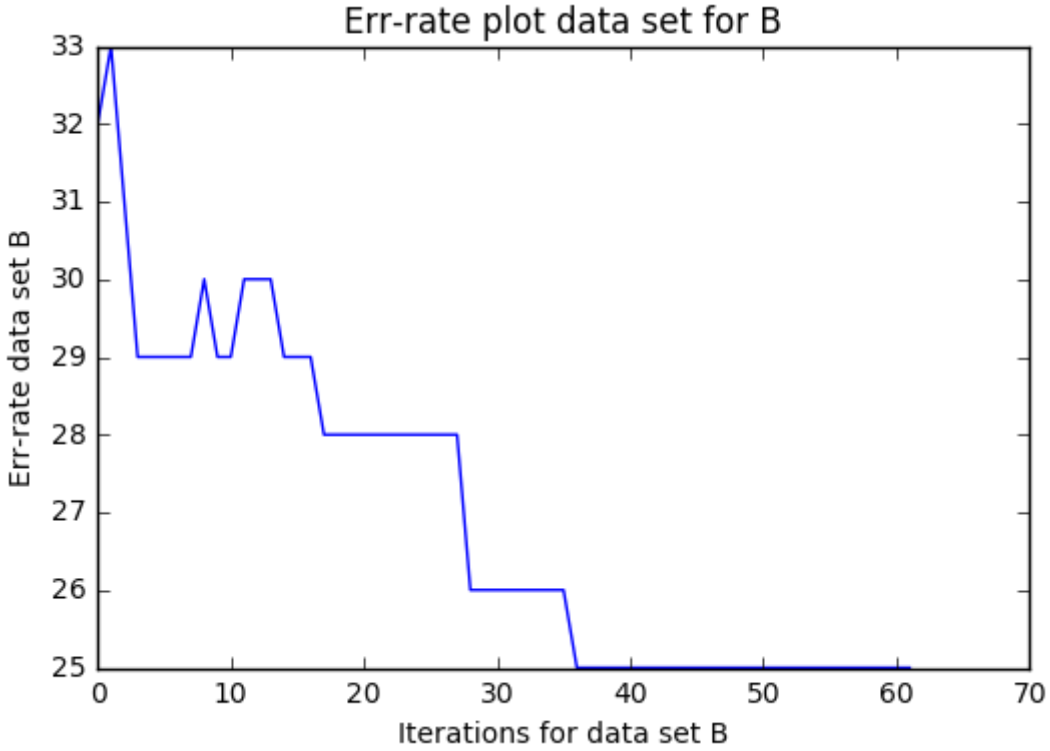
In [3]:

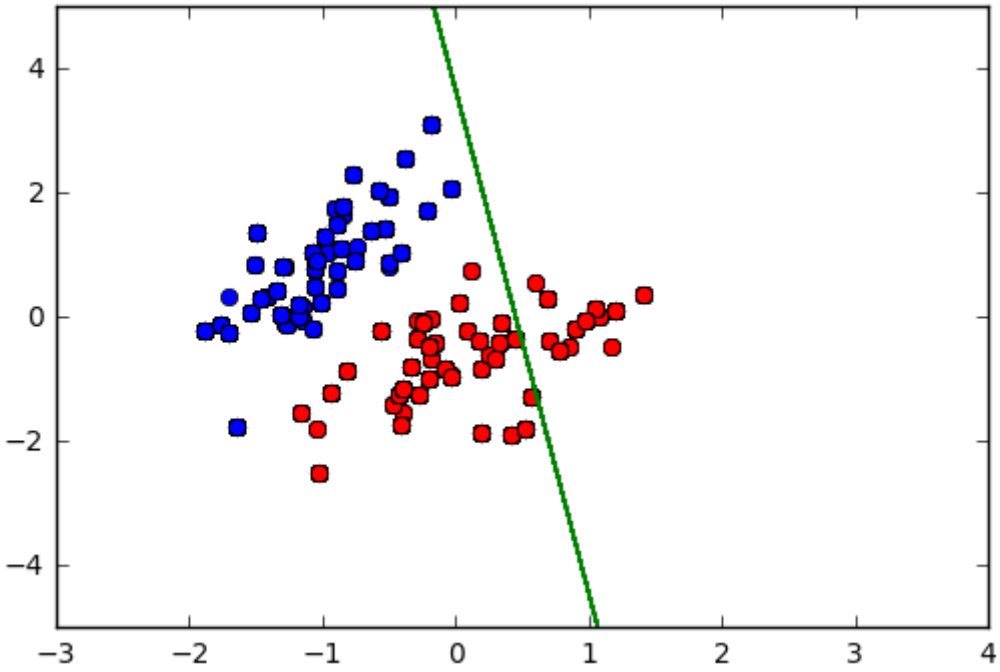
```
#num interations = 5000 & step size = 1

plt.clf()
reload(lc2)
learner=lc2.logisticClassify2();
learner.train(XA,YA,stopIter=5000,initStep=1.0)
plt.plot(learner.loss)
plt.xlabel("Iterations for data set A")
plt.ylabel("Loss for data set A")
plt.show()

plt.title("Err-rate plot data set for A")
plt.plot(learner.error)
plt.xlabel("Iterations for data set A")
plt.ylabel("Err-rate data set A")
plt.show()



plt.clf()
reload(lc2)
learner=lc2.logisticClassify2();
learner.train(XB,YB,stopIter=5000,initStep=1.0)
plt.plot(learner.loss)
plt.xlabel("Iterations for data set B")
plt.ylabel("Loss for data set B")
plt.show()


plt.title("Err-rate plot data set for B")
plt.plot(learner.error)
plt.xlabel("Iterations for data set B")
plt.ylabel("Err-rate data set B")
plt.show()


plt.title("final convereged classifier data set A")
learner.plotBoundary(XA,YA)
plt.ylim(-5,5)
plt.show()
plt.title("final convereged classifier data set B")
learner.plotBoundary(XB,YB)
plt.ylim(-5,5)
#plt.ylim(max(YB),min(YB))
plt.show()
```

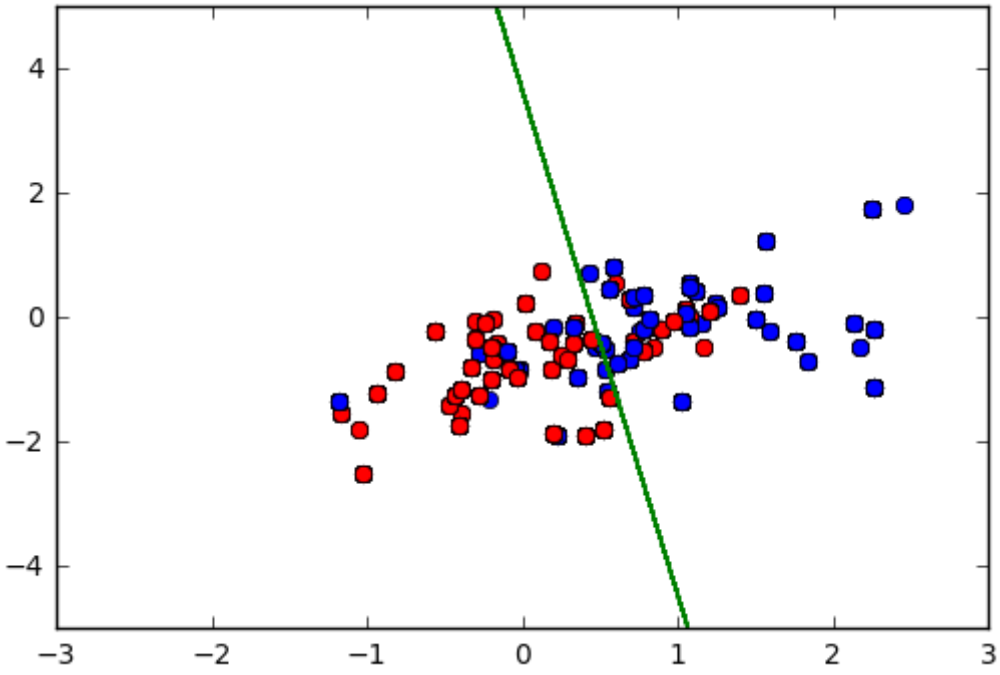

0.20
0.15
0.10
0.05
0.00
Loss for data set A
0
5
10
15
20
25
30
35
40
Iterations for data set A


Err-rate plot data set for A
7
6
5
4
3
2
1
0
Err-rate data set A
0
5
10
15
20
25
30
35
40
Iterations for data set A


0.72
0.70
0.68
0.66
0.64
0.62
0.60
0.58
0.56
0.54
Loss for data set B
0
10
20
30
40
50
60
70
Iterations for data set B


Err-rate plot data set for B
33
32
31
30
29
28
27
26
25
Err-rate data set B
0
10
20
30
40
50
60
70
Iterations for data set B


4
2
0
−2
−4
−3
−2
−1
0
1
2
3
4


4
2
0
−2
−4
−3
−2
−1
0
1
2
3

In [ ]:

```
2.a) for the case T(a+bx1) the desicion boundary will be a line parallel to the x2
axis.
    this will shatter the figures a,b
in case c,d  when the (2,2)&(6,4) are of same class this will not be shatter.

2.b)for T( (x1-a)2 + (x2-b)2 + c ) the desicion boundary will be a circle centered at a,
.
    this will shatter the cases a,b

    but in the case of c & d if we look at the points A(2,2) & B(4,8) when they are in
the same class
    and C(6,4)is difeerent class
    then there is no possible way to draw a circle containing A,B which will not encomp
ass the C(6,4)
    as (6,4) piont is exactly half the distanceof(AB) from the midpoint of AB segment (3
5).


2.c)for T( (a*b)x1 + (c/a)x2 ) the desicion boundary will be a plane passing throug ori
gin.
    in the cases a,b it will shatter.
    but in the case c,d when the (4,8) & (6,4) are ofsame class & (2,2) is in different
 class then
    using this desicion boundary we cannot shatter.
    As the slope of the point (2,2) lies between slope of(4,8) and slope (6,4).
```

$$z = x^{(i)} \cdot \theta^T$$

$$\sigma(z) = \frac{1}{1+e^{-z}}.$$

$$J_j(\theta) = -y^{(i)} \log\left[\frac{1}{1+e^{-z}}\right] - (1-y^{(i)}) \log\left[1 - \frac{1}{1+e^{-z}}\right]$$

$$= -y^{(i)} \log[\sigma(z)] - (1-y^{(i)}) \log[1-\sigma(z)].$$

$$\frac{\partial J_j(\theta)}{\partial \theta_j} = \left[+y^{(i)} \frac{1}{\sigma(z)} - (1-y^{(i)}) \frac{1}{1-\sigma(z)}\right] \frac{\partial}{\partial \theta_j} \sigma(z)$$

$$= \left[\frac{+y^{(i)}}{\sigma(z)} - \frac{(1-y^{(i)})}{1-\sigma(z)}\right] \frac{\partial}{\partial \theta_j}\left(\frac{1}{1+e^{-z}}\right) \quad \left[\because \text{Substituting } \sigma(z)\right]$$

$$= \left[\frac{+y^{(i)}}{\sigma(z)} - \frac{(1-y^{(i)})}{1-\sigma(z)}\right] \left[\frac{e^{-z}}{(1+e^{-z})^2}\right] \times \frac{\partial z}{\partial \theta_j} \quad \left[\text{as } \frac{\partial \sigma(z)}{\partial \theta_j} = \frac{\partial \sigma(z)}{\partial z} \times \frac{\partial z}{\partial \theta_j}\right]$$

$$= \left[\frac{+y^{(i)}}{\sigma(z)} - \frac{(1-y^{(i)})}{1-\sigma(z)}\right] \left[1 - \frac{1}{(1+e^{-z})}\right]\left[\frac{1}{1+e^{-z}}\right] x_j$$

$$\left[\text{as } \frac{\partial z}{\partial \theta_j} = x^j\right]$$

$$= \left[\frac{+y^{(i)}}{\sigma(z)} - \frac{(1-y^{(i)})}{1-\sigma(z)}\right] \left[(1-\sigma(z)) \times \sigma(z)\right] x_j$$

$$= \left[+y^{(i)} \times (1-\sigma(z)) - (1-y^{(i)}) \sigma(z)\right] x_j$$

$$= (y^{(i)} - \sigma(z)) x_j$$