Problem 2

1. a)

   From the question, $x_i = \frac{1}{1+e^{-s_i}}$,
   We can calculate:
   $$\frac{\partial x_i}{\partial s_i} = (-1)\frac{e^{-s_i}(-1)}{(1+e^{-s_i})^2}$$
   Therefore,
   $$\begin{aligned}\frac{\partial E}{\partial s_i} &= -[t_i\frac{1}{x_i} + (1-t_i)\frac{-1}{1-x_i}] \cdot \frac{e^{-s_i}}{(1+e^{-s_i})^2} \\ &= -\frac{t_i - x_i}{x_i(1-x_i)} \cdot x_i(1-x_i) \\ &= x_i - t_i\end{aligned}$$
   According to the chain rule, we have:
   $$\begin{aligned}\frac{\partial E}{\partial w_{ji}} &= \frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial w_{ji}} \\ &= (x_i - t_i)y_j\end{aligned}$$
   We use the same method to w_{kj}.
   Denote:
   $$y_j = \frac{1}{1+e^{-u_j}} \qquad u_j = \sum_k z_k w_{kj}$$
   According to the chain rule, we have:
   $$\begin{aligned}\frac{\partial E}{\partial w_{kj}} &= \sum_i \frac{\partial E}{\partial s_i}\frac{\partial s_i}{\partial y_j}\frac{\partial y_j}{\partial u_j}\frac{u_j}{w_{kj}} \\ &= \sum_i (x_i - t_i) \cdot w_{ji} \cdot (-1)\frac{e^{-u_j}(-1)}{(1+e^{-u_j})^2} \cdot z_k \\ &= \sum_i (x_i - t_i) w_{ji} y_j (1-y_j) z_k\end{aligned}$$

2. b)

   Firstly, we consider $x_k = \frac{e^{s_k}}{\sum_{c=1}^{m} e^{s_c}}$.

When $k = i$,

$$\begin{aligned}
\frac{\partial x_k}{\partial s_i} &= \frac{\partial x_i}{\partial s_i} \\
&= \frac{e^{s_i} \cdot \sum_{c=1}^m e^{s_c} - e^{s_i} \cdot e^{s_i}}{(\sum_{c=1}^m e^{s_c})^2} \\
&= \frac{e^{s_i}}{\sum_{c=1}^m e^{s_c}} \cdot \frac{\sum_{c=1}^m e^{s_c} - e^{s_i}}{\sum_{c=1}^m e^{s_c}} \\
&= x_i(1 - x_i)
\end{aligned}$$

When $k \neq i$,

$$\begin{aligned}
\frac{\partial x_k}{\partial s_i} &= e^{s_k} \cdot [-\frac{e^{s_i}}{(\sum_{c=1}^m e^{s_c})^2}] \\
&= -x_i x_k
\end{aligned}$$

Therefore,

$$\begin{aligned}
\frac{\partial E}{\partial s_i} &= \sum_{k=i} -(t_k \frac{1}{x_k}) x_i (1 - x_i) + \sum_{k \neq i} -(t_k \frac{1}{x_k}) \cdot (-x_i x_k) \\
&= -t_i(1 - x_i) + \sum_{k \neq i} x_i t_k \\
&= -t_i + \sum_k x_i t_k \\
&= x_i - t_i
\end{aligned}$$

According to the chain rule, we have

$$\begin{aligned}
\frac{\partial E}{\partial w_{ji}} &= \frac{\partial E}{\partial s_i} \frac{\partial s_i}{\partial w_{ji}} \\
&= (x_i - t_i) y_j
\end{aligned}$$

We use the same method to w_{kj}.
Denote

$$y_j = \frac{1}{1 + e^{-u_j}} \qquad u_j = \sum_k z_k w_{kj}$$

According to the chain rule, we have

$$\begin{aligned}
\frac{\partial E}{\partial w_{kj}} &= \sum_i \frac{\partial E}{\partial s_i} \frac{\partial s_i}{\partial y_j} \frac{\partial y_j}{\partial u_j} \frac{\partial u_j}{\partial w_{kj}} \\
&= \sum_i (x_i - t_i) \cdot w_{ji} \cdot (-1) \frac{e^{-u_j}(-1)}{(1 + e^{-u_j})^2} \cdot z_k \\
&= \sum_i (x_i - t_i) w_{ji} y_j (1 - y_j) z_k
\end{aligned}$$

Problem 3

Given the discrete distribution

$$\{p_k | k = 1, 2, ...N\}$$

with entropy

$$H = -\sum_{k=1}^{N} p_k . \log p_k$$

The constraint condition for maximization of the entropy is $\sum_{k=1}^{N} p_k = 1$
Lagrange multipliers equation is given by

$$L(p_1, p_2, ...p_N, \lambda) = -\sum_{k=1}^{N} p_k . \log p_k - \lambda . (\sum_{k=1}^{N} p_k - 1)$$

By Differentiating with respect to p_k and λ and equating to zero we get

$$\frac{\partial L(p_1, p_2, ...p_N, \lambda)}{p_k} = -\log p_k - 1 - \lambda = 0 \longrightarrow (1)$$

$$\frac{\partial L(p_1, p_2, ...p_N, \lambda)}{\lambda} = -\sum_{k=1}^{N} p_k + 1 = 0 \longrightarrow (2)$$

from 1 and 2 we get $\lambda = -\log p_k - 1 = \log(\frac{1}{p_k}) - 1$ and $\sum_{k=1}^{N} p_k = 1$ this is true if and only if $p_k = 1/N$
The condition for maximization of the entropy $\sum_{k=1}^{N} p_k = N.\frac{1}{N} = 1$ is also satisfied for $p_k = 1/N$
Thus the maximum entropy is

$$H_{max} = -\sum_{k=1}^{N} \frac{1}{N} . \log(\frac{1}{N}) = \log N$$

Entropy is maximized when the distribution is uniform $p_1 = p_2 = ...p_N = 1/N$

Problem 1

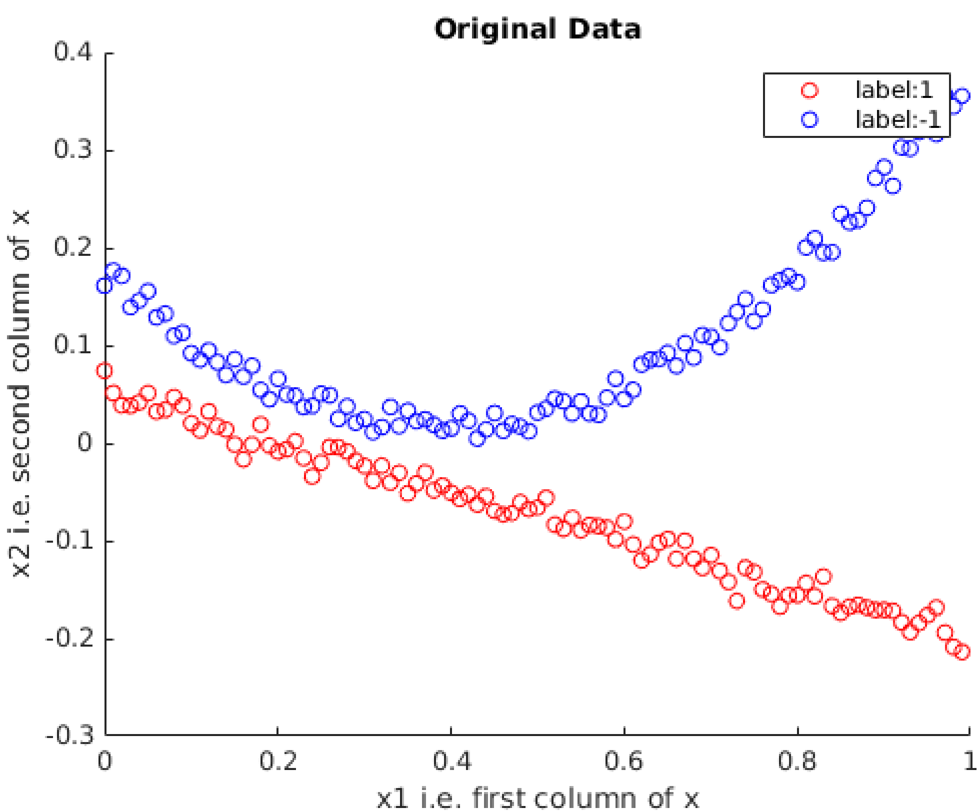
Import the data and labels

```
data = load('data3.mat').data;
x = data(:,1:2);
y = data(:,3);
```

Initial plot to see data

```
% x1 represents data with label 1
x1 = data(data(:,3)==1,1:2);
% x2 represents data with label -1
x2 = data(data(:,3)==-1,1:2);

figure;
scatter(x1(:,1),x1(:,2),'r')
hold on;
scatter(x2(:,1),x2(:,2),'b');
legend('label:1','label:-1');
xlabel('x1 i.e. first column of x');
ylabel('x2 i.e. second column of x');
title('Original Data')
```



Gradient descent set up

```
% z = x0*\theta where x0 is the feature matrix with a ones column appended
% z(i) = thetas(1)*1 + thetas(2)*x0(i,2) + thetas(3)*x0(i,3)
m = length(y);
n = length(x(1,:));
xo = [ones(m,1),x];
thetas = rand(n+1,1)*2-1;
iterations = 1000;
learning_rate = 0.1;
```

Gradient Descent

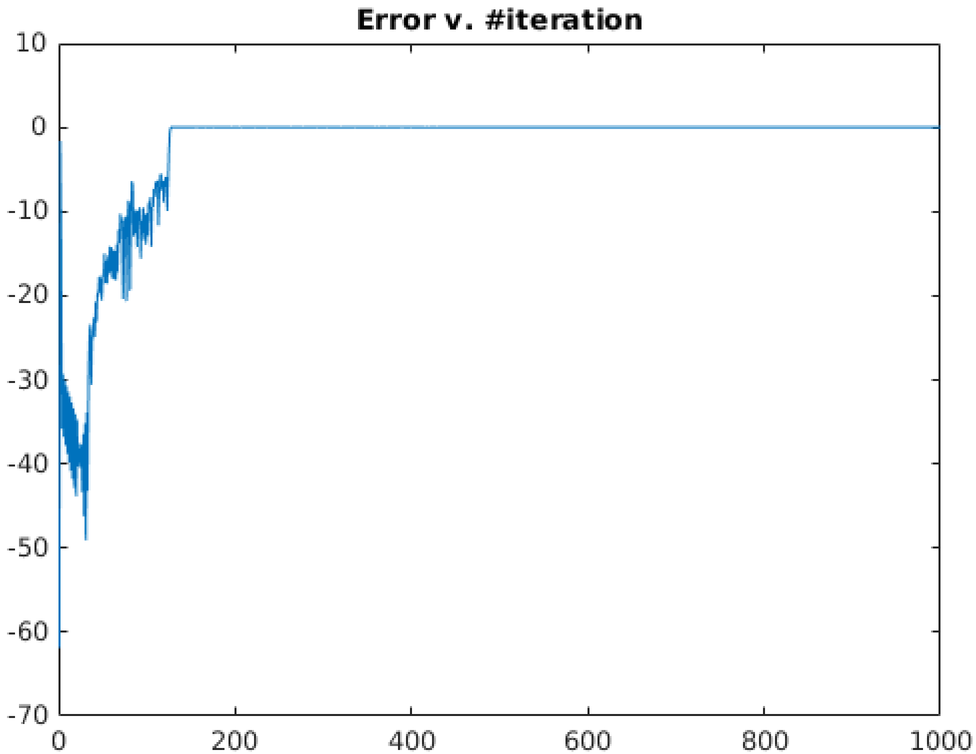
```
% errors = zeros(iterations,1);
% for i = 1:iterations
%     % get the predicted labels for the iteration
%     z = xo*thetas;
%     y_hat = ones(m,1);
%     y_hat_boolean = (z>=0);
%     y_hat(y_hat_boolean==0) = -1;
%     % get the misclassified values indices
%     misclassified_indices = (y_hat~=y);
%     %get gradient
%     gradient = -(1/m)*sum( y(misclassified_indices).*xo(misclassified_indices,:) );
%     %update thetas
%     thetas = thetas - learning_rate*transpose(gradient);
%     % get the error for the iteration
%     temp = xo*thetas;
%     errors(i) = -(1/m)*sum( y(misclassified_indices).*temp(misclassified_indices) );
% end
```

Stochastic GD

```
errors = zeros(iterations,1);
for i = 1:iterations
    % get the predicted labels for the iteration
    z = xo*thetas;
    y_hat = ones(m,1);
    y_hat_boolean = (z>=0);
    y_hat(y_hat_boolean==0) = -1;
    % get the misclassified values indices
    misclassified_indices = (y_hat~=y);
    %get gradient
    gradient = -y(misclassified_indices).*xo(misclassified_indices,:);
    %update thetas
    thetas = thetas - transpose(sum(gradient));
    % get the error for the iteration
    temp = xo*thetas;
    errors(i) = -(1/m)*sum( y(misclassified_indices).*temp(misclassified_indices) );
end
```

Plotting errors against iteration

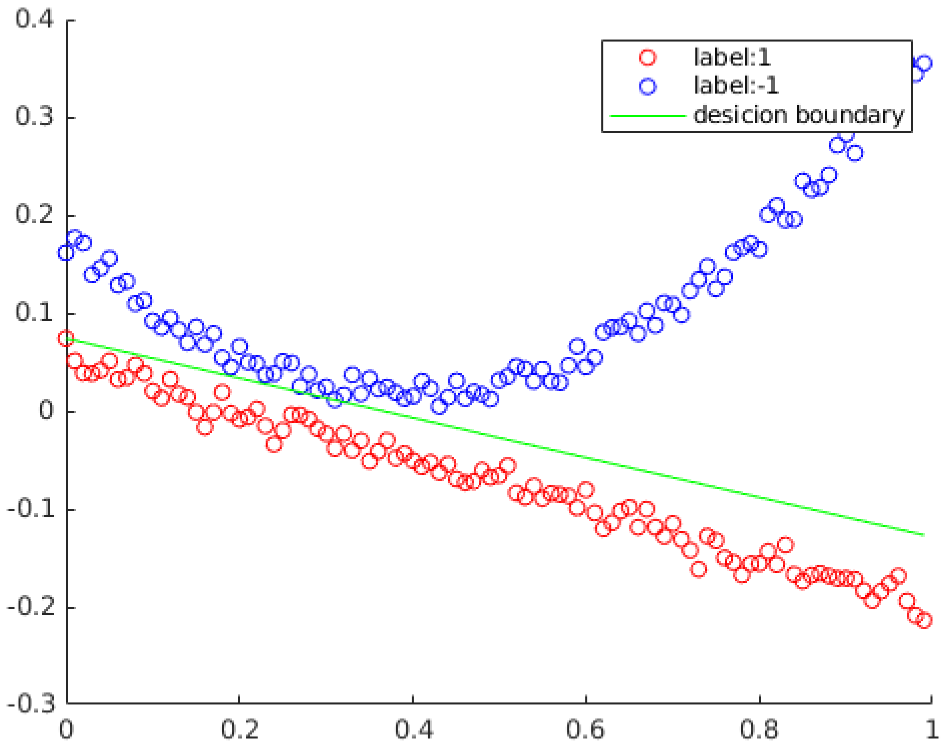
```
figure;
plot(errors)
title('Error v. #iteration')
```



Plot decision boundary

```
x1_lin = linspace(min(x(:,1)),max(x(:,1)),1000);
x2_lin = -(thetas(1) + thetas(2)*x1_lin)/thetas(3);

figure;
scatter(x1(:,1),x1(:,2),'r')
hold on;
scatter(x2(:,1),x2(:,2),'b');
hold on;
plot(x1_lin,x2_lin,'g-');
legend('label:1','label:-1','desicion boundary')
```


label:1
label:-1
desicion boundary
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
0
0.2
0.4
0.6
0.8
1

Axis aligned squares: $h = 3$. Why?

- There exist 3 points that can be shattered.

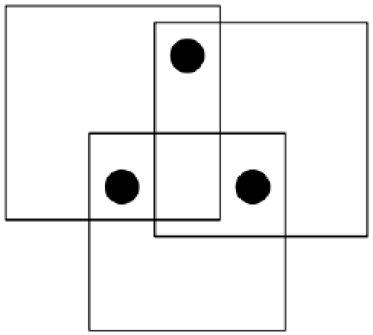


Figure: Figure from A. Bhaskar and I. Sukhar's class notes. VC dimension: axis-aligned squares. 3 points.

- No set of 4 points can be shattered.

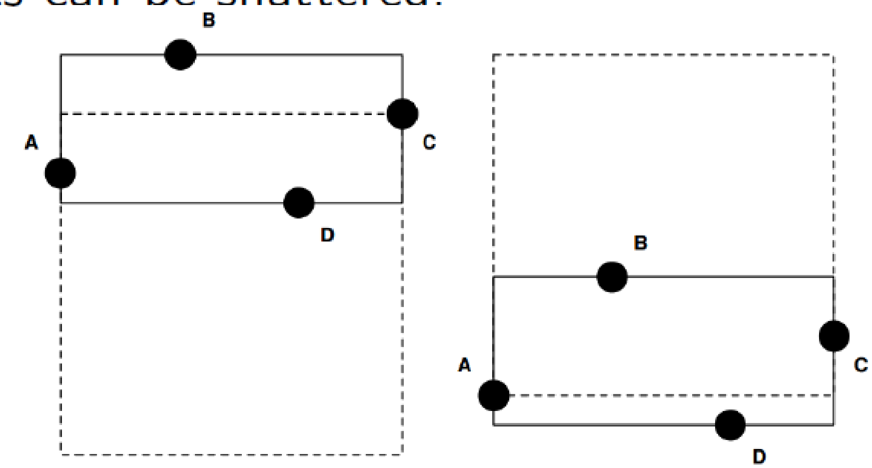


Figure: Figure from A. Bhaskar and I. Sukhar's class notes. VC dimension: axis-aligned squares. 4 points.