ECE m146 HW. #3

1. $J(w) = -\sum_{n=1}^{N} \left[y^{(n)} \log h_w(x^{(n)}) + (1-y^{(n)}) \log(1-h_w(x^{(n)})) \right] + \frac{1}{2} \sum_i w_i^2$

(a) $\frac{\partial J}{\partial w_0} = -\sum_{n=1}^{N} \left[y^{(n)} \frac{1}{h_w(x^{(n)})} h_w(x^{(n)}) (1-h_w(x^{(n)})) x_j^{(n)} + (1-y^{(n)}) \frac{-1}{1-h_w(x^{(n)})} h_w(x^{(n)}) \right.$

$\left. (1-h_w(x^{(n)})) \, x_j^{(n)} \right] + w_j$

$= -\sum_{n=1}^{N} y^{(n)} x_j^{(n)} - y^{(n)} x^{(n)} h_w(x^{(n)}) + y^{(n)} x^{(n)} h_w(x^{(n)}) - x_j^{(n)} h_w(x^{(n)})] + w_j$

$= -\sum_{n=1}^{N} \left[y^{(n)} x_j^{(m)} - \underbrace{h_w(x^{(n)})}_{\sigma(w^T x^{(n)})} x_j^{(n)} \right] + w_j$

$$W_{new} = W_{old} - \eta \frac{-\sum_{n=1}^{N} \left[y^{(n)} x_j^{(n)} - \sigma(w^T x^{(n)}) x_j^{(n)} \right] + w_j}{\left\| -\sum_{n=1}^{N} \left[y^{(n)} x_j^{(n)} - \sigma(w^T x^{(n)}) x_j^{(n)} \right] + w_j \right\|}$$

(b) $\nabla_w^2 J(w) = \begin{bmatrix} \frac{\partial^2 J}{\partial w_1^2} & \frac{\partial^2 J}{\partial w_1 w_2} \\ \frac{\partial^2 J}{\partial w_1 w_2} & \frac{\partial^2 J}{\partial w_2^2} \end{bmatrix}$

$\frac{\partial J}{\partial w_1} = -\sum_{n=1}^{N} \left[y^{(n)} - \sigma(w^T x^{(n)}) \right] + w_1$

$\frac{\partial J}{\partial w_2} = -\sum_{n=1}^{N} \left[y^{(n)} \cancel{x_2^{(n)}} - \sigma(w^T x^{(n)}) \, x_2^{(n)} \right] + w_2$

$\frac{\partial^2 J}{\partial w_1^2} = \sum_{n=1}^{N} \left[\sigma(w^T x^{(n)}) (1-\sigma(w^T x^{(n)})) \right] + 1$

$\frac{\partial^2 J}{\partial w_2^2} = \sum_{n=1}^{N} \left[\sigma(w^T x^{(n)}) (1-\sigma(w^T x^{(n)})) \right] + 1$

$\frac{\partial^2 J}{\partial w_1 \partial w_2} = \sum_{n=1}^{N} \left[\sigma(w^T x^{(n)}) (1-\sigma(w^T x^{(n)}) \, x^{(n)} \right]$

2. $w^* = \arg\max_w \prod p(y_i|x_i, w)\, f(w)$

$\log w^* = \arg\max_w \sum \log\left(p(y_i|x_i), w)\right) + \log f(w)$

$= \arg\max_w \sum \left[\log\left([\sigma(w^T x_i)]^{y_i} \cdot [1-\sigma(w^T x_i)]^{1-y_i} \right) + \log f(w) \right]$

$= \arg\max_w \sum \left[y_i \log\left(\sigma(w^T x_i)\right) + (1-y_i) \log\left(1-\sigma(w^T x_i)\right) + \log f(w) \right]$

$= -\arg\min_w \sum \left[y_i \log(\sigma(w^T x_i)) + (1-y_i)\log(1-\sigma(w^T x_i)) + \log\left(\cancel{\frac{1}{(2\pi)^{m/2}}}\right)^{\to 0} \quad \text{b/c constant} \right.$
$\left. + -\sum_{i=1}^{m} \frac{w_i^2}{2} \right]$

$= \arg\min_w -\sum \left[y_i \log(\sigma(w^T x_i)) + (1-y_i)\log(1-\sigma(w^T x_i)) \right] + \sum_{i=1}^{m} \frac{w_i^2}{2}$

Same as $\boxed{(1)}$ ▫

3. a) $H(\text{is Good Restaurant}) = H\left(\frac{6}{8}\right) = -\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4} = \boxed{0.811}$

b) ~~H(outdoor)~~ =

$H(\text{good} \mid \text{outdoor}) = \frac{5}{8} \cdot H\left(\frac{3}{5}\right) + \frac{3}{8} H\left(\cancel{\frac{3}{3}}\right)^{\to 0} = \boxed{0.607}$

c) Has bar : $\frac{4}{8}\left(H(4/4)\right) + \frac{4}{8} \cdot H\left(\frac{2}{4}\right) = \boxed{\frac{1}{2}}$

is clean : $\frac{4}{8} H(3/4) + \frac{4}{8} H\left(\frac{3}{4}\right) = \boxed{0.811}$

good At : $\frac{5}{8} \cdot H(1) + \frac{3}{8} H\left(\frac{1}{3}\right) = \boxed{0.344}$

(d) $I(\text{good}; \overset{\text{outdoor}}{\cancel{\text{bar}}}) = 0.811 - 0.607 = 0.204$

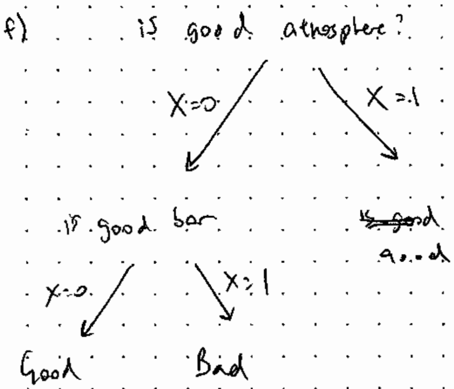
$I(\text{good}; \text{bar}) = 0.811 - 0.5 = 0.311$

clean = $= 0$

at ws = $= \cancel{0.467}\ 0.467$

e) atmosphere is the highest.

f)



outdoor	has bar	clean	atmos	good
1	1	0	0	0
1	1	1	0	0
1	0	1	0	1
0	0	0	1	1
0	1	1	1	1
1	0	0	1	1
1	1	0	1	1
0	0	1	1	1

g) 9 and 10 are both good restaurants because their atmospheres are good and that is enough to determine that it is good.

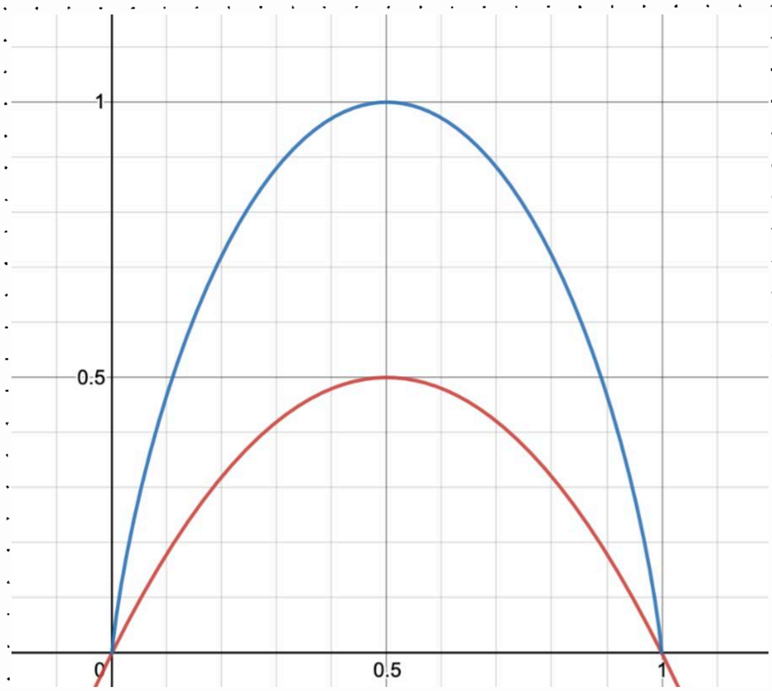
given atmos = 0

$H(\text{good} \mid \text{outdoor}) = 1 \cdot H(\frac{1}{3})$

$H(\text{has bar}) = \frac{1}{3} \cdot \cancel{H(1)} + \frac{2}{3} \cdot \cancel{H(0)} = 0$

$H(\text{clean}) = \frac{2}{3} \cdot H(\frac{1}{2}) + \frac{1}{3} \cdot \cancel{H(0)}$

#4 part a.



similarities:
- both symmetric about q(V)=0.5
- both are 0 at 0
- both are 0 at 1

red is gini(q(V)) and blue is H(q(V))

4) (b) if concave: $i(\lambda a_1 + (1-\lambda) a_2) \geq \lambda i(a_1) + (1-\lambda) i(a_2)$

$$\Rightarrow i(\lambda a_1 + (1-\lambda) a_2) - \lambda i(a_1) - (1-\lambda) i(a_2) \geq 0$$

$$\lambda = p(V_1, V) \qquad p(V_1, V) = \frac{|V_1|}{|V|}$$
$$1-\lambda = p(V_2, V) \qquad p(V_2, V) = \frac{|V_2|}{|V|}$$

$$\lambda a_1 + (1-\lambda) a_2 = \frac{|V_1|}{|V|} q(V_1) + \frac{|V_2|}{|V|} q(V_2)$$

$$= \frac{|\{i : i \in V_1, y_i = 1\}| + |\{i : i \in V_2, y_i = 1\}|}{|V|}$$

$$= \frac{|\{i : i \in V, y_i = 1\}|}{|V|} = q(V) \checkmark$$

since $i(\lambda a_1 + (1-\lambda) a_2) - \lambda i(a_1) - (1-\lambda) i(a_2) \geq 0$, b/c concave

$$i(q(V)) - (p(V_1, V) i(q(V_1)) + p(V_2, V) i(q(V_2))) \geq 0$$

$$I(V_1, V_2, V) \geq 0 \quad \square$$

(c) $y = -x \log x - (1-x) \log(1-x)$

$$= -\log x - x \cdot \frac{1}{x \cdot \ln 2} - (-1) \log(1-x) + \cancel{(1-x)} \cdot \frac{1}{\cancel{(1-x)} \cdot \ln 2} \cdot (-1)$$

$$= -\log x - \frac{1}{\ln 2} + \log(1-x) + \frac{1}{\ln 2}$$

$$= -\log_2 x + \log_2 (1-x)$$

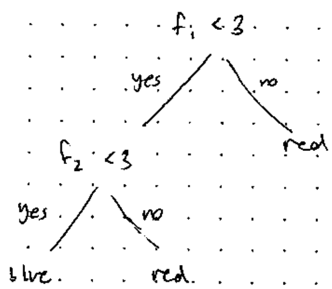
$$\frac{dy}{dx} = \frac{-1}{x \cdot \ln 2} + \frac{-1}{(1-x)\ln 2} < 0 \quad \text{so concave}$$

(d) $\text{gini}(x) = 2x(1-x) = -2x^2 + 2x$

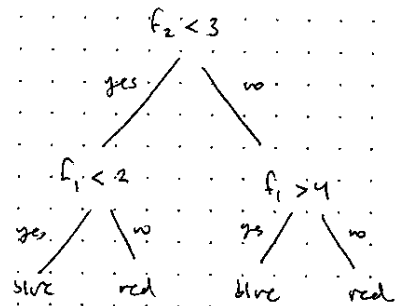
$$\frac{d^2 g}{dx^2} = -4 < 0 \quad \text{therefore concave}$$

5) (a) Depth 2 decision tree:

Ex: 1

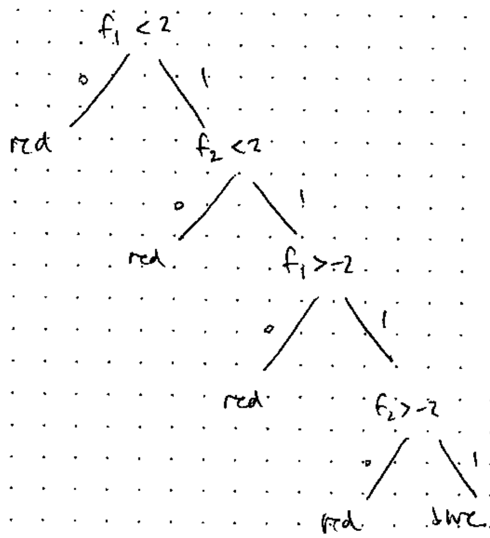


Ex: 3



(b) #2 is most complex b/c since it isn't axis aligned, there is a jagged edge btwn the two classes.

(c) #4 is now separable.



PROBLEM 6 ON HW 3

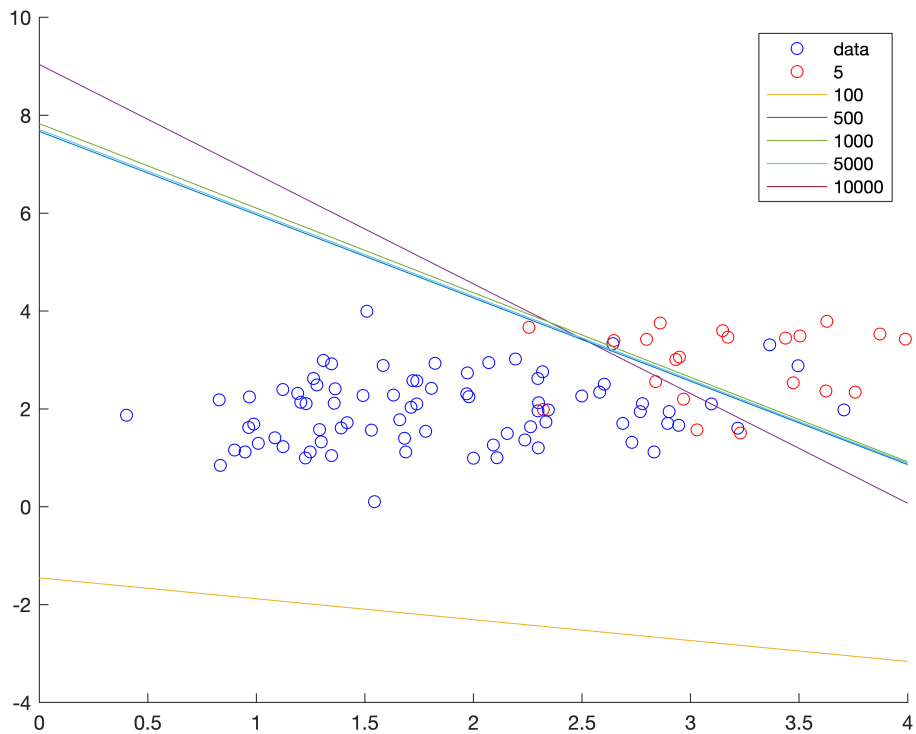
(a)

```
close all;
stat = readtable("UCLA_EE_grad_2030.csv");

x = [ones(1, size(stat, 1)); stat{:,1:2}'];
y = (stat{:,3} + 1) / 2;
w = zeros(3, 1);

marks = [5, 100, 500, 1000, 5000, 10000];

figure(1);
hold on;

GPA = stat{:,1};
GRE = stat{:,2};
scatter(GPA(y == 0), GRE(y == 0), 'blue');
scatter(GPA(y == 1), GRE(y == 1), 'red');
x1 = (0:0.1:4);

for iters = 1:10000
    hw = 1 ./ (1 + exp(-(w' * x)));
    del = sum((hw - y') .* x, 2);
    w = w - 0.01 * del;

    if any(marks(:) == iters)
        x2 = (-w(1) - x1 * w(2)) / w(3);
        plot(x1, x2);
        fprintf("Iteration %d:\n", iters);
        fprintf("    Weights: [%.4f %.4f %.4f]\n", w');
        J = -sum(y' .* log(hw) + (1 - y)' .* log(1 - hw));
        fprintf("    Loss J(w): %.4f\n", J);
        errors = 0;
        for point = 1:100
            if w' * x(:, point) * (y(point) * 2 - 1) < 0
                errors = errors + 1;
            end
        end
        fprintf("    Accuracy: %d%%\n", 100 - errors);
    end
end
```

```
Iteration 5:
    Weights: [-0.6077 -0.1788 -0.4186]
    Loss J(w): 62.1028
    Accuracy: 79%
Iteration 100:
    Weights: [-5.0591 1.2546 0.5597]
    Loss J(w): 31.5298
    Accuracy: 89%
Iteration 500:
    Weights: [-9.6026 2.1187 1.2261]
    Loss J(w): 23.3780
```

```
    Accuracy: 90%
Iteration 1000:
    Weights: [-11.2396 2.4857 1.4577]
    Loss J(w): 22.7539
    Accuracy: 90%
Iteration 5000:
    Weights: [-12.5365 2.7815 1.6341]
    Loss J(w): 22.6232
    Accuracy: 90%
Iteration 10000:
    Weights: [-12.5426 2.7829 1.6349]
    Loss J(w): 22.6232
    Accuracy: 90%
```

```
legend('data', '5', '100', '500', '1000', '5000', '10000');
```

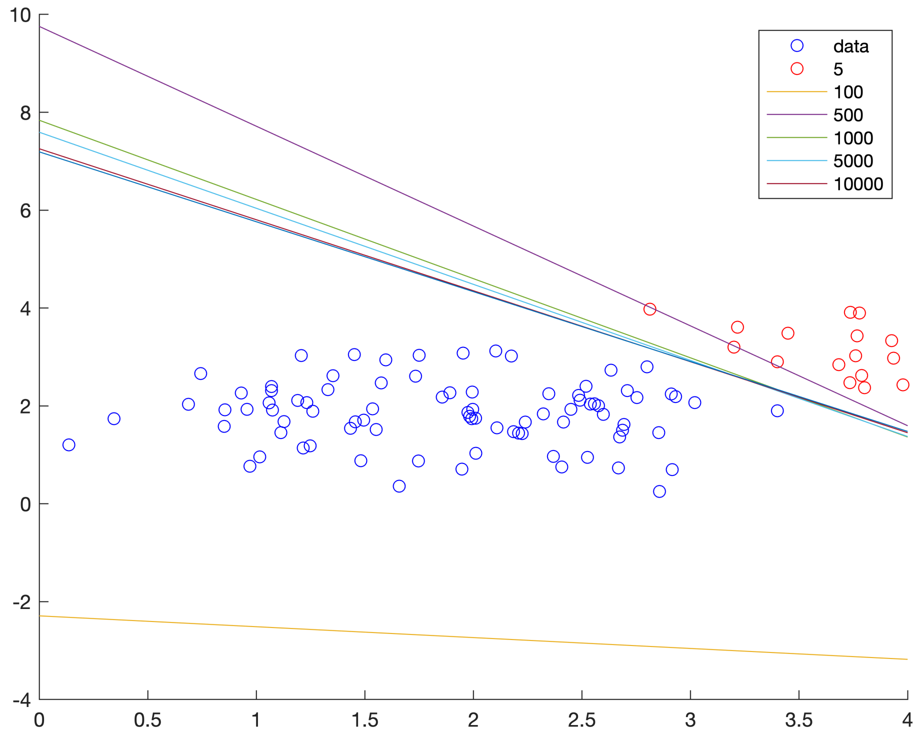



```
stat = readtable("UCLA_EE_grad_2031.csv");

x = [ones(1, size(stat, 1)); stat{:,1:2}'];
y = (stat{:,3} + 1) / 2;
w = zeros(3, 1);

marks = [5, 100, 500, 1000, 5000, 10000];

figure(2);
hold on;

GPA = stat{:,1};
```

```
GRE = stat{:,2};
scatter(GPA(y == 0), GRE(y == 0), 'blue');
scatter(GPA(y == 1), GRE(y == 1), 'red');
x1 = (0:0.1:4);

for iters = 1:10000
    hw = 1 ./ (1 + exp(-(w' * x)));
    del = sum((hw - y') .* x, 2);
    w = w - 0.01 * del;

    if any(marks(:) == iters)
        x2 = (-w(1) - x1 * w(2)) / w(3);
        plot(x1, x2);
        fprintf("Iteration %d:\n", iters);
        fprintf("    Weights: [%.4f %.4f %.4f]\n", w');
        J = -sum(y' .* log(hw) + (1 - y)' .* log(1 - hw));
        fprintf("    Loss J(w): %.4f\n", J);
        errors = 0;
        for point = 1:100
            if w' * x(:, point) * (y(point) * 2 - 1) < 0
                errors = errors + 1;
            end
        end
        fprintf("    Accuracy: %d%%\n", 100 - errors);
    end
end
```

```
Iteration 5:
    Weights: [-0.6147 -0.0596 -0.2684]
    Loss J(w): 52.5866
    Accuracy: 84%
Iteration 100:
    Weights: [-5.6729 1.1863 0.5815]
    Loss J(w): 18.4853
    Accuracy: 98%
Iteration 500:
    Weights: [-11.8075 2.4381 1.5066]
    Loss J(w): 7.0009
    Accuracy: 100%
Iteration 1000:
    Weights: [-15.1482 3.1011 1.9950]
    Loss J(w): 4.5682
    Accuracy: 100%
Iteration 5000:
    Weights: [-25.0273 5.0049 3.4495]
    Loss J(w): 1.6140
    Accuracy: 100%
Iteration 10000:
    Weights: [-30.3694 6.0317 4.2228]
    Loss J(w): 0.9940
    Accuracy: 100%
```

```

legend('data', '5', '100', '500', '1000', '5000', '10000');
```



(b) The linearly non-separable dataset, aka the first graph (graduation class of 2030) converges in the sense that the loss function barely decreases anymore. However, the accuracy is still not perfect, at 90%. The line also barely changes after 10000 iterations

tldr; if we were to run this algorithm until the data was perfectly separable, the algorithm would never converge. however, the value of loss function does seem to converge.

(c) On the other hand, the other graph (graduation class of 2031) aka the linearly separable dataset, seems to not converge with regards to loss function. It i still decreasing even after 10000 iterations. However, the accuracy with regards to testing data is 100%, and will not change from there. The line also is barely changing positions now. I tried running the second dataset with 1000000 iterations (1 million), and the loss function reached all the way down to 0.0203!!. It will just keep decreasing, as a better and better fit for the data will be the perfect minimal value of the loss function even though accuracy is stilll 100% through all of this. There is an empty space in between the lineraly separable data where the point can lie anywhere, and at this point the line is just perfecting its placement.

tldr; if we were to run this algorithm until the data was perfectly separable, the algorithm would converge in under 500 iterations. however, the value of the loss function does not seem to converge even after 1 million iterations.

7. (a) $k=1$

Ex 1: (0.5, 0.9) (0.9, 0.5), (0.5, 0.1) (0.1, 0.5)

work. everything else is wrong

~~$\boxed{4/14}$~~ $\boxed{10/14}$

Ex 2: (2.5, 2) and (~~+~~1.5, 2) don't work.

$\boxed{12/14}$

^^ supposed to be 12/14 are correct
cant see because i overlapped the box

$k=3$

Ex 1: row, (0.6, 0.8) works

error = 14 - 4.2 = 6

$\boxed{6/14}$

Ex 2: middle 4 points don't work.

~~k~~ 14 - 4 = 10

~~$\boxed{10/14}$~~ $\boxed{4/14}$

$k=5$

Ex 1: red + in top right get squished

error: $\boxed{4/14}$

Ex 2: outer 4 plus signs incorrectly classified

error: $\boxed{4/14}$

$k=7$

Ex 1 same as $k=5$ $\boxed{4/14}$

Ex 2 top + bottom blue misclassified 2

outer + minus classif. 2 + 6

$\boxed{8/14}$

(b)

Ex 1: $k=5, 7$ minimize

V.E.

Ex 2: $k=1$ minimizes

V.E.

big k: could include points from another cluster that are a diff. class

small k: risk a misclassification data point ruining data.

$k=13$: $\boxed{14/14}$ uh oh...