CSE 151: Homework 6 Solutions

1 Convexity and linear classification

1. *Convexity.*
   (a) $f''(x) = 2$: convex
   (b) $f''(x) = -2$: concave
   (c) $f''(x) = 2$: convex
   (d) $f''(x) = 0$: both convex and concave
   (e) $f''(x) = 6x$ and $x \in \mathbb{R}$: neither convex nor concave
   (f) $f''(x) = 12x^2$ and $x \in \mathbb{R}$: convex
   (g) $f''(x) = -\frac{1}{x^2}$ and $x \in \mathbb{R}$: concave

2. $M = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. For any vector $x = (x_1, x_2)$, we have $x^T M x = 2x_1 x_2$. This is not always ≥ 0; for instance, take $x_1 = 1$ and $x_2 = -1$. Thus M is not PSD.

3. $M = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$. For any vector $x = (x_1, x_2)$, we have $x^T M x = x_1^2 - 2x_1x_2 + x_2^2 = (x_1 - x_2)^2 \geq 0$. Thus M is PSD.

4. Let U be the matrix where the ith row is v_i. I.e.

$$U = \begin{pmatrix} \text{---} & v_1 & \text{---} \\ \text{---} & v_2 & \text{---} \\ & \vdots & \\ \text{---} & v_n & \text{---} \end{pmatrix}$$

   Then $(UU^T)_{ij} = v_i \cdot v_j = M_{ij}$. Thus M can be written as UU^T and is positive semidefinite.

5. $F(x)$ is convex. To see this, we take double partial derivatives to get

$$\frac{\partial F}{\partial x_i} = 2(x_i - u_i)$$

   and

$$\frac{\partial^2 L}{\partial x_i \partial x_j} = \begin{cases} 2 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

   Thus the Hessian H of F is a diagonal matrix with every diagonal entry set to 2. This is positive semidefinite since $z^T H z = 2\|z\|^2 \geq 0$ for all $z \in \mathbb{R}^d$.

6. Recall $F(x) = e^{u \cdot x}$.
   (a) We have $dF/dx_j = e^{u\cdot x} u_j$ and $d^2F/dx_i dx_j = e^{u \cdot x} u_i u_j$. Thus the Hessian matrix is $H(x) = e^{u \cdot x} uu^T$.
   (b) For any $z \in \mathbb{R}^d$, and any $x \in \mathbb{R}^d$, we have

$$z^T H(x) z = e^{u \cdot x} z^T uu^T z = e^{u \cdot x}(u \cdot z)^2 \geq 0.$$

   Thus $H(x)$ is positive semidefinite, and F is convex.

7. We want to analyze $F(p) = -\sum_{i=1}^m p_i \ln p_i$. We will show that F is concave by demonstrating that $G(x) = -F(x)$ is convex.

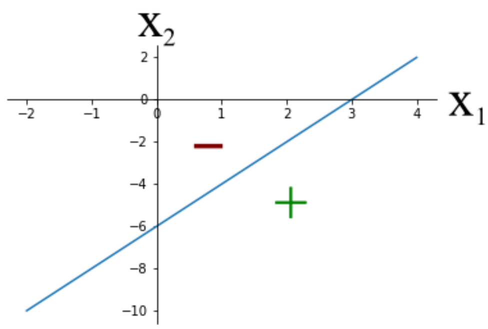
$$\frac{\partial G}{\partial p_i} = \ln p_i + \frac{p_i}{p_i} = 1 + \ln p_i$$

and

$$\frac{\partial^2 G}{\partial p_i \partial p_k} = \begin{cases} \frac{1}{p_i} & \text{if } i = k \\ 0 & \text{if } i \neq k \end{cases}$$

Since the Hessian is a diagonal matrix with nonnegative entries, it is positive semidefinite, and thus G is convex and F is concave.

8. The decision boundary plot should look something like the plot below.



9. (a) **Definitely true.** If the data set were not linearly separable, Perceptron would never converge.
   (b) **Definitely true.** Since the data is linearly separable, Perceptron is guaranteed to converge, no matter what the ordering of the points might be.
   (c) **Possibly false.** Different orderings of the data can produce different numbers of updates before convergence. We saw examples of this in class.
   (d) **Possibly false.** There could be several updates on any given data point, and thus k is not necessarily upper-bounded by n.

10. Each time the Perceptron algorithm performs an update a point with label y, it updates its offset b as $b = b + y$. Thus if we start with $b = 0$ and perform p updates on points with $y = -1$ and q updates on points with $y = +1$, then the final value of b is $b = q - p$.

2 Programming exercise

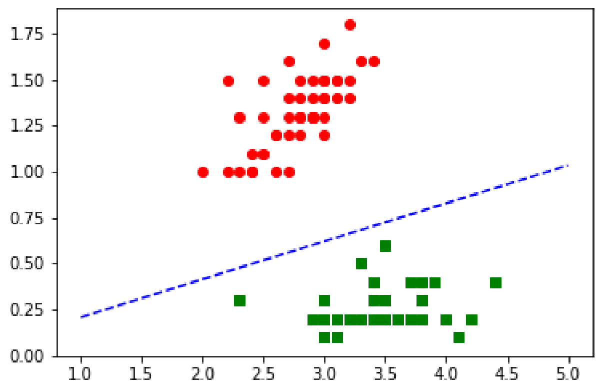
(a) The classification code can be written as follows.

```
def classify(w, b, x):
    return np.sign(np.dot(w,x) + b)
```

The perceptron algorithm can be written as follows.

```
def perceptron(data, labels):
    n = len(labels)
    inds = np.random.permutation(n)
    data = data[inds,:]
    labels = labels[inds]
    n_correct = 0
    w = np.zeros(np.shape(data)[1])
    b = 0
    while(n_correct < n):
        n_correct = 0
        for i in range(n):
            if (classify(w, b, data[i,:]) == labels[i]):
                n_correct += 1
            else:
                w = w + labels[i]*data[i,:]
                b = b + labels[i]
    return(w,b)
```

(c) The perceptron boundary should look something like the following plot.



(d) The histogram should look something like the following.