CSE 250B: Section 5 - Sharad Vikram

1. Perceptron Practice

   Recall in the perceptron algorithm, our classifier is of the form:

$$h_w(x) = \begin{cases} +1 & \text{if } w^T x > 0 \\ -1 & \text{if } w^T x \leq 0 \end{cases}$$

   Let's try running through a few iterations of the perceptron algorithm! Let $w_0 = [0, 0]$ and recall that the update rule is if a training point (x, y) is misclassified, update $w_{t+1} = w_t + yx$. Please record the following: (a) whether or not the point is misclassified (b) the new weights, and (c) a plot of the decision boundary. The first two have been filled out for you.

   1. Training point: $x_1 = (1, 1), y_1 = +1$
      (a) Point is *misclassified*.
      (b) $w_1 = w_0 + [1, 1] = [1, 1]$
      (c) See board
   2. Training point: $x = (2, -1), y = -1$
      (a) Point is *misclassified*.
      (b) $w_2 = w_1 - [2, -1] = [-1, 2]$
      (c) See board
   3. Training point: $x = (0, 2), y = +1$

      Solution:
      (a) $w^T x = 4 > 0$, the classification is correct.
      (b) No update.

   4. Training point: $x = (-3, -1), y = -1$

      Solution:
      (a) $w^T x = 1 > 0$, the classification is incorrect.
      (b) $w' = w - [-3, -1] = [2, 3]$

   5. Training point: $x = (3, 1), y = 1$

      Solution:
      (a) $w^T x = 9 > 0$, the classification is correct.
      (b) No update.

2. Matrix calculus
   Let A be a $n \times n$ matrix, and let $x \in \mathbb{R}^n$. Show the following identities are true (the gradient will be typically be an $n \times 1$ vector):

   (a) $\nabla_x(x^T y) = y$

(b) $\nabla_x(y^T x) = y$
(e) $\nabla_x(x^T x) = 2x$
(c) $\nabla_x(Ax) = A$
(d) $\nabla_x(x^T A) = A^T$
(f) $\nabla_x(x^T Ax) = (A + A^T)x$ (hint: use product rule)

Solution: From HW2, we know that

$$x^\top Ax = \sum_i \sum_j a_{ij} x_i x_j$$

Let's differentiate this with respect to a single element x_k:

$$\frac{\partial}{\partial x_k}(x^\top Ax) = \frac{\partial}{\partial x_k}\left(\sum_i \sum_j a_{ij} x_i x_j\right)$$

We can drop all terms that don't contain x_k:

$$= \frac{\partial}{\partial x_k}\left[\left(\sum_i a_{ik} x_i x_k\right) + \left(\sum_j a_{kj} x_k x_j\right) - a_{kk} x_k^2\right]$$

Isolating the x_k^2 terms gives

$$\begin{aligned}
&= \frac{\partial}{\partial x_k}\left[\left(\sum_{i\neq k} a_{ik} x_i x_k\right) + \left(\sum_{j\neq k} a_{kj} x_k x_j\right) + a_{kk} x_k^2\right] \\
&= \frac{\partial}{\partial x_k}\left[\left(\sum_{i\neq k} a_{ik} x_i\right) x_k + \left(\sum_{j\neq k} a_{kj} x_j\right) x_k + a_{kk} x_k^2\right]
\end{aligned}$$

Now we can differentiate with respect to x_k

$$\begin{aligned}
&= \left(\sum_{i\neq k} a_{ik} x_i\right) + \left(\sum_{j\neq k} a_{kj} x_j\right) + 2a_{kk} x_k \\
&= \left(\sum_i a_{ik} x_i\right) + \left(\sum_j a_{kj} x_j\right) \\
&= (k^{\text{th}} \text{ column of A})^\top x + (k^{\text{th}} \text{ row of A})^\top x
\end{aligned}$$

Placing all partial derivatives into a single vector, we get

$$\frac{d}{dx}(x^T Ax) = (A^T + A)x$$

Notice that if A is symmetric, this reduces to

$$\frac{d}{dx}(x^T Ax) = 2Ax$$

3. Linearly Separable Data with Logistic Regression

   Show (or explain) that for a linearly separable data set, the maximum likelihood solution for the logistic regression model is obtained by finding a vector β whose decision boundary $\beta^T x = 0$ separates the classes, and taking the magnitude of β to be infinity.

Solution:

Because the data is linearly separable, it is possible to find a hyperplane with unit normal vector β such that each halfspace induced by this hyperplane contain all samples of one class.

Consider all points on the half space defined by $\beta^T x \geq 0$. Without loss of generality, let's say that all these points come from class 1, while the points such that $\beta^T x < 0$ come from class -1. For some point x_1 in class 1,

$$P(y = 1|x_1) = \mu_i = \frac{1}{1 + exp(-\beta^T x_1)} > 0.5$$

because $\beta^T x_1 \geq 0$. Likewise, for a point x_{-1} in class -1,

$$P(y = -1|x_{-1}) = 1 - P(y = 1|x_{-1}) = 1 - \mu_i > 0.5$$

since $\beta^T x_{-1} < 0$. Now, when we inspect the likelihood of the data, given by

$$L(\beta|D) = \prod_{i=1}^{n} \mu_i^{y_i}(1 - \mu_i)^{1-y_i} = \prod_{i \in w_1} \mu_i \prod_{j \in w_{-1}} (1 - \mu_j)$$

we see that if we take some arbitrary $c > 1$ and scale the unit vector β by c, our likelihood will increase, since all of the individual probabilities in the likelihood will increase. In fact, we can set c = ∞, which will maximize our likelihood. This will render the sigmoid function to be infinitely steep at $\beta^T x_i = 0$ (making it a step function). $P(y = y_i|x_i) = 1$ for all x_i, and the likelihood will be 1. Obviously this is severely overfitting the data, and regularization for this problem would help us avoid that issue.

4. Quadratic Kernel

   Find a feature mapping Φ such that $\Phi(x)^T \Phi(y) = K(x, y)$ where the kernel function is $K(x, y) = (x^T y + 1)^2$. For simplicity, you may assume that the data is 2-dimensional, i.e. $x = [x_1, x_2]^T$.

Solution:

$$\begin{aligned}
K(x, y) &= (x^T y + 1)^2 = (x_1 y_1 + x_2 y_2 + 1)^2 \\
&= 1 + x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 y_1 x_2 y_2 \\
&= 1 + x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 y_1 x_2 y_2 \\
&= [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2, x_2^2][1, \sqrt{2}y_1, \sqrt{2}y_2 \sqrt{2}y_1 y_2, y_1^2, y_2^2]^T \\
&= \Phi(x)^T \Phi(y)
\end{aligned}$$

where

$$\Phi(x) = [1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2]^T$$

5. Fun with Newton's method for root-finding

(a) Write down the iterative update equation of Newton's method for finding a root $x : f(x) = 0$ for a real-valued function f.

Solution: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$

(b) Prove that if $f(x)$ is a quadratic function ($f(x) = ax^2 + bx + c$), then it only takes one iteration of Newton's Method to find the minimum/maximum.

Solution: The Newton's method update for finding a mininum/maximum is

$$x_{n+1} = x_n - \frac{f'(x_n)}{f''(x_n)} = x_n - \frac{2ax_n + b}{2a} = \frac{-b}{2a}$$

And this is the point for mininum/maximum.