Assignment 1

ML Class: CS 65375.501

September 2, 2020

1 Assignment Policies for CS 6375

The following are the policies regarding this assignment.

1. This assignment needs be done individually by everyone.
2. You are expected to work on the assignments on your own. If I find the assignments of a group of (two or more) students very similar, the group will get zero points towards this assignment. You may possibly also be reported to the judiciary committee.
3. Please use Python for writing code. You can submit the code as a Jupyter notebook
4. For the theory questions, please use Latex
5. This Assignment is for 25 points.
6. This will be due on September 18th.

2 Questions

1. **Loss Functions for Linear Regression (6 points):** Assume that the hypothesis function is $f(w, b, x) = w^T x + b$. In the standard linear regression case, given an instance x_i, y_i on the training set, the loss function is defined as $L_i(w, b) = [f(w, b, x_i) - y_i]^2$. Imagine that we are still interested in posing the optimization problem (over the dataset) as:

$$\min_{w,b} \sum_{i=1}^{n} L_i(w, b) \tag{1}$$

What if we were to use some slightly different loss functions? Can you answer if the following loss functions make sense?

(a) $L_i(w, b) = [f(w, b, x_i) - y_i]^3$

(b) $L_i(w, b) = [f(w, b, x_i) - y_i]^4$

(c) $L_i(w, b) = \exp[f(w, b, x_i) - y_i]$

(d) $L_i(w, b) = \max(0, -y_i f(w, b, x_i))$

Part 1: Please answer exactly why these loss functions may or may not be a good choice for regression.
Part 2: Also, compute the gradients of the final loss function.
(Bonus Question): Wherever the loss function makes sense, can you answer how the resulting solution will be different from the standard squared loss? what are the pros and cons of these loss functions compared to the squared loss.

Part 1 is for four points and part two is for two points. The bonus question will be for two additional points.

2. **Loss Functions for Classification (6 Points):** Again, assume that the hypothesis function is $f(w,b,x) = w^T x + b$. In the case of SVM and Perceptrons, we saw the following two loss functions: $L_i(w,b) = \max(0, -y_i f(w,b,x_i))$ for Perceptron and $L_i(w,b) = \max(0, 1 - y_i f(w,b,x_i))$ for Hinge Loss (SVM). Similar to question 1, let us see if the following loss functions are good choices:

   (a) $L_i(w,b) = \max(0, 1 - y_i f(w,b,x_i))^2$

   (b) $L_i(w,b) = [y_i - f(w,b,x_i]^4$

   (c) $L_i(w,b) = \exp[f(w,b,x_i) - y_i]$

   (d) $L_i(w,b) = \exp[-y_i f(w,b,x_i)]$

   Part 1: Please answer exactly why these loss functions may or may not be a good choice for classification.
   Part 2: Also, compute the gradients of the final loss function in each of the cases above.

3. **Polynomial and Higher order Features (3 Points):** Let us use polynomial features with the Perceptron. Consider the dataset shown below. [Hint: The dataset is not separable]. Note that this dataset consists of 2-dimensional points $x = [x_1, x_2]$.

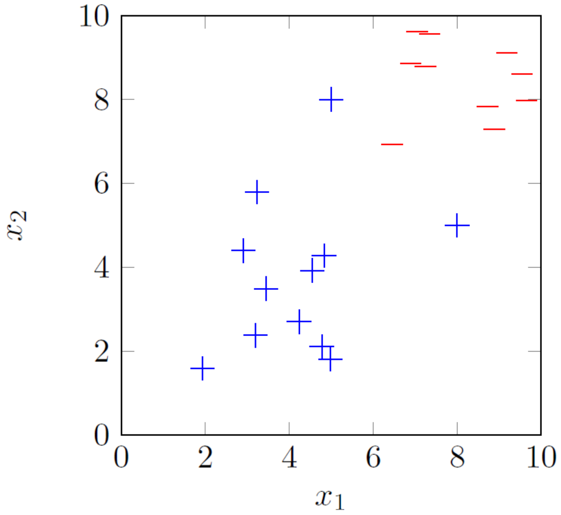

Figure 1: Two Dimensional Data

Part 1: Write down the perceptron loss function with quadratic features. First write down what will be the features, the dimensionality of the expanded (quadratic) feature set and the loss function. Is the dataset linearly separable with quadratic features?

Part 2: Draw approximately the output of the perceptron algorithm on this dataset.

4. **Lagrangian and Dual:** (5 Points)

   Part 1: Let us start with Linear Regression. Recall that the loss function of linear regression is

   $$\min_w \sum_{i=1}^{M} [w^T x^i + b - y^i]^2.$$

   Suppose, we add a constraint that the L2-distance between w and a prior weight vector w_0 is less than k – i.e. suppose we have some reason to believe that the weight vector should be close to w_0. How can you pose this as an optimization problem? What is the Lagrangian formulation? Can you put down the steps involved to get the dual? It may be a little difficult to exactly write down the dual, but even if you can write down the steps involved, you will get the points. Getting the dual exactly will be 2 bonus points.

   Part 2: Let us do this for SVM. For simplicity, let us consider linear separability. Recall that the SVM formulation becomes:

   $$\begin{gathered} min_{w,b}\ \tfrac{1}{2}||w||^2 \\ \text{subject to: } y_i(w^T x_i + b) \geq 1, \forall i = 1, \cdots, M \end{gathered} \tag{2}$$

   Similar to part 1, what if we add the constraint that the L2-distance between w and a prior weight vector w_0 is less than k? How can you pose this as an optimization problem? What is the Lagrangian formulation? Obtain the dual solution from the Lagrangian.

5. **Implement Linear Regression from Scratch (5 Points):** Implement a Linear Regression from scratch. Implement a loss function (squared loss) and implement a very simple gradient descent algorithm in python. Test your implementation on the house price prediction dataset from the demo on Linear Regression. Finally, compare your solution to the solution obtained by sklearn (from the demo) and comment on it briefly. You do not need to consider the more complicated quadratic case. Just the simple linear regression on the features is enough for this.