ORIE 4740 Problem Set I

Out: Midnight Friday January 23, 2015

Due: Noon Monday February 2, 2015

Rules (read carefully)

- This homework must be individual work. You may not discuss your solution with other students in the class. Concepts and material related to the homework may be discussed.
- All work must be shown for full credit
- Any plots must be scaled such that the main features of the graph are displayed. Scale your axes thoughtfully, label axes and display a legend.
- Relevant portions of your code (i.e. code snippets that solve the problem as opposed to code that imports a bunch of libraries) must be included inline with your solution. Any problem that requires an implementation implicitly requires including the code as part of the submission.
- You can use any programming language of your choosing.
- Late assignments lose 10% of the grade per day until solutions are released. This will typically occur by Midnight of the following Wednesday. Late assignments are due in the same dropbox location.

Problem 1 [25]

Consider the sum of squares error function that we looked at in class with only a single parameter w (slope):

$$E(w) = \sum_{n=1}^{N} (y_n - wx_n)^2$$

For this and the remaining problems, we will consider a small dataset $D = \{(x_n, y_n)\}$ given in Table 1. All of the questions below assume that we are "training" (or fitting) our linear model using all of the example pairs (x_n, y_n) in D.

n	x_n	y_n
1	1	1
2	2	2
3	3	4
4	4	3
5	5	30

Table 1: Dataset D

1. **What does the error function look like?**

   Plot $E(w)$ as a function of the parameter we are interested in learning, using all of the data given in your dataset D (please review the guidelines above for how to present your plots).

2. **What is the optimal parameter?**

   Find the optimal w analytically (i.e. in closed form). Optimal w refers to the value of w that minimizes $E(w)$ on the complete dataset D. Show all work.

Problem 2 [25]

Often with big data, solving for optimal parameters analytically is inefficient. A simple, flexible and efficient alternative is to perform a technique known as **gradient descent**. The idea is simple and can be summarized as follows:

- Initialize the parameters you are interested in learning to a random value.
- Compute a derivative of the error function with respect to the parameter(s) of interest, i.e. $\frac{dE(w)}{dw}$.
- Update the parameter(s) as follows $w_{updated} = w_{old} - \gamma \frac{dE(w_{old})}{dw}$. γ is an arbitrarily small step-size. Set $\gamma = 0.01$ for the problems in this assignment.

Implement the **gradient descent** procedure to learn the optimal value of w with the data and the error function in problem 1. Confirm that you obtain the same value that you found analytically (refer to the guidelines above regarding expected code submission).

Problem 3 [25]

One advantage of **gradient descent** is that you can find optimal parameters for error functions where this can't be done analytically, giving you freedom to design these functions to suit your application. Consider for example the following modification to the error function in Problem 1.

$$E(w) = \sum_{n=1}^{N} |y_n - wx_n|$$

1. **What does the error function look like?**

   Repeat Problem 1 part 1 with the error function given above.

2. **What is the optimal parameter?**

   Unlike in Problem 1, there is no closed-form solution for the error function given in this problem. Implement the **gradient descent** procedure described in Problem 2 to find the optimal value of the parameter w given the data in D. Note that the derivative of $|x|$ is undefined when $x = 0$. As this corresponds to the case when when the prediction is exactly equal to the true value (verify this for yourself), it's convenient to define the derivative to be equal to 0 in this case.

3. **How do the lines of fit compare?**

   Plot the optimal lines of fit overlaying the original data in D (Table 1) corresponding to the two error functions (Problem 1 and Problem 3). Summarize one striking difference between the resulting lines of fit. What about the error functions accounts for this difference?

Problem 4 [25]

Suppose that we want to penalize over-estimation twice as much in comparison to under-estimation, i.e. an error of the same magnitude costs twice as much when it's above the true value than when below.

1. Consider now a general form for the error functions considered in Problems 1-3:

   $$E(w) = \sum_{n=1}^{N} l(wx_n, y_n)$$

   where $l(\hat{y}, y)$ is often referred to as the **loss function**. In Problem 1, for example, $l(wx_n, y_n) = (y_n - wx_n)^2$.

   For this problem, design a loss function $l(wx_n, y_n)$ that satisfies the requirement of this problem (stated at the beginning of the problem).

2. **What does the error function look like?**

   Repeat Problem 1 part 1 using the loss function that you just designed.

3. **What is the optimal parameter?**

   As in Problem 3, it's unlikely that the loss function that you designed has a nice closed-form solution. Implement the **gradient descent** procedure described in Problem 2 to find the optimal value of the parameter w for the data in D.

4. Plot the line of fit (overlaying the data D) corresponding to the optimal w that you just found. Experiment with different penalty ratios between overshooting and undershooting errors. What effect does does this asymmetric penalty in the error function have on the resulting line of best fit?