CS6923 Machine Learning, Spring 2019
Prof. Linda Sellie, NYU School of Engineering

**Homework 5**[1]

**Submit on NYU Classes by Thurs. Oct. 10 at 8:00 p.m.** You may work together with one other person on this homework. If you do that, hand in JUST ONE homework for the two of you, with both of your names on it. You may *discuss* this homework with other students but YOU MAY NOT SHARE WRITTEN ANSWERS OR CODE WITH ANYONE BUT YOUR PARTNER.

**IMPORTANT SUBMISSION INSTRUCTIONS:** Please submit your solutions in 3 separate files: one file for your written answers to Part I, one file for your written answers/output for the questions in Part II, and one file with your code (a zip file if your code requires more than one file).

# Part I: Written Exercises

1. Suppose we are training a logistic classifier to solve a binary classification problem (i.e. we are performing logistic regression). The classifier corresponds to a function of the form $h(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}}$ whose output is an estimate of the probability that $\mathbf{x}$ belongs to class 1.

   Suppose while performing gradient ascent, the weights become $\mathbf{w}^T = [0.66, -2.24, -0.18]$.

   The table below shows the result of using these weights to predict the class on the training examples.

| | $x_1$ | $x_2$ | $h(\mathbf{x})$ | y |
|---|---|---|---|---|
| 1 | 0.49 | 0.09 | 0.389 | 0 |
| 2 | 1.69 | 0.04 | 0.042 | 0 |
| 3 | 0.04 | 0.64 | 0.613 | 0 |
| 4 | 1. | 0.16 | 0.167 | 0 |
| 5 | 0.16 | 0.09 | 0.572 | 1 |
| 6 | 0.25 | 0. | 0.526 | 1 |
| 7 | 0.49 | 0. | 0.393 | 1 |
| 8 | 0.04 | 0.01 | 0.638 | 1 |

   (a) For a decision boundary of 0.5, create the confusion matrix.

   (b) Plot the points on a graph and draw the decision boundary (I would suggest using some sort of plotting library and a image editor) https://www.devtalking.com/articles/machine-learning-11/

   (c) For the data set above what is the FPR?

   (d) For the data set above what is the TPR?

   (e) What is the accuracy?

   (f) What is the recall?

   (g) What is the precision?

   (h) In logistic regression, we are trying to maximize the log likelihood

$$\ell(\mathbf{w}) = \sum_{i=1}^{N} y^{(i)} \ln(h(\mathbf{x}) + (1 - y^{(i)})(1 - h(\mathbf{x})))$$

   which is the same as minimizing the error function

$$-\left(\sum_{i=1}^{N} y^{(i)} \ln(h(\mathbf{x}) + (1 - y^{(i)})(1 - h(\mathbf{x})))\right)$$

[1] Some of these are modified from Prof. Rangan's questions.

This quantity is sometimes called the *cross-entropy* of the classifier on the dataset. Using the initial weights, what is the cross-entropy of the classifier on the given training set when the threshold $t = 0.5$?

(i) Given $\mathbf{w}$ as described above and $\mathbf{w}' = (1.33, -2.96, -2.77)^T$, which is more likely to to have generated the dataset given above.

(j) Perform one step of gradient ascent using the $\mathbf{w}$ given above.[2]

(k) How did the data points near the decision boundary contribute to the new value of $\mathbf{w}$?

(l) How did the data points which were correctly classified and far away from the decision boundary contribute to the new value of $\mathbf{w}$?

(m) How did incorrectly classified points contribute to the new value of $\mathbf{w}$?

(n) Using the updated weights, what is the cross-entropy (error) of the classifier on the given training set?

(o) Did the cross-entropy (error) go up or down after one iteration of the gradient ascent (or descent)? Is this what you expected? Why or why not?

2. When using gradient ascent/descent to minimize an error function, there are common problems that people encounter. For each of the following problems, explain why it might be happening, and suggest a way to fix the problem.

   (a) The error doesn't decrease steadily with the number of iterations; it sometimes goes up, and it sometimes goes down. Also, the weights don't converge.

   (b) The error decreases steadily, but very slowly. Even after $1,000,000$ iterations, the error is still not much smaller than it was at the beginning.

   (c) The error decreases and it converges to a single value, but that value is large.

3. A data scientist is hired by a political candidate to predict who will donate money. The data scientist decides to use two predictors for each possible donor:

   - $x_1$ = the income of the person(in thousands of dollars), and
   - $x_2$ = the number of websites with similar political views as the candidate the person follows on Facebook.

   To train the model, the scientist tries to solicit donations from a randomly selected subset of people and records who donates or not. She obtains the following data:

| Income (thousands \$), $x_1^{(i)}$ | 30 | 50 | 70 | 80 | 100 |
|---|---|---|---|---|---|
| Num websites, $x_2^{(i)}$ | 0 | 1 | 1 | 2 | 1 |
| Donate (1=yes or 0=no), $y^{(i)}$ | 0 | 1 | 0 | 1 | 1 |

   (a) Draw a scatter plot of the data labeling the two classes with different markers.

   (b) Find a linear classifier that makes at most one error on the training data. The classifier should be of the form,

$$\hat{y}_i = \begin{cases} 1 & \text{if } z^{(i)} > 0 \\ 0 & \text{if } z^{(i)} < 0, \end{cases} \qquad z^{(i)} = \mathbf{w}^{\mathsf{T}}\mathbf{x}^{(i)}$$

   What is the weight vector $\mathbf{w}$ of your classifier?

[2] You do not need to perform this by hand - but make sure you can perform this by hand.

(c) Now consider a logistic model of the form,

$$P(y^{(i)} = 1|\mathbf{x}^{(i)}) = \frac{1}{1 + e^{-z^{(i)}}}, \quad z^{(i)} = \mathbf{w}^{\mathsf{T}}\mathbf{x}^{(i)}$$

Using $\mathbf{w}$ from the previous part, which sample $i$ is the *least* likely (i.e. $P(y^{(i)}|\mathbf{x}^{(i)})$ is the smallest). If you do the calculations correctly, you should not need a calculator.

(d) Now consider a new set of parameters

$$\mathbf{w}' = \alpha\mathbf{w},$$

where $\alpha > 0$ is a positive scalar. Would using the new parameters change the values $\hat{y}$ in part (b)? Would they change the likelihoods $P(y_i|\mathbf{x}_i)$ in part (c)? If they do not change, state why. If they do change, qualitatively describe the change as a function of $\alpha$.

4. (Do not turn in this question) How does the logistic function change when $w_0$ changes? You can just run some simulations and describe what you notice. (Or state mathematically what happens)

5. (Do not turn in this question) How does the logistic function change if you use $\mathbf{w}' = 2\mathbf{w}$ instead of $\mathbf{w}$? You can just run some simulations and describe what you notice. (Or state mathematically what happens)

6. (Do not turn in this question) Regularization:

   - Add lasso regularization to the log likelihood function for logistic regression
   - Add ridge regularization to the log likelihood function for logistic regression
   - Determine the derivative of log likelihood function for logistic regression with ridge regularization.

# Part II: Programming Exercise

Implement a logistic regression classifier using gradient ascent and apply it to a two-class classification problem to detect breast cancer. Use the gradient ascent algorithm given in the lecture notes.[3]

The initial hyper-paramters for this assignment are:

- *threshold* = 0.5
- *learning rate* = 0.5
- number of iterations to run your algorithm = 5000

After running your logistic regression classifier, report the following:

- the values of the coefficient vector **w**
- for the test data set, determine the
  - precision
  - recall
  - f1 score
  - confusion matrix
- plot the value of the log-likelihood (the objective function) on every 100 iteration's of your gradient ascent, with the iteration number on the horizontal axis and the objective value on the vertical axis.

  The code to plot the objective function is already written for you in the Jupyter notebook
- use the test set as a validation set and see if you can find a better setting of the hyper-parameters. Report the best values you found. (The goal is for you to experiment and see the result of different values - you do not need to find the best values.)

[3]Before the first iteration, the coefficient weights are initialized to 0.