ECE 2372 - Homework 2 on Logistic Regression

These homework is on the implementation of logistic regression model with different algorithms. You can refer to our Lecture3.pdf for additional information on the method. Also section 4.4 from ELSII text is a good background reading for this homework.

Please upload your solutions including your implementation code to Canvas by February 8, 2024

Problem 1: Gradient Descent

Let's implement Logistic Regression and use gradient descent algorithm to perform the maximum likelihood estimate of our parameters. For the step size I would like you to experiment with different values, this is a common approach in deciding step size. Explicitly start somewhere $\alpha \approx 1$ and work your way down to α values well below that $\alpha \ll 1$ (may by cutting in half or tenth), also play around with your stopping criteria.

a. Implement gradient descent algorithm for this problem.

b. Test your code on the four given synthetic data sets. Report the value of step size you use and the iterations required for each dataset. Please plot classifiers your implementation returns for each data set. I would like you to also compare your results with the results you obtained with LDA in the first assignment.

c. Did you notice any speed difference in the convergence among these synthetic data sets? If so, why would you think this might of happened?

Problem 2: Newton's Method Now let's try out Newton's method as a solver to MLE step in logistic regression.

a. Implement Newton's method for this section. Note that I calculated the Hessian below, (you can verify if you want to but not need to):

$$\frac{\nabla^2 l(\theta)}{\partial \theta} = -\sum_{i=1}^{n} \tilde{x}_i \tilde{x}_i^T g(\theta^T \tilde{x}_i)(1 - g(\theta^T \tilde{x}_i))$$

b. Test your code on the four given synthetic data sets. Report the number of iterations required for each dataset and compare it with the results you obtained from the first problem. Plot the classifier (should be same with the problem 1).

Problem 3: Stochastic Gradient Descent

Lastly let's implement stochastic gradient descent to solve the maximum likelihood estimation step in the logistic regression. In regular gradient descent, in order to compute the gradient we need to comput the following sum

$$\sum_{i=1}^{n} \tilde{x}_i \, (y_i - g(\theta^T \tilde{x}_i))$$

This sum gets challenging as n gets large (and this is not a problem in the data-sets we are using here). To tackle the computational burden with increasing n, we can select one $\tilde{x}_i$ at random and compute $\tilde{x}_i \, (y_i - g(\theta^T \tilde{x}_i))$ as a rough approximation to the sum. This method is known as stochastic gradient descent and in general results in more iterations while enabling to do each iteration cheaper and can help when n is very large.

Implement stochastic gradient version of Logistic regression. Test your implementation with the same data sets. Please write down the number of iterations required for each datasets and compare the results with the two previous versions. Include a copy of figures depicts the resulting classifiers.