CSE6250: Homework 2

Shaanjot Gill

February 8, 2019

1 Logistic Regression

1.1 Batch Gradient Descent

a. Derive the gradient of the negative log-likelihood in terms of **w** for this setting.

Calculate the partial derivative of the negative log-likelihood wrt **w**

$$\frac{\partial}{\partial \mathbf{w}} NLL(D, \mathbf{w}) = \left(y \frac{1}{\sigma(t)} - (1-y) \frac{1}{1-\sigma(t)} \right) \frac{\partial}{\partial \mathbf{w}} \sigma(t) \tag{1}$$

Calculate the partial derivative of the sigmoid function wrt t

$$\frac{\partial}{\partial t} \sigma(t) = \sigma(t) \big(1 - \sigma(t)\big) \tag{2}$$

Substitute (2) into (1)

$$\frac{\partial}{\partial \mathbf{w}} NLL(D, \mathbf{w}) = \left(y \frac{1}{\sigma(t)} - (1-y) \frac{1}{1-\sigma(t)} \right) \frac{\partial}{\partial \mathbf{w}} t \cdot \sigma(t) \big(1 - \sigma(t)\big) \tag{3}$$

Simplifying

$$\big(y - \sigma(t)\big) \frac{\partial}{\partial \mathbf{w}} t \tag{4}$$

1.2 Stochastic Gradient Descent

a. Show the log likelihood, l, of a single $(\mathbf{x}_t, y_t)$ pair.

$$l(D_t, \mathbf{w}) = (1 - y_t) \log(1 - \sigma(\mathbf{w^T x_t})) + y_t \cdot \log \sigma(\mathbf{w^T x_t}) \tag{5}$$

b. Show how to update the coefficient vector $\mathbf{w}_t$ when you get a patient feature vector $\mathbf{x}_t$ and physician feedback label y_t at time t using $\mathbf{w}_{t-1}$ (assume learning rate η is given).

Using (4) and recalling $t = \mathbf{w^T} x_t$

$$\mathbf{w_t} = \mathbf{w_{t-1}} + \eta\big(y_t - \sigma(\mathbf{w^T} x_t)\big)x_t \tag{6}$$

c. What is the time complexity of the update rule from **b** if $\mathbf{x}_t$ is very sparse?

$$O(N * mean(\text{no. of non-zero features}))$$

d. Briefly explain the consequence of using a very large η and very small η.

Very large η has the risk of overshooting the minima, while a very small η will converge extremely slowly.

e. Show how to update $\mathbf{w}_t$ under the penalty of L2 norm regularization. In other words, update $\mathbf{w}_t$ according to $l - \mu\|\mathbf{w}\|_2^2$, where μ is a constant. What's the time complexity?

$$\mathbf{w_t} = \mathbf{w_{t-1}} + \Bigg(\eta\big(y_t - \sigma(\mathbf{w^T} x_t) x_t\big) - \eta\mu\mathbf{w^T}\Bigg) \tag{7}$$

time complexity is $O(N)$

2 Programming

2.1 Descriptive Statistics

b. Use *events.csv* and *mortality.csv* provided in **data** as input and fill Table 1 with actual values. We only need the top 5 codes for common diagnoses, labs and medications. Their respective counts are not required.

Metric	Alive patients	Deceased patients
Event Count		
1. Average Event Count	1029.059	682.647
2. Max Event Count	16829	12627
3. Min Event Count	2	1
Encounter Count		
1. Average Encounter Count	24.861	18.669
2. Max Encounter Count	375	391
3. Min Encounter Count	1	1
Record Length		
1. Average Record Length	151.397	194.65
2. Max Record Length	2601	3103
3. Min Record Length	0	0
Common Diagnosis	DIAG320128	DIAG320128
Common Laboratory Test	LAB3009542	LAB3009542
Common Medication	DRUG19095164	DRUG19095164

Table 1: Descriptive statistics for alive and dead patients

2.2 SGD Logistic Regression

b. Show the ROC curve generated by test.py in this writing report for different learning rates η and regularization parameters μ combination and briefly explain the result.

Figures 1, 2, and 3 show ROC curves for SGD Logistic Regression. In my implementation, I noted that the learning rate had the largest effect, with a negative influence for any value above 0.1. The regularization parameter had little effect and never seemed to improve the test performance, but it did slightly decrease test performance when it was set very high.

I expect μ to have a larger effect, and maybe improve test performance.

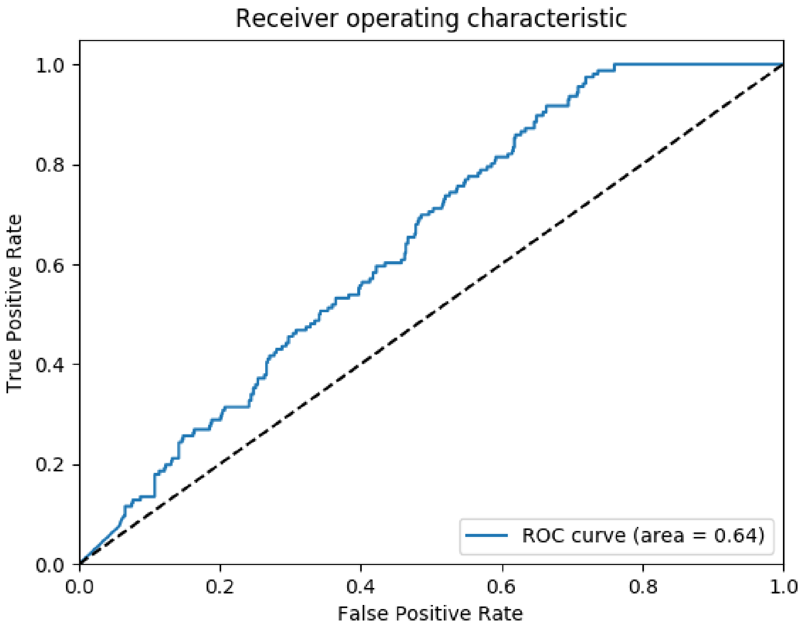

Figure 1: $\eta = 0.01$, $\mu = 0.0$

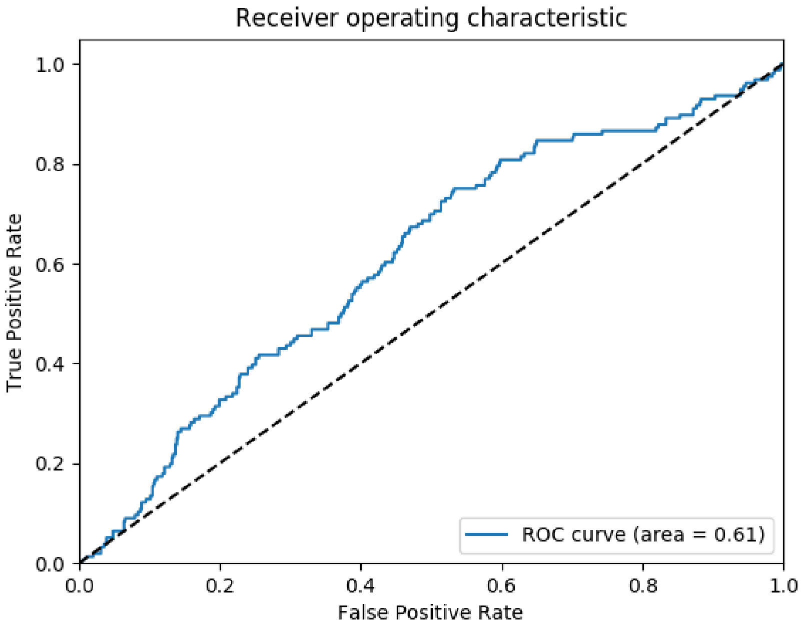

Figure 2: $\eta = 0.10$, $\mu = 0.1$

2.3 Hadoop

c. Compare the performance with that of previous problem and briefly analyze why the difference.

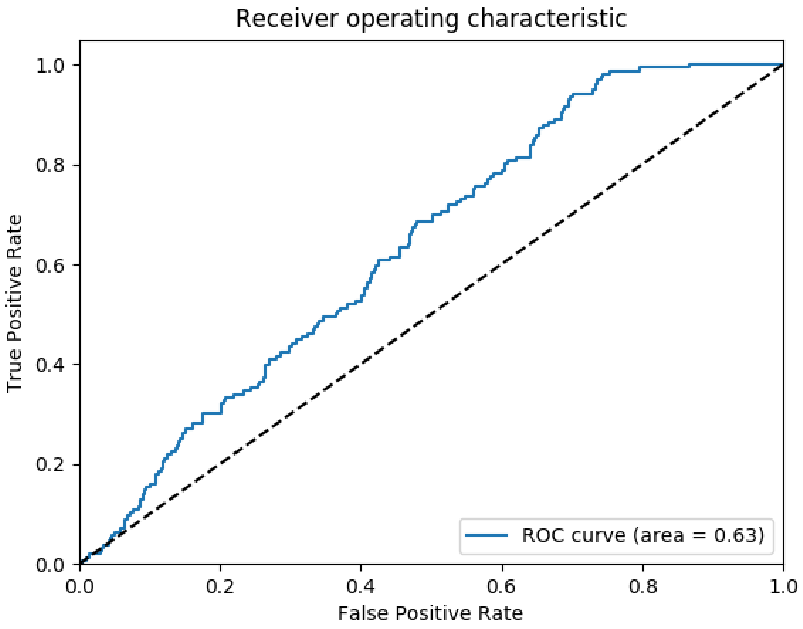

Figure 3: $\eta = 0.01$, $\mu = 0.1$

My ensemble learner's performance was not much different than single learner, but it was much less sensitive to an increase in learning rate. These results are encouraging as we can use tools like Hadoop to train many small models in parallel and achieve similar results with an overall more robust model.