Homework 2 Write-up

N. Nauman

February 3, 2019

1.1)

$$J(D, \mathbf{w}) = -\sum_{i=1}^{N}[(1-y_i)log(1-\sigma(\mathbf{w}^\intercal \mathbf{x}^{(i)}) + y_i log(\sigma(\mathbf{w}^\intercal \mathbf{x}^{(i)}))]$$

$$\frac{\partial J}{\partial w_j} = -\sum_{i=1}^{N}[(1-y_i)\frac{\partial}{\partial w_j}log(1-\sigma(\mathbf{w}^\intercal \mathbf{x}^{(i)}) + y_i \frac{\partial}{\partial w_j} log(\sigma(\mathbf{w}^\intercal \mathbf{x}^{(i)}))]$$

$$Note: \frac{\partial \sigma(\mathbf{w}^\intercal \mathbf{x}^{(i)})}{\partial w_j} = \sigma(\mathbf{w}^\intercal \mathbf{x}^{(i)})(1-\sigma(\mathbf{w}^\intercal \mathbf{x}^{(i)}))x_j^{(i)}$$

$$\Rightarrow \frac{\partial J}{\partial w_j} = -\sum_{i=1}^{N}[(y_i - 1)\sigma(\mathbf{w}^\intercal \mathbf{x}^{(i)}) + y_i(1-\sigma(\mathbf{w}^\intercal \mathbf{x}^{(i)}))]x_j^{(i)}$$

$$= \boxed{-\sum_{i=1}^{N}(y_i - \sigma(\mathbf{w}^\intercal \mathbf{x}^{(i)})x_j^{(i)}}$$

1.2)

a)

$$LL = -NLL = (1-y_t)log(1-\sigma(\mathbf{w}^\intercal \mathbf{x}^{(t)}) + y_t log(\sigma(\mathbf{w}^\intercal \mathbf{x}^{(t)}))$$

b)

$$\mathbf{w}^{(t)} = \mathbf{w}^{(t-1)} - \eta(\sigma(\mathbf{w}^{(t-1)\intercal}\mathbf{x}^{(t)}) - y_t)x^{(t)}$$

c)

First the dot product within the sigmoid is O(n) + O(n) = O(n). But since x is sparse, this operation is actually only O(n-c) where c is the number zero elements in x. Taking the sigmoid and subtacting by y are O(1) operations. Multiplying the input by a scalar is again O(n-c). Likewise, the subtraction of the weight matrix with the sparse 2nd term in the equation. So the overall complexity is: O(n-c) where c is the number of zero features in the input vector.

d)

Very large learning rate results in SGD making large jumps at each iteration. If these jumps are large enough, SGD will diverge and not settle to a minima. Very small learning rate will result in the the algorithm taking a very long time to converge to the minima as SGD will only make baby steps at each iteration towards the minima.

e)

$$\mathbf{w} := \mathbf{w} - \eta[(\sigma(\mathbf{w}^\intercal \mathbf{x}^{(t)}) - y_t)x^{(t)} - \mu\|\mathbf{w}\|_2^2]$$

The complexity is now O(n) since we are not making the assumption that the weight matrix is sparse (n represents the number of features).

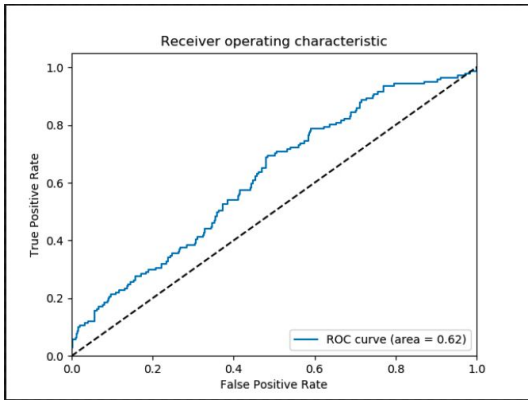
2.1 Descriptive Statistics

Metric	Deceased patients	Alive patients
Event Count		
1. Average Event Count	1027.7385229540919	683.1552587646077
2. Max Event Count	16829	12627
3. Min Event Count	2	1
Encounter Count		
1. Average Encounter Count	24.839321357285428	18.695492487479132
2. Max Encounter Count	375	391
3. Min Encounter Count	1	1
Record Length		
1. Average Record Length	157.04191616766468	194.70283806343906
2. Median Record Length	25	16
3. Max Record Length	5364	3103
4. Min Record Length	0	0
Common Diagnosis	DIAG320128 416 DIAG319835 413 DIAG313217 377 DIAG197320 346 DIAG132797 297	DIAG320128 1018 DIAG319835 721 DIAG317576 719 DIAG42872402 674 DIAG313217 641
Common Laboratory Test	LAB3009542 32765 LAB3023103 28395 LAB3000963 28308 LAB3018572 27383 LAB3016723 27060	LAB3009542 66937 LAB3000963 57751 LAB3023103 57022 LAB3018572 54721 LAB3007461 53560
Common Medication	DRUG19095164 6396 DRUG43012825 5451 DRUG19049105 4326 DRUG956874 3962 DRUG19122121 3910	DRUG19095164 12468 DRUG43012825 10389 DRUG19049105 9351 DRUG19122121 7586 DRUG956874 7301

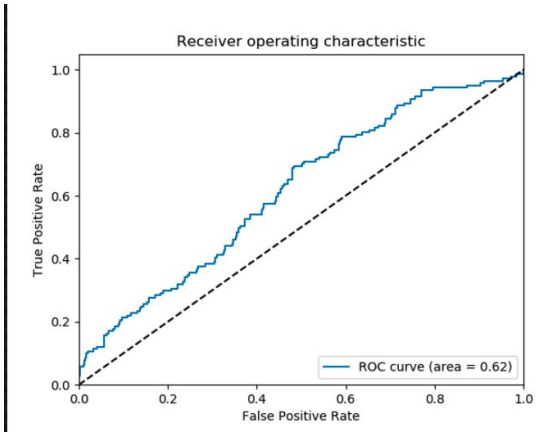
2.3)

b)

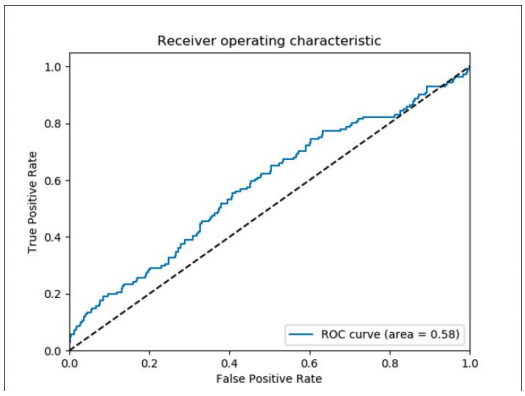
Default (eta=0.01, C=0)



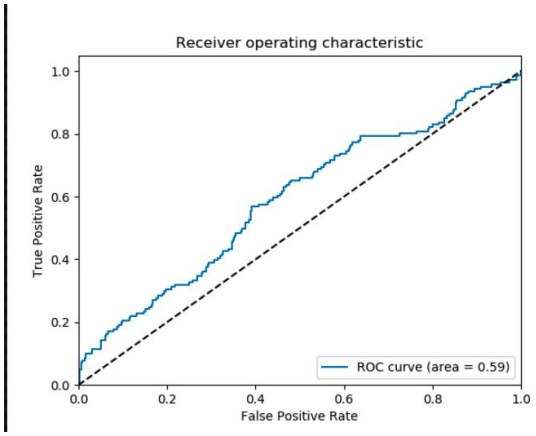
(eta= 0.01, C=0.001)



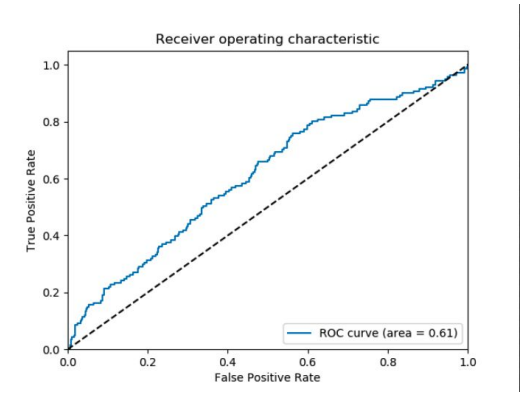
(eta = 0.001, C=0)



(eta = 0.01, C=1)



(eta = 0.5, C=0)

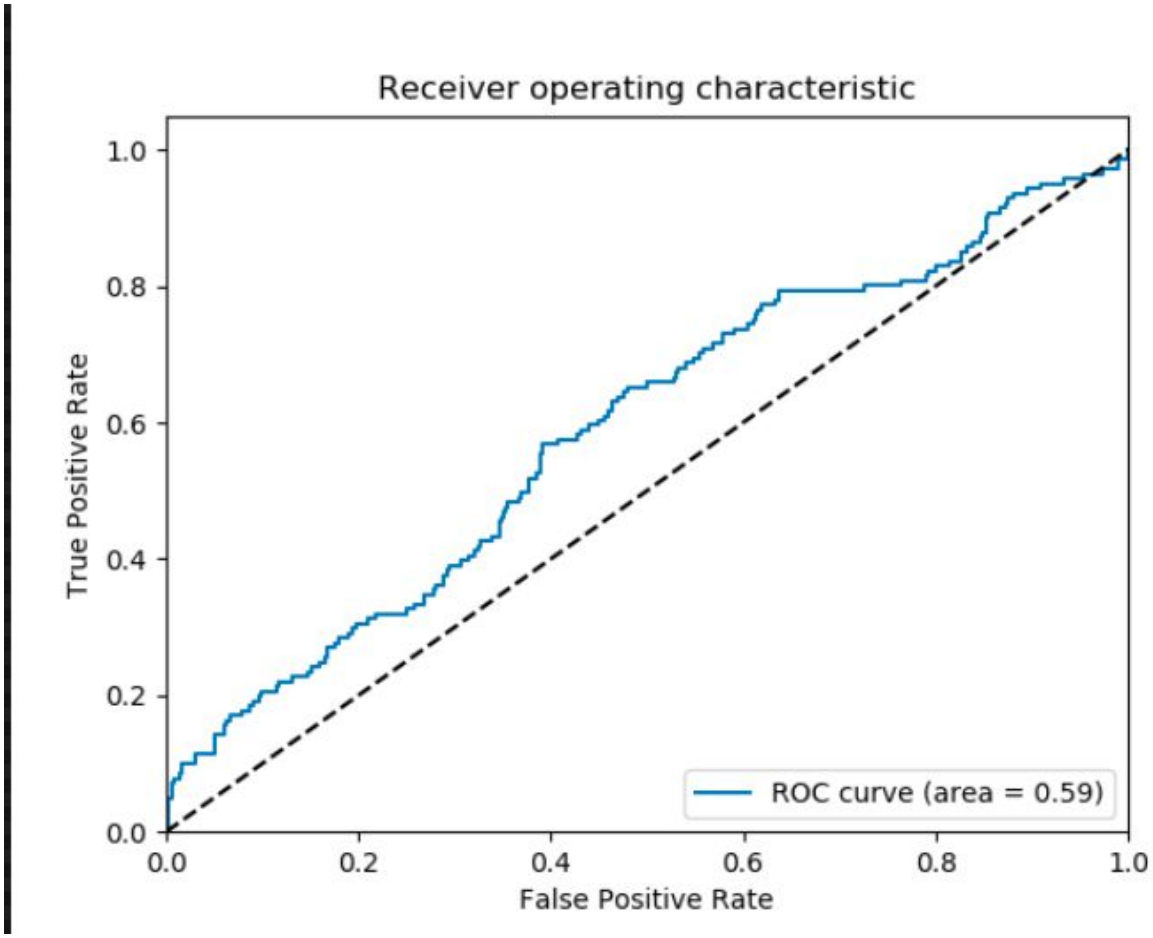


We see that the default parameters give us the best results. This means that the model is not overfitting hence regularization is not helping (in fact it is making the model worse so perhaps we are under fitting the data and could use a model of higher complexity). Eta is also about right; increasing or decreasing it from the default value degrades performance as illustrated by the ROC curve.

2.4)

c)

Ensemble (n=5, r=0.4)



We see that the ensemble actually achieves worse performance than the when we use a single model. The reason this is happening might be because our ratio is too small and the models aren't trained on enough data hence overfitting the training set and leading to poor generalization. But we can't make the ratio too large or else all the models will be too similar and therefore making ensembling pointless. Another solution would be too increase the number of models in the ensemble as this will reduce variance and lead to better generalization. We could also try adjusting the actual classifier parameters and ideally, try implementing models other than lr so we're ensembling sufficiently different models which will also result in less variance (and this generalize better to the testing set).