COMPM041

Zunran Guo

March 2016

1 Introduction

In this project, linear regressions and logistic regressions are trained by stochastic gradient descent (SGD) and batch gradient descent (BGD) to classify spam emails. The spam email data set used for training is retrieved from the UCI Machine Learning Repository. This data set contains 4601 data points, each of which has 57 attributes and the spam/non-spam label has been assigned.

The code snippet below is an example data point. The first 57 entries are attribute information of an email (frequency of certain words, frequency of certain characters, average length of uninterrupted sequences of capital letters, etc.) and the last entry is the label that denotes whether the email is spam (1) or not (0). More details can be found in the documentation contained in the data set package downloaded.

```
# example data point
0,0.64,0.64,0,0.32,0,0.64,0,0,0,0.32,0,1.29,...,1.93,0,0.96,0,0,0,0,0.778,0,0,3.756,61,278,1
```

2 Partition and normalize the data set

For this project, the programming language used is Python and the numpy package is leveraged widely given its capability of fast matrix computation. The data set is first partitioned to 10 folds for the later cross-validation. i.e. Group 1 will consist of points 1,11,21,...; Group 2 will consist of points 2,12,22,...; ... and Group 10 will consist of points 10,20,30,.... Fold k (k = 1, 2, ... 10) will consist of testing on Group k with the model obtained by training remaining 9 groups.

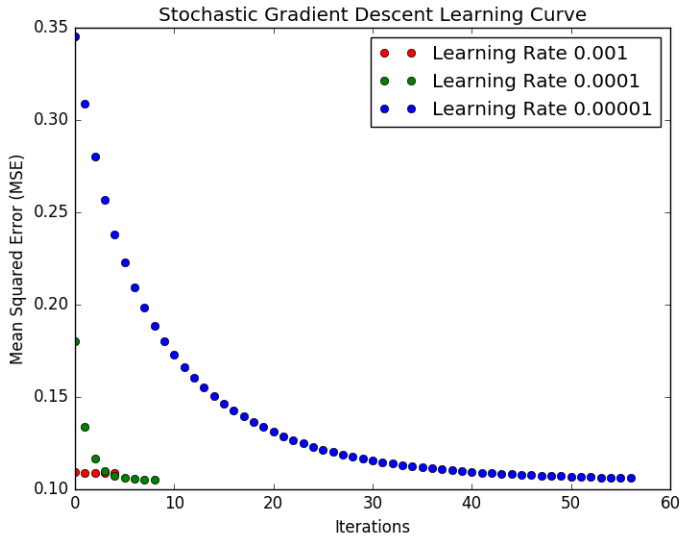
The attributes in all data points are also normalized to their z-scores, i.e. the distances to the attributes means divided by the corresponding standard deviations. For example, say Attribute 1 has a mean of 0.10455 and a standard deviation of 0.30536 across the entire data set. If a particular data point has an Attribute 1 value of 0.5, the corresponding z-score of Attribute 1 in that data point would be (0.5 - 0.10455)/0.30536 = 1.295.

3 Linear stochastic gradient descent result

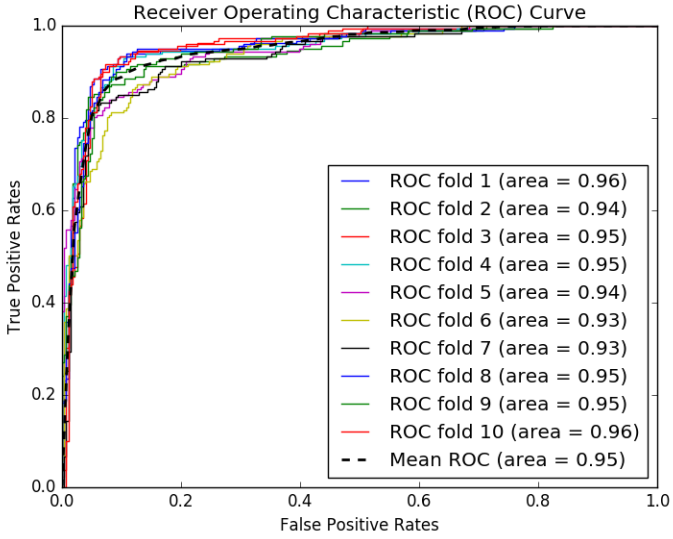
```
repeat until convergence {
    for i = 1 to m, {
        θ_j := θ_j + α(y^(i) − h_θ(x^(i)))x_j^(i) (for every j)
    }
}
```

Convergence criterion: the difference of Mean Squared Error (MSE) smaller than 0.0001

The learning curves are plotted below. Learning rate greater than 0.001 will lead to divergence.



The Receiver Operating Characteristic (ROC) curves are plotted below at the most efficient learning rate.

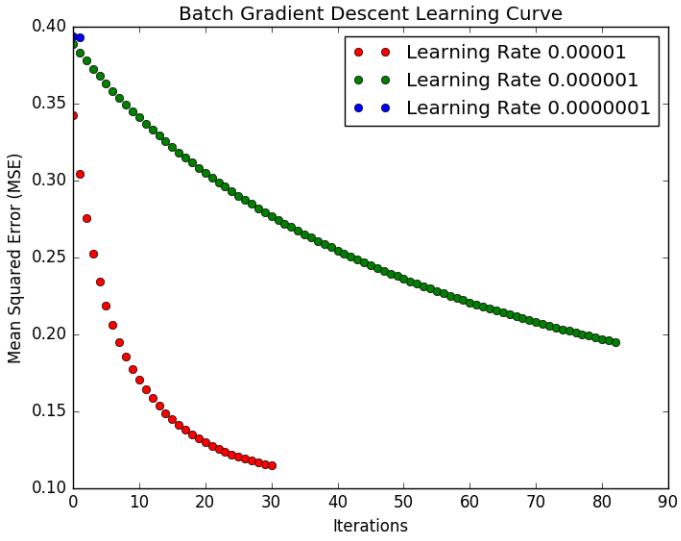


4 Linear batch gradient descent result

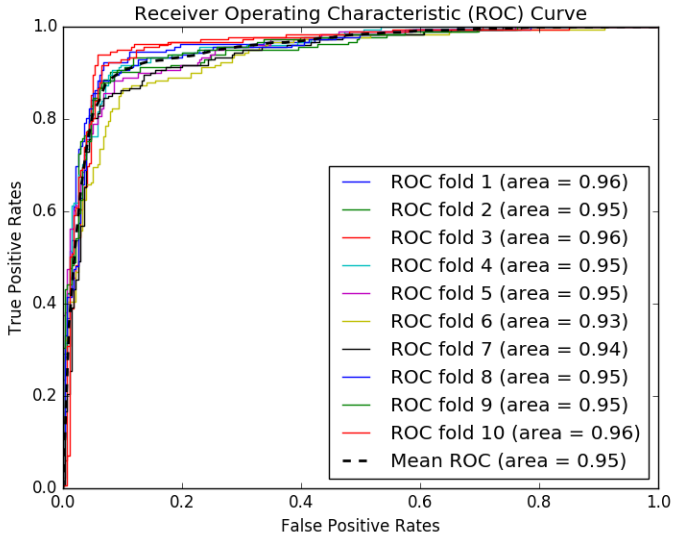
```
repeat until convergence {
    for i = 1 to m, {
        θ_j := θ_j + α Σ_{i=1}^{m} (y^(i) − h_θ(x^(i))) x_j^(i)  (for every j)
    }
}
```

Convergence criterion: the difference of Mean Squared Error (MSE) smaller than 0.0001

The learning curves are plotted below. Learning rate greater than 0.00001 will lead to divergence.



The Receiver Operating Characteristic (ROC) curves are plotted below at the most efficient learning rate.



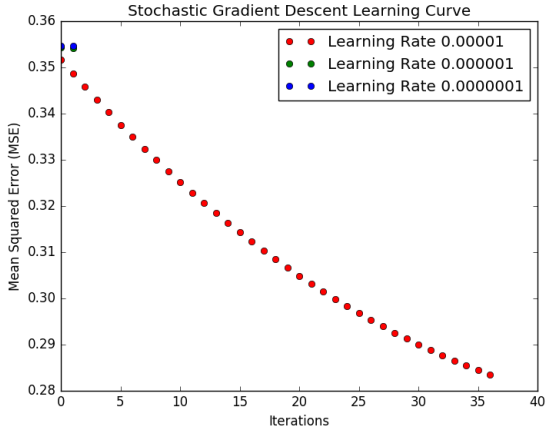
I have also plotted the ROC curves for each fold just as comparisons with the mean ROC curve.

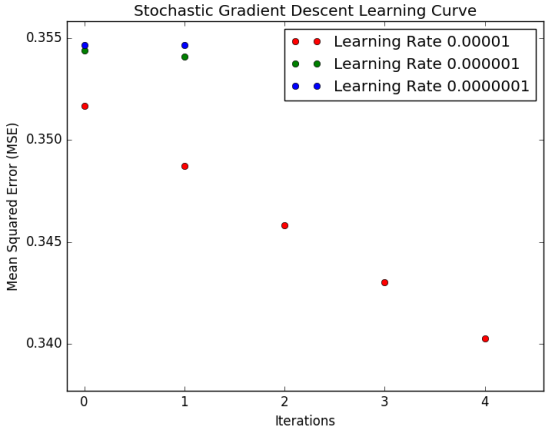
5 Logistic stochastic gradient descent result

Convergence criterion: the difference of Mean Squared Error (MSE) smaller than 0.001

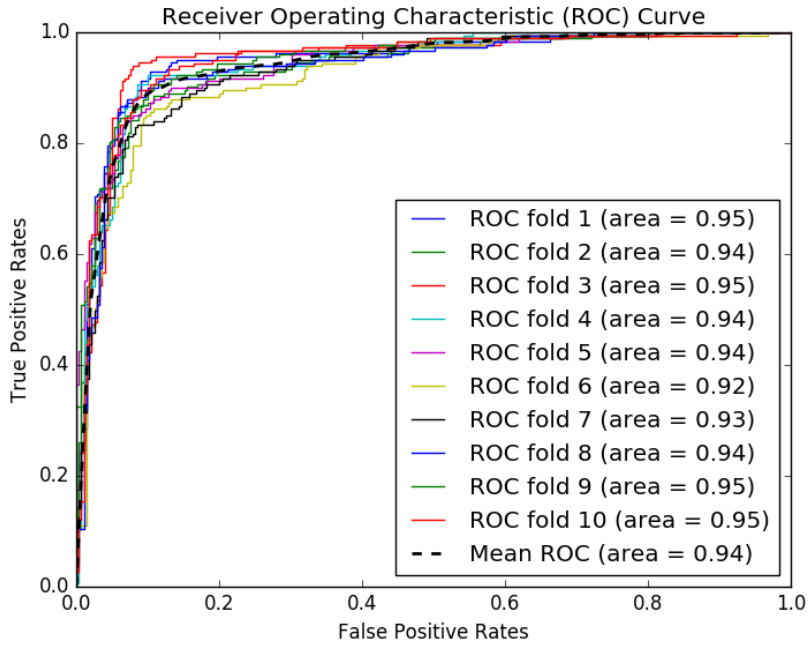
The learning curves are plotted below. A enlarged version is placed to the right of the original

version. Learning rate greater than 0.001 will lead to divergence.





The Receiver Operating Characteristic (ROC) curves are plotted below at the most efficient learning rate.

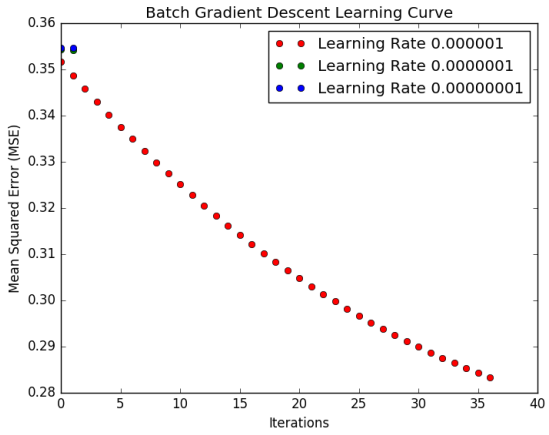


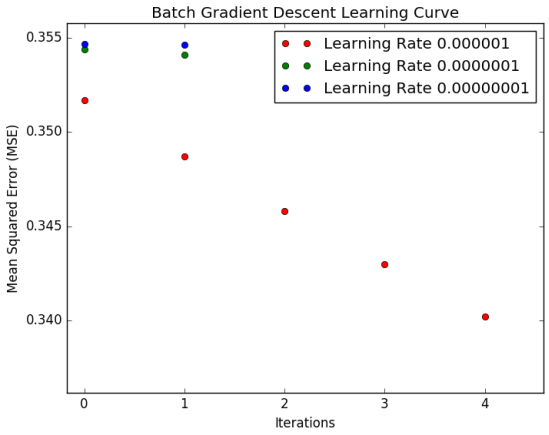
Note: The above ROC curves are obtained at a convergence criterion of 0.001. If the convergence criterion is reset to 0.0001, MSEs of higher learning rates could be reduced very close to the one we obtained from lower learning rates. However the iterations required for each fold would be well above 300 and the total iterations for 10 folds could be well above 3000. I have also plotted the ROC curves for each fold just as comparisons with the mean ROC curve.

6 Logistic batch gradient descent result

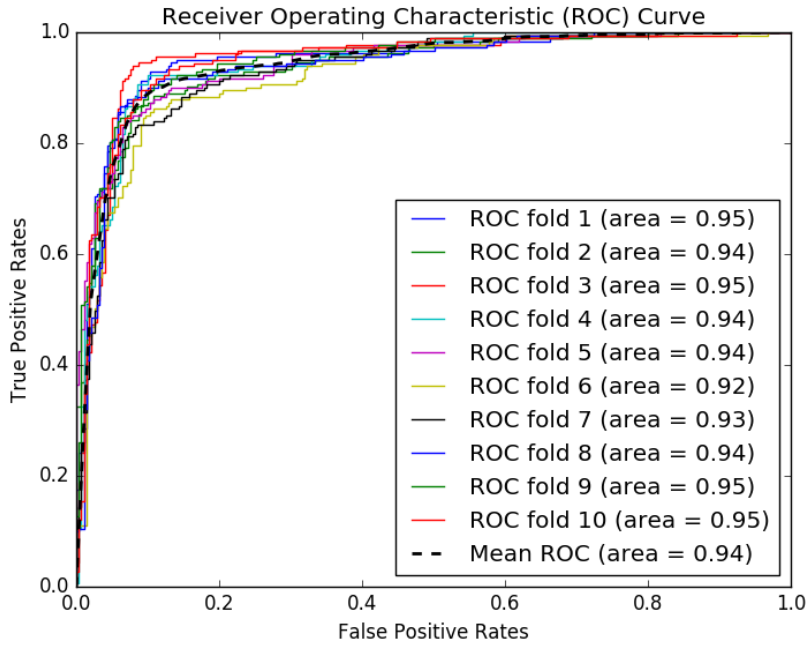
Convergence criterion: the difference of Mean Squared Error (MSE) smaller than 0.001

The learning curves are plotted below. A enlarged version is placed to the right of the original version. Learning rate greater than 0.000001 will lead to divergence.





The Receiver Operating Characteristic (ROC) curves are plotted below at the most efficient learning rate.



Note: The above ROC curves are obtained at a convergence criterion of 0.001. If the convergence criterion is reset to 0.0001, MSEs of higher learning rates could be reduced very close to the one we obtained from lower learning rates. However the iterations required for each fold would be well above 300 and the

total iterations for 10 folds could be well above 3000. I have also plotted the ROC curves for each fold just as comparisons with the mean ROC curve.