**Project Report 2**

**Learning to Rank using Linear Regression**

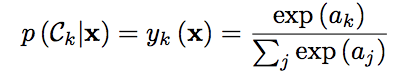
CSE-574 - Fall 2017

Deepika Chaudhary, Manpreet Dhanjal

(UB IT Name: d25 , PN: 50248725 ), (UB IT Name: dhanjal , PN: 50248725 )

**Logistic Regression:**

We used 1-of-K coding scheme t = [t1, ..., tK]. for our multiclass classfication task. The model can be represented as:



where the activation ak are given by ak = w⊤k x + bk.

The gradient of error is given by

../../../../Desktop/Screen%20Shot%202017-11-15%20at%2011.52.50%20AM.png

The weights are updated by the following method

../../../../Desktop/Screen%20Shot%202017-11-15%20at%2011.54.25%20AM.png

We used Mini batch SGD method to update the weights. The computation of gradient error can be done by matrix multiplication and dividing the data in batches largely out performs the speed of computing the gradient error.

**MNIST DATA:**

The input has 60000 x 784 features which has to be classified into 10 models

**Tuning hyper parameters : lambda and learning rate(Eta)**

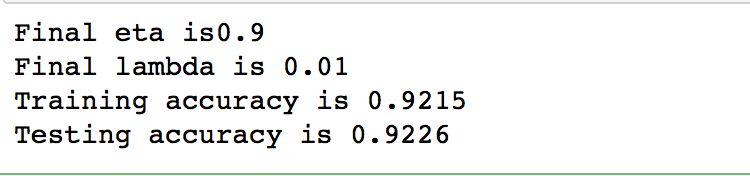
The learning rate and lambda are selected via grid search. The model is first trained using the training data 55000. The weight estimated through this training is then given to validation data and the accuracy is seen in all the combinations of lambda and eta. The maximum accuracy was estimated to be 96.3% on validation data at eta = 0.9 and lambda=0.001.

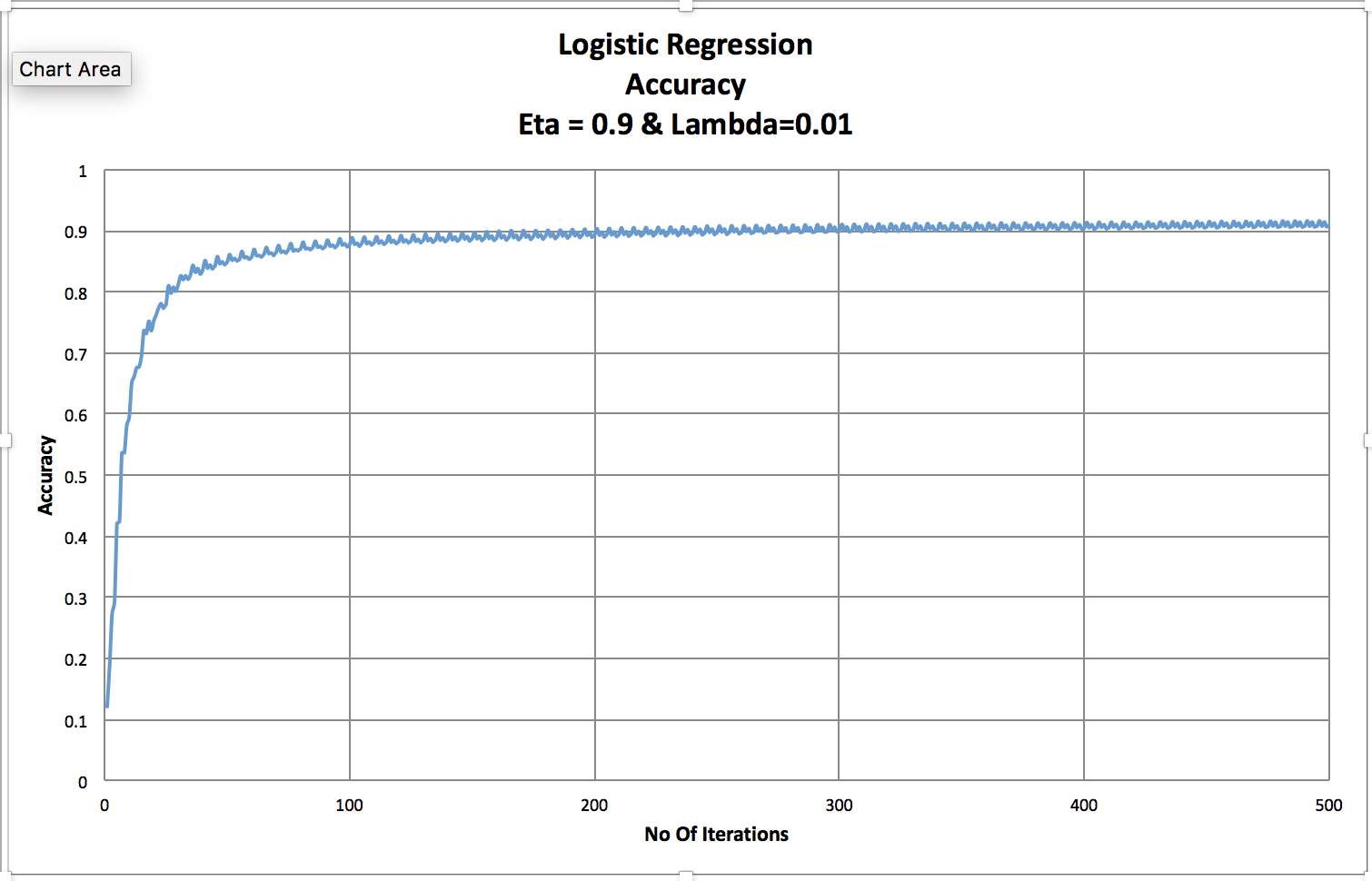
|  |  |  |
| --- | --- | --- |
| **Learning Rate (ETA)** | **Lambda** | **Accuracy** |
| 0.01 | 0.001 | 0.9358 |
| 0.01 | 0.01 | 0.9358 |
| 0.01 | 0.1 | 0.9358 |
| 0.03 | 0.001 | 0.9388 |
| 0.03 | 0.01 | 0.9388 |
| 0.03 | 0.1 | 0.9388 |
| 0.09 | 0.001 | 0.9424 |
| 0.09 | 0.01 | 0.9424 |
| 0.09 | 0.1 | 0.9424 |
| 0.3 | 0.001 | 0.9516 |
| 0.3 | 0.01 | 0.9516 |
| 0.3 | 0.1 | 0.9516 |
| 0.9 | 0.001 | 0.963 |
| 0.9 | 0.01 | 0.963 |
| 0.9 | 0.1 | 0.963 |

Accuracy on training data = 92.15%

Accuracy on validation data = 96.3%

Accuracy on testing data = 92.26%





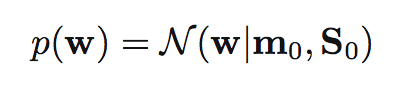
**USPS DATA**

The input has 20000 x 784 features which has to be classified into 10 models

Accuracy on training data = 36.58%

**Bayesian Logistic Regression:**

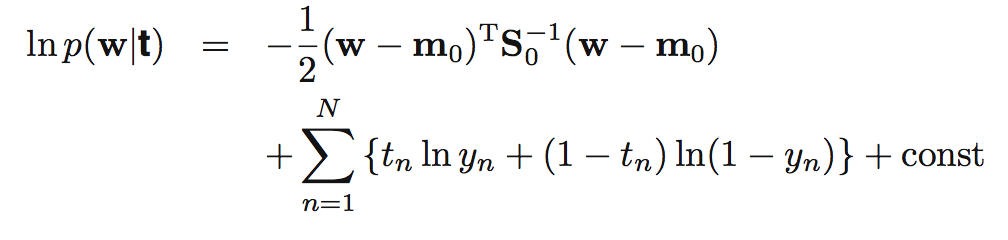
In bayesian logistic regression, we start with an initial belief about the distribution that is the posterior, which is our updated belief about the weights given evidence, is proportional to our prior (initial belief) times the likelihood. Adding a prior is equivalent to adding a regularization term equivalent to L2 norm. Because we seek a Gaussian representation for the posterior distribution, it is natural to begin with a Gaussian prior, which we write in the general form



where m0 and S0 are fixed hyper parameters. The posterior distribution over w is given by

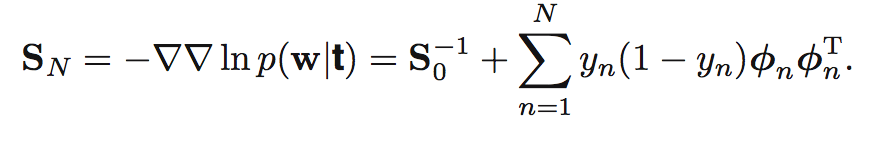
../../../../Desktop/Screen%20Shot%202017-11-19%20at%209.25.41%20PM.png where **t** = (t1 , . . . , tN )T .

The likelihood function is maximized by taking log both sides then the above equation takes form:

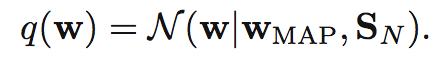


where yn = σ(wTφn)

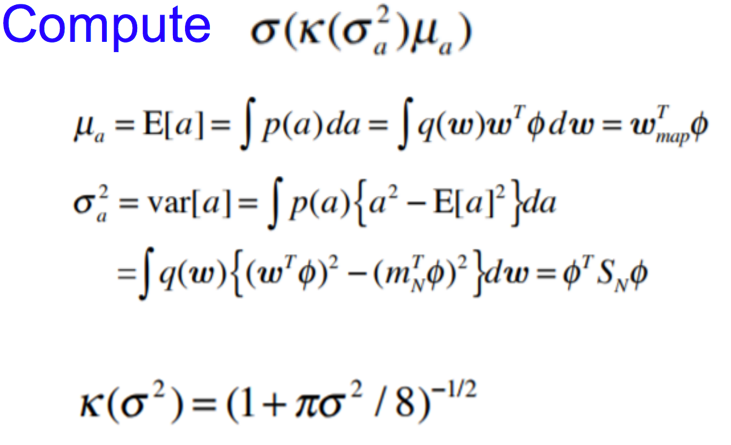
To obtain a Gaussian approximation to the posterior distribution, we first maximize the posterior distribution to give the MAP (maximum posterior) solution wMAP, which defines the mean of the Gaussian. The covariance is then given by the inverse of the matrix of second derivatives of the negative log likelihood, which takes the form

****

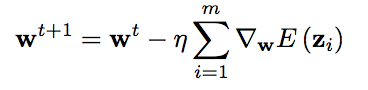
The Gaussian approximation to the posterior distribution therefore takes the form

****

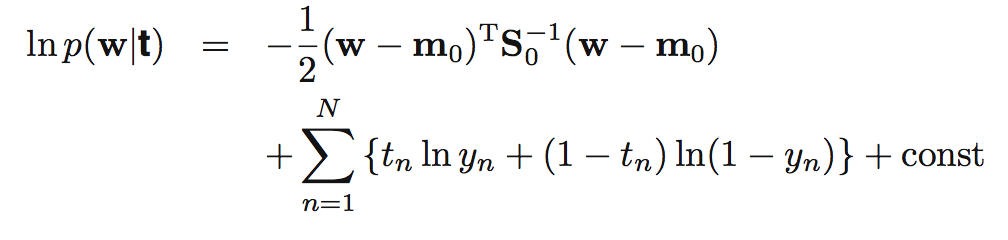
Initially we select S0 as a diagonal matrix of size m x m where, m (784) is the no of features of any given sample N (let 60000). Here we are given data which has to be classified into 10 models. For that first the hyper parameters SN and Wmap are trained first on the training data and then the prediction is done by the approximation given below using the updated SN and Wmap:



Initially Wmap is set to be a random Gaussian distribution which is then updated at every iteration using the formula



where gradient is given by taking derivative of equation:



This is done till we are able to classify our data correctly. The accuracy of the prediction is noted and the final weights and variance is used further on testing data.

W\_mapT  X Theta