Machine Learning Final Project Report

- *Sudhamayi Nanduri*

***References**:*

1. *http://en.wikipedia.org/wiki/Multinomial_logistic_regression*
2. *http://en.wikipedia.org/wiki/Confusion_matrix*
3. *Machine Learning from Murphy*

Project Title:

Predictive Analysis of Pen-based Recognition of Hand written Digits using Multinomial Logistic Regression

Description:

This project's purpose is to predict a hand-written digit (0-9). The predictive analysis is done by multinomial logistic regression on the data set of pen-based recognition of hand-written digits.

Data Set:

The data set considered for this algorithm is **Pen-based Recognition of Hand written Digits** from the UCLA repository.

The dataset is **multi-class** with each digit corresponding to a single class. Each record has 16 features and a class label.

The **training set** consists of 7494 samples with the following distributions for each class.

0: 780
1: 779
2: 780
3: 719
4: 780
5: 720
6: 720
7: 778
8: 719
9: 719

The **testing set** consists of 3497 sample with the following distributions for each class.

0: 363
1: 364
2: 364
3: 336
4: 364
5: 335
6: 336
7: 364
8: 335
9: 336

Data Pre-processing and Assumptions:

1. The mean, standard deviation and z-scores of the dataset are calculated for normalization.
2. The records in the data set are shuffled randomly to normalize the input.
3. Estimation of Intercept: 0 is chosen as category reference variable.

Algorithm Used: Multinomial Logistic Regression

1. This is of the form,

$$p(y = c|\mathbf{x}, \mathbf{W}) = \frac{\exp(\mathbf{w}_c^T \mathbf{x})}{\sum_{c'=1}^{C} \exp(\mathbf{w}_{c'}^T \mathbf{x})}$$

2. For Training Set, the following are calculated.
3. Loglikelihood,

$$\begin{aligned} \ell(\mathbf{W}) &= \log \prod_{i=1}^{N} \prod_{c=1}^{C} \mu_{ic}^{y_{ic}} = \sum_{i=1}^{N} \sum_{c=1}^{C} y_{ic} \log \mu_{ic} \\ &= \sum_{i=1}^{N} \left[\left(\sum_{c=1}^{C} y_{ic} \mathbf{w}_c^T \mathbf{x}_i \right) - \log \left(\sum_{c'=1}^{C} \exp(\mathbf{w}_{c'}^T \mathbf{x}_i) \right) \right] \end{aligned}$$

4. Negative Loglikelihood,

$$f(\mathbf{w}) = -\ell(\mathbf{w})$$

5. Gradient,

$$\mathbf{g}(\mathbf{W}) = \nabla f(\mathbf{w}) = \sum_{i=1}^{N} (\boldsymbol{\mu}_i - \mathbf{y}_i) \otimes \mathbf{x}_i$$

6. Weights for c-th column,

$$\nabla_{\mathbf{w}_c} f(\mathbf{W}) = \sum_i (\mu_{ic} - y_{ic}) \mathbf{x}_i$$

7. Hessian,

$$\mathbf{H}(\mathbf{W}) \quad = \quad \nabla^2 f(\mathrm{w}) = \sum_{i=1}^{N} (\mathrm{diag}(\boldsymbol{\mu}_i) - \boldsymbol{\mu}_i \boldsymbol{\mu}_i^T) \otimes (\mathrm{x}_i \mathrm{x}_i^T)$$

8. Minimizing,

$$\begin{aligned} f'(\mathbf{W}) &= f(\mathbf{W}) + \frac{1}{2} \sum_c \mathrm{w}_c \mathbf{V}_0^{-1} \mathrm{w}_c \\ \mathrm{g}'(\mathbf{W}) &= \mathrm{g}(\mathbf{W}) + \mathbf{V}_0^{-1} (\sum_c \mathrm{w}_c) \\ \mathbf{H}'(\mathbf{W}) &= \mathbf{H}(\mathbf{W}) + \mathbf{I}_C \otimes \mathbf{V}_0^{-1} \end{aligned}$$

9. This is passed to BFGS gradient optimizer.
10. The updated weights obtained are used to calculate the Hessian and probability predictions on the Testing Set.
11. Error rate is calculated and confusion matrix is created based on the probability predictions.

Observations and Analysis:

1. **Iteration Number Vs. Error Rate**:

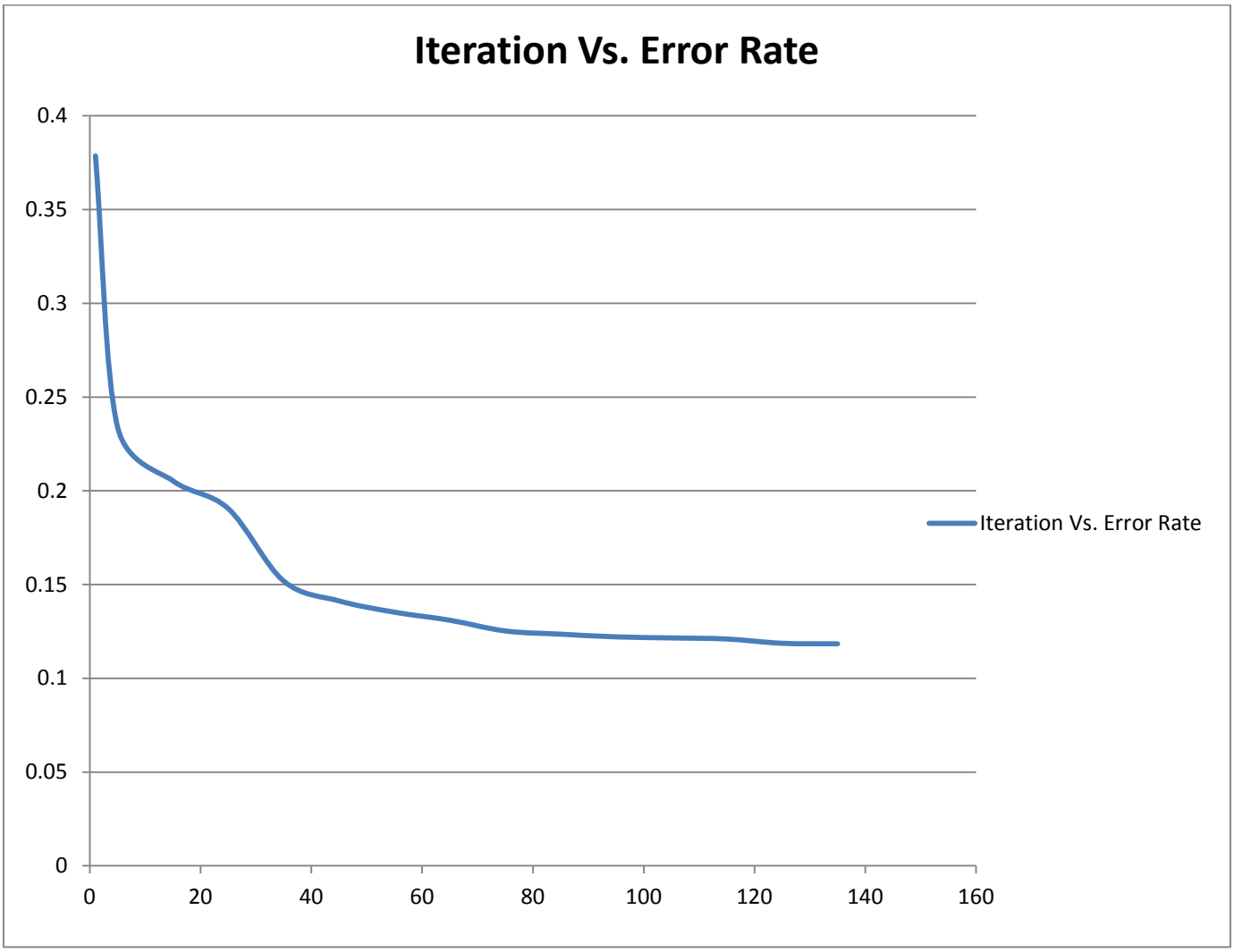
   The following table shows the decrease in error rate as the number of iterations are increased till convergence (difference between successive error rates <0.001 which is negligible).

Iteration Number	Error Rate
1	0.378610237346
5	0.233914784101
15	0.205318844724
25	0.190448956248
35	0.15184443809
45	0.14126394052
55	0.135258793251
65	0.130969402345
75	0.12525021447
85	0.123534458107
95	0.122104661138
105	0.121532742351
115	0.120960823563
125	0.118673148413
135	0.118387189019

From the above table we can observe that the error rate decreases with the increase in iteration number. The error rate is almost constant (difference is negligible) after certain number of iterations.

The error rate achieved for this dataset using multinomial logistic regression is 0.118, i.e., 11%.

2. Plot for Iterations Vs. Error Rate



The above graph shows a decreasing curve, showing the decrease in error rate.

3. **Confusion Matrix**:
   The confusion matrix for the final 135th iteration is given below.

Class	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	330	0	0	0	0	0	0	0	32	1
1	0	259	21	3	1	64	0	2	0	14
2	0	5	359	0	0	0	0	0	0	0
3	0	1	0	326	0	0	1	7	0	1
4	0	0	0	0	348	10	0	0	0	6
5	1	2	0	9	6	275	2	2	12	26
6	0	0	0	0	0	1	316	0	19	0
7	0	40	10	1	6	0	6	292	6	3
8	18	2	11	2	0	14	2	12	274	0
9	0	15	0	3	6	5	0	2	1	304

4. **Tables of Confusion**
   The following are the tables of confusion derived from the above matrix for each class.

   Class 0:

True Positive: 330	**False Negative**: 33
False Positive: 19	**True Negative**: 3116

- 330 samples have been classified correctly as ‘0’
- 33 samples which belong to ‘0’ are classified into other classes
- 19 samples from other classes are classified to ‘0’
- 3116 samples have been rightly not classified to ‘0’
- Error Rate = 0.136125

Class 1:

True Positive: 259	**False Negative**: 105
False Positive: 65	**True Negative**: 3069

- 259 samples have been classified correctly as ‘1’
- 105 samples which belong to ‘1’ are classified into other classes
- 65 samples from other classes are classified to ‘1’
- 3069 samples have been rightly not classified to ‘1’
- Error Rate = 0.396703

Class 2:

True Positive: 359	**False Negative**: 5
False Positive: 42	**True Negative**: 3092

- 359 samples have been classified correctly as ‘2’
- 5 samples which belong to ‘2’ are classified into other classes
- 42 samples from other classes are classified to ‘2’
- 3092 samples have been rightly not classified to ‘2’
- Error Rate = 0.115763

Class 3:

True Positive: 326	**False Negative**: 10
False Positive: 18	**True Negative**: 3144

- 326 samples have been classified correctly as ‘3’
- 10 samples which belong to ‘3’ are classified into other classes
- 18 samples from other classes are classified to ‘3’
- 3144 samples have been rightly not classified to ‘3’
- Error Rate = 0.079096

Class 4:

True Positive: 348	**False Negative**: 16
False Positive: 19	**True Negative**: 3115

- 348 samples have been classified correctly as ‘4’
- 16 samples which belong to ‘4’ are classified into other classes
- 19 samples from other classes are classified to ‘4’
- 3115 samples have been rightly not classified to ‘4’
- Error Rate = 0.091383

Class 5:

True Positive: 275	**False Negative**: 60
False Positive: 94	**True Negative**: 3069

- 275 samples have been classified correctly as ‘5’
- 33 samples which belong to ‘5’ are classified into other classes
- 19 samples from other classes are classified to ‘5’
- 3116 samples have been rightly not classified to ‘5’
- Error Rate = 0.358974

Class 6:

True Positive: 316	**False Negative**: 20
False Positive: 11	**True Negative**: 3151

- 316 samples have been classified correctly as ‘6’
- 20 samples which belong to ‘6’ are classified into other classes
- 11 samples from other classes are classified to ‘6’
- 3151 samples have been rightly not classified to ‘6’
- Error Rate = 0.089337

Class 7:

True Positive: 292	**False Negative**: 72
False Positive: 25	**True Negative**: 3109

- 292 samples have been classified correctly as ‘7’
- 72 samples which belong to ‘7’ are classified into other classes
- 25 samples from other classes are classified to ‘7’
- 3109 samples have been rightly not classified to ‘7’
- Error Rate = 0.249357

Class 8:

True Positive: 274	**False Negative**: 61
False Positive: 70	**True Negative**: 3093

- 274 samples have been classified correctly as ‘8’
- 61 samples which belong to ‘8’ are classified into other classes
- 70 samples from other classes are classified to ‘8’
- 3093 samples have been rightly not classified to ‘8’
- Error Rate = 0.323456

Class 9:

True Positive: 304	**False Negative**: 32
False Positive: 51	**True Negative**: 3111

- 304 samples have been classified correctly as ‘9’
- 32 samples which belong to ‘9’ are classified into other classes
- 51 samples from other classes are classified to ‘9’
- 3111 samples have been rightly not classified to ‘9’
- Error Rate = 0.214470

From the above tables and error rate values, we can see that class ‘1’ has most wrong predictions, followed by class ‘5’. The class ‘3’ has the least wrong predictions, followed by class ‘6’.

The sequential order of classes in decreasing error rate is given below.

1 > 5 > 8 > 7 > 9 > 0 > 2 > 4 > 6 > 3

5. **With and without Z-Scores**

The data set has been pre-processed to calculate z-scores for normalization.

Condition	Error Rate
With Z-Scores	0.118387189019
Without Z-Scores	0.120102945382

We can observe that the error rate is slightly higher without normalization of dataset. Hence, we can say that the dataset is slightly skewed.

6. **With and without shuffle**

The records in the dataset have been shuffled.

Condition	Error Rate
Training Set- Shuffled	0.118387189019
Training Set – Not Shuffled	0.118673148413

The error rate very slightly decreased with shuffled input. The difference is almost negligible. Hence, the data set is very slightly skewed.

Conclusion:

The multinomial logistic trainer is trained to give an error rate of 11%. The z-scores reduce the error rate slightly. The shuffle function also reduces the error rate slightly. From the above observations we can conclude that the dataset is slightly skewed.

Also, we can see that the trainer classifies few classes efficiently and few classes non-efficiently. Hence, by dissolving these less efficient classes, we can obtain a better error rate.