Logistic Regression 2-Class and K-Class

1. Problem Statement

The data set should be classified using discriminative learning model. The data with two class and K class should be trained and tested using the logistic regression model. Logistic regression can be seen as a special case of generalized linear model and thus analogous to linear regression. The model of logistic regression is based on the relationship between dependent and independent variables from those of linear regression. The 10-fold cross validation has to be done to analyze the performance of the model. The accuracy of the model is analyzed with the training and testing error calculated by Mean Square Error.

2. Problem Solution

The solution is to find the parameters of logistic regression. The parameters are similar to linear regression but inferred with logistic function. There are different logistic function but we have implemented sigmoid for 2-Class and soft-max function for K-class. Based on the logistic function we find the parameter of the given data and determine the class label of the data using discriminant function. The features and the labels are in the form of vectors. The predictor function is used for modeling.

3. Implementation Details

For two class data the parameters are found using sigmoid function and for K-class data the parameter is found using soft-max function. The logistic function is then used in stochastic gradient descent function. The first step is to assume the initial parameters which are random and closer to zero, the number of hidden layers. The second step is to set find the hidden layer output. The third step is finding the output layer output for each class. The discriminant function is to find the class of the given data using the output of the output layer.

I have implemented the code in ipython notebook. The filename has to be mentioned and in the tool bar option “Cell” -> “Run All” will implemented the whole file and the results will be printed.

4. Results and Discussion

The data set used is "**data_banknote_authentication.txt**"

a) Parameter with no bias:

```
[-2.68534191 -1.64560428 -1.74869086 -0.82080437]
```

b) Parameter with Bias:

```
[ 3.36153006, -3.22054039, -1.83463974, -2.21676291, -0.16402614]
```

c) 10 fold validation for data with no bias

```
Fold 0 Accuracy 0.927536231884
Fold 1 Accuracy 0.920289855072
Fold 2 Accuracy 0.905109489051
Fold 3 Accuracy 0.912408759124
Fold 4 Accuracy 0.963503649635
Fold 5 Accuracy 0.890510948905
Fold 6 Accuracy 0.941605839416
Fold 7 Accuracy 0.897810218978
Fold 8 Accuracy 0.941605839416
Fold 9 Accuracy 0.978102189781
Average Accuracy  0.927848302126
```

d) Test Error accuracy with no bias

```
Fold 0 Testing Error [ 0.06401945]
Fold 1 Testing Error [ 0.05996759]
Fold 2 Testing Error [ 0.06072874]
Fold 3 Testing Error [ 0.06072874]
Fold 4 Testing Error [ 0.06477733]
Fold 5 Testing Error [ 0.05910931]
Fold 6 Testing Error [ 0.07287449]
Fold 7 Testing Error [ 0.06963563]
Fold 8 Testing Error [ 0.06558704]
Fold 9 Testing Error [ 0.06882591]
Average Mean Square Error
0.0646254240513
```

e) 10 fold validation for data with bias

```
Fold 0 Accuracy 0.985507246377
Fold 1 Accuracy 0.985507246377
Fold 2 Accuracy 0.985401459854
Fold 3 Accuracy 0.978102189781
Fold 4 Accuracy 0.992700729927
Fold 5 Accuracy 0.992700729927
Fold 6 Accuracy 0.992700729927
Fold 7 Accuracy 0.963503649635
Fold 8 Accuracy 0.978102189781
Fold 9 Accuracy 0.978102189781
Average Accuracy  0.983232836137
```

f) Test error accuracy with bias

```
Average Mean Square Error
0.0163587031411
```

g) Evaluation

```
Confusion Matrix
[[752  10]
 [  3 607]]
Accuracy 0.990524781341
Precision [0.9960264900662251 2, 0.98379254457050247]
Recall [0.98687664041994749, 0.99508196721311 47]
F_score [0.99143045484508896, 0.98940505297473513]
```

h) 10 fold validation for data with data mapped to higher dimensional,

```
Average Accuracy
0.995625727282

Average Mean Square Error
0.0034012690372
```

i) Evaluation on higher dimension

```
Confusion Matrix
[[762   0]
 [  8 602]]
Accuracy 0.99416909621
Precision [0.98961038961038961, 1.0]
Recall [1.0, 0.9868852459016394]
F_score [0.99477806788511747, 0.99339933993399332]
```

K-Class

a) Data set used is "iris.data.txt"

b) Mean Error without bias

```
Average Mean Square Error
Training Error         Testing Error
0.107407407407       0.12

Mean Accuracy 0.88
```

c) Cross Validation on biased data

```
Fold 0 Testing Error [ 0.2]
Fold 1 Testing Error [ 0.]
Fold 2 Testing Error [ 0.13333333]
Fold 3 Testing Error [ 0.06666667]
Fold 4 Testing Error [ 0.13333333]
Fold 5 Testing Error [ 0.]
Fold 6 Testing Error [ 0.33333333]
Fold 7 Testing Error [ 0.]
Fold 8 Testing Error [ 0.13333333]
Fold 9 Testing Error [ 0.06666667]
Average Mean Square Error
Training Error         Testing Error
0.0962962962963      0.106666666667
```

d) Confusion matrix through the cross validation

```
array([[50,  0,  0],
       [ 0, 34, 16],
       [ 0,  0, 50]])
```

K-Class MNIST Data

The digits data training set is reduced in size and tested for the whole testing data and below is the evaluation result.

Confusion Matrix

Class	Class 0	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8	Class 9
Class 0	**942**	0	2	2	0	6	18	1	9	0
Class 1	0	**1063**	14	3	1	6	4	0	44	0
Class 2	30	21	**823**	26	17	2	41	24	41	7
Class 3	7	1	31	**856**	1	35	10	14	33	22
Class 4	6	9	4	0	**858**	1	18	2	11	73
Class 5	26	2	1	89	20	**648**	36	11	46	13
Class 6	16	3	12	1	20	12	**888**	1	5	0
Class 7	6	31	48	1	14	0	0	**887**	7	34
Class 8	11	19	8	37	17	26	17	18	**800**	21
Class 9	22	12	12	10	55	12	3	36	11	**836**
Total	1066	1161	955	1025	1003	748	1035	994	1007	1006

Accuracy 0.8601

Precision	
Class 0	0.883677
Class 1	0.91559
Class 2	0.86178
Class 3	0.835122
Class 4	0.855434
Class 5	0.86631
Class 6	0.857971
Class 7	0.892354
Class 8	0.794439
Class 9	0.831014

Recall	
Class 0	0.961224
Class 1	0.936564
Class 2	0.797481
Class 3	0.847525

Class 4	0.873727
Class 5	0.726457
Class 6	0.926931
Class 7	0.86284
Class 8	0.821355
Class 9	0.828543

F-score	
Class 0	0.920821
Class 1	0.925958
Class 2	0.828384
Class 3	0.841278
Class 4	0.864484
Class 5	0.790244
Class 6	0.891119
Class 7	0.877349
Class 8	0.807673
Class 9	0.829777

5. References

https://en.wikipedia.org/wiki/Logistic_regression