CMPE 462 - Project 1
Binary Classification with Logistic Regression

Student ID1: 2015400069
Student ID2: 2015400177
Student ID3: 2019700087

April 2020

1 Feature Extraction

In this project we are asked to implement logistic regression to distinguish two digits between each other which were given in a handwritten data set. First, we need to load all the files which contain required training and test features and labels. Briefly, both training and test sets have almost the same ratio between digits (˜%60 for Digit 1, ˜%40 for Digit 5)

Figure 1: Distribution of digits among training and test sets

Each row in the data set has 256 values as an array. These values must be reshaped to a matrix from an array. After reshaping operation, first samples of each classes were taken and visualized in a plot.

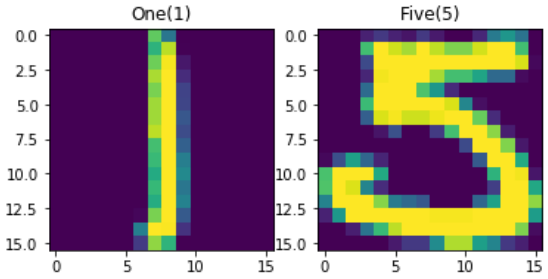

Figure 2: Visualization of digits

1.1 Representative 1

Some features must be extracted to create a meaningful approach and to provide better understanding towards data. As stated in the assignment, symmetry with respect to the y-axis and average intensity of the picture were calculated to form the Representation 1. Also, the given data sets and labels are ordered by an index. To prevent overfitting, these data sets must be rearranged(shuffled). After extracting features from the training data set, scatter plots were provided to show distinction between two digits e.g. classes.

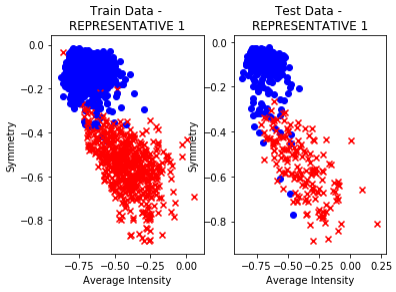

Figure 3: Scatter plot for representation 1

1.2 Representative 2

To provide more evident distinction, we came up with a different feature extraction method. We used the center of gravity feature with x-axis symmetry. The reason behind this decision is the structure of these two digits. By using x-axis symmetry, these two digits can be distinguished from each other. If we consider digit 1, it can clearly be seen that the value of an index is almost identical to the value in the symmetrical index since digit 1 seems like a line across the y-axis. But if we consider digit 5, it has edges instead of a linear form. Also, again considering the structures of digits, these digits have center of gravity points in different positions. If we consider digit 1, it can be inferred that it has a center of gravity which is close to the center of the image (16x16 matrix). On the other hand, digit 5 has a more scattered structure than digit 1 so, it is expected that is has a center of gravity point which is distant from the center of the image. We took advantage of this difference by finding the midpoint of the pixels which have greater values than -1 and calculating the distance between center of the image and these points. As shown in the plots, by using x-axis symmetry and center of gravity features two classes are separated from each other perfectly despite some outliers. Also, the distance between two classes is much higher than Representation 1. So, we can infer that Representation 2 performed a better distinction between classes.

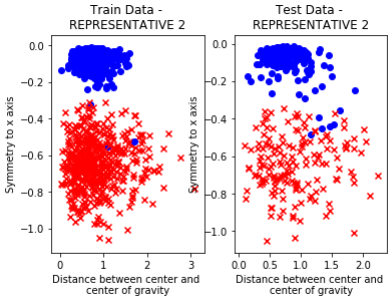

Figure 4: Scatter plot for representation 2

2 Logistic Regression

After extracting features, we have implemented some functions to create logistic regression model,

- adding an intercept term to our data vector.
- sigmoid function
- loss function
- gradient calculation
- loss calculation for each iteration

2.1 Gradient Derivation

2.1.1 Reminders

- Sigmoid function

$$\theta(s) = \frac{1}{1 + \exp(-s)} \tag{1}$$

- Derivative of ln

$$f(x) = \ln\left(g(x)\right) \tag{2}$$

$$f'(x) = \frac{g'(x)}{g(x)} \tag{3}$$

- Derivative of exp

$$f(x) = \exp\left(g(x)\right) \tag{4}$$

$$f'(x) = g'(x)\exp\left(g(x)\right) \tag{5}$$

- Derivative of vector 1

$$f(w) = w^T A w \tag{6}$$

$$\frac{\partial f}{\partial w} = (A + A^T)w \tag{7}$$

- Derivative of vector 2

$$f(w) = w^T b \tag{8}$$

$$\frac{\partial f}{\partial w} = b \tag{9}$$

2.1.2 Derivation

$$L = \frac{1}{N}\sum_{n=1}^{N} \ln\left(1 + \exp\left(-y_n w^T x_n\right)\right) + \lambda||w||_2^2 \tag{10}$$

$$\frac{\partial L}{\partial w} = \frac{1}{N}\sum_{n=1}^{N} \frac{-y_n x_n \exp\left(-y_n w^T x_n\right)}{1 + \exp\left(-y_n w^T x_n\right)} + 2\lambda w \tag{11}$$

$$= \frac{1}{N}\sum_{n=1}^{N} \frac{-y_n x_n}{1 + \exp\left(y_n w^T x_n\right)} + 2\lambda w \tag{12}$$

$$= \frac{1}{N}\sum_{n=1}^{N} -y_n x_n \theta(-y_n w^T x_n) + 2\lambda w \tag{13}$$

First, we added intercept term to form data vector. To decide, which learning rate is the best for our data set. We trained our model for [0.2, 0.4, 0.6, 0.8, 1] separately and plotted our loss values in plots below.

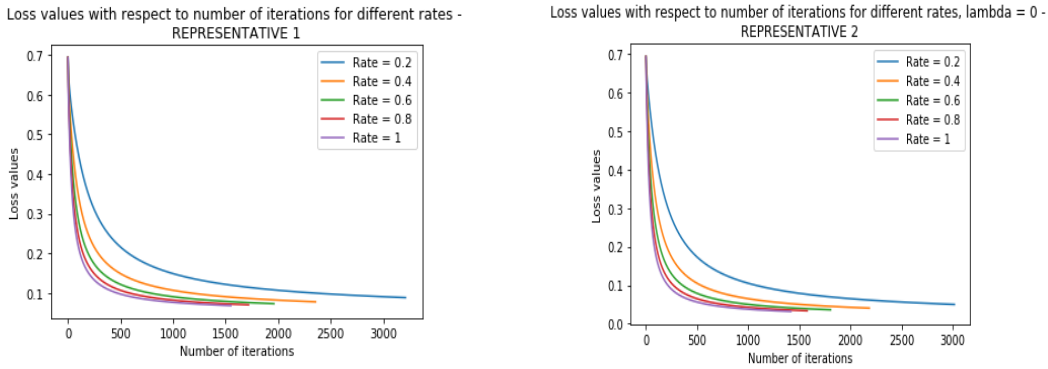

Figure 5: Loss values with respect to number of iterations or different rates

As can be seen on the graphs above, number of iterations for both representation 1 and 2 are increasing with respect to decrease in learning rate. So, learning rate will be selected as 1 to provide a minimum iteration number while implementing logistic regression.

After deciding learning rate, logistic regression model was trained with both representation 1 and 2 with lambda value of 0.01 and loss values with respect to number of iterations were plotted and can be seen below.

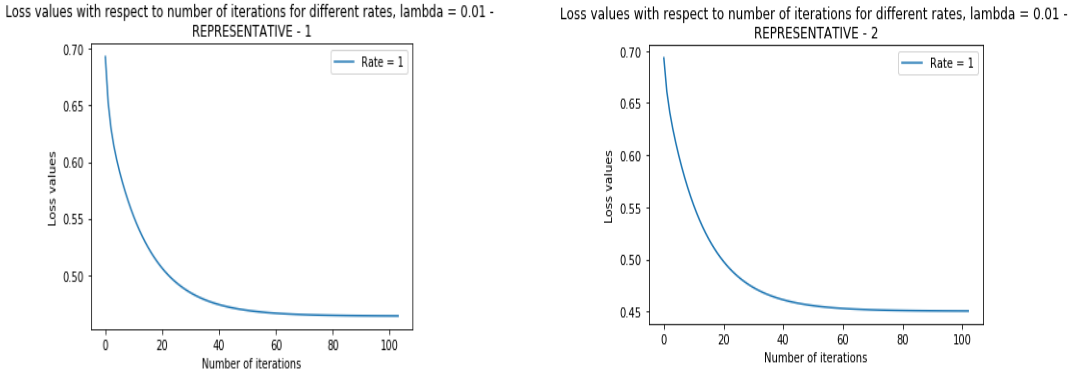

Figure 6: Loss values with respect to number of iterations or different rates with regularization

2.2 Cross Validation

After training our logistic regression model with training data, we implemented a 5-fold cross validation to find the optimal lambda value. We chose 4 values between 0 and 1 [0.001, 0.01, 0.05, 0.1] to achieve this goal. We kept track of accuracy values for each lambda values in every fold and these values can be seen in the tables below for representation 1 and 2.

Lambda	Accuracy					Mean	Std
0.001	0.96153846	0.98717949	0.98076923	0.98076923	0.95192308	0.97	0.013
0.01	0.92948718	0.93910256	0.92628205	0.92628205	0.89423077	0.92	0.015
0.05	0.71474359	0.75961538	0.65064103	0.70512821	0.65384615	0.70	0.040
0.1	0.65384615	0.69551282	0.61217949	0.65705128	0.60897436	0.65	0.030

Table 1: 5-fold cross validation results for Representation 1

Lambda	Accuracy					Mean	Std
0.001	0.99038462	0.99358974	0.99358974	0.99358974	0.97435897	0.99	0.007
0.01	0.96794872	0.96474359	0.97115385	0.99038462	0.94230769	0.97	0.015
0.05	0.76923077	0.83012821	0.78525641	0.83974359	0.70192308	0.79	0.049
0.1	0.66987179	0.72115385	0.6474359	0.7275641	0.60576923	0.67	0.046

Table 2: 5-fold cross validation results for Representation 2

As can be seen in the tables, mean of accuracy values for lambda = 0.001 have the highest value among other lambda values. So, 0.001 was chosen as best lambda value for both representatives.

3 Evaluation

3.1 Accuracy Calculation

Training and test classification accuracies are calculated using following formula:

$$Accuracy = \frac{number of correctly classified samples}{total number of samples} x100 \tag{14}$$

3.2 Testing

After training and cross validating our logistic regression model, we evaluate our results by using test data set. In theoretical application of logistic regression contains sigmoid function classify training samples. Sigmoid function lies between 0 and 1 but it can take any real numbers in this interval. 0.5 was selected to satisfy distinction between classes. In other words, any result which is greater than 0.5 will be marked as **digit 1**, others will be marked as **digit 5**. This methodology provides a fundamental binary classification between two classes.

We were simply asked to find training and test accuracy values with and without regularization. In earlier parts of our implementation, we found 0.001 as best lambda value by using 5-fold cross validation outputs.

Train/Test	Repr.	Lambda	Accuracy(%)
Train	1	0	97.95
Test	1	0	96.22
Train	1	0.001	97.37
Test	1	0.001	95.04

Table 3: Accuracy table for Representative 1.

Train/Test	Repr.	Lambda	Accuracy(%)
Train	2	0	99.68
Test	2	0	98.58
Train	2	0.001	99.17
Test	2	0.001	98.35

Table 4: Accuracy table for Representative 2.

3.3 Visualizing Decision Boundary

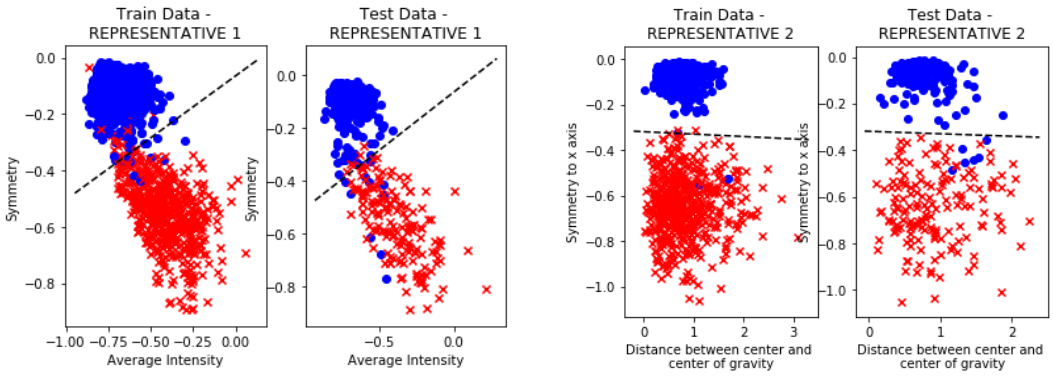

Figure 7: Accuracy values & decision boundaries for representatives

4 Q&A

1. Did regularization improve the generalization performance (did it help reducing the gap between training and test accuracies/errors)? Did you observe any difference between using Representation 1 and 2?

   - Regularization reduces the training and test accuracy values and increases the difference between training and test accuracy values. (In representation 2, the gap stays almost the same.)

2. Which feature set did give the best results? Which one is more discriminative?

   - Representation 2 gave better results on both training and test sets. Also, representation 2 is more discriminative compared to representation1. In scatter plots, representation 2 has a higher distance between classes compared to representation 1.

3. What would be your next step to improve test accuracy?

   - Since two classes have different number of samples (~%60 and ~%40 for **digit 5** and **digit 1**), we have unbalanced data set. Balancing the number of samples for each class, may increase training and test accuracy.
   - Increasing the dimension of extracted features might be useful for better test accuracy values.
   - A better feature extraction method could be used to increase test accuracy such as using an algorithm for edge detection and angle of pixels can be calculated to classify images with higher accuracy.
   - Also, increasing the training set size could help logistic regression model to draw a better decision boundary between two classes.