Project 2 Report

Redwan Ibne Seraj Khan(UBITname: rikhan)
Department of Computer Science and Engineering
University at Buffalo
Buffalo, NY-14228
rikhan@buffalo.edu

Abstract

In this project different machine learning models like linear regression, logistic regression and neural networks was used to solve a problem of determining the similarity between the handwritten samples of the known and the questioned writer by using linear regression, logistic regression and neural networks.

1 Linear Regression

1.1 Closed Form and Gradient Descent on Human Observed Dataset

At first the data was preprocessed. The target data was collected and separated. No features were null variant features, so no removing was necessary to inverse the co-variance matrix. The data was split into Training, Validation and Testing sets. Training consisted of 80% of the data and both validation and testing consisted 10% of the data each.

t $= w^T \varphi(\mathbf{x})$ is our linear regression model for closed form, where t is the target value. From this equation we get w = $(\varphi^T \varphi)^{-1} \varphi^T$t . As $\varphi(\mathbf{x})$ is not a square matrix we need the pseudo inverse.

After the preprocessing of data, the training dataset had a shape of (18, 235059) for concatenated features and (9, 235059) for subtracted features. The validation data had a shape of (18,29382) for concatenated features and (9,29382) for subtracted features. The test data had a shape of (18,29381) for concatenated features and (9, 29381) for subtracted features.

The computation needed basis function for converting input vector x into a scalar value. The number of basis functions was taken as 10 as the starting point. Gaussial radial basis function was taken for working on this project.

$$\phi_j(\mathbf{x}) = \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_j)^\top \Sigma_j^{-1}(\mathbf{x}-\boldsymbol{\mu}_j)\right)$$

Here the dimensions of the quantities that would be found from computation of concatenated features are shown below.

x - μ_j : 1x18

Σ_j^{-1} : 18 x18

So the calculation of the above basis function would give a scalar value of dimension 1x1.

BigSigma is a diagonal matrix which was created with the variances of feature vectors. The dimension of BigSigma was 18x18. The value of μ is obtained by clustering of data.

Let us take the example of training set data to clarify the solution. The design matrix φ(**x**) for training data will have the dimension of 55699x10. The matrix multiplication for the calculation of w is shown below.

$$(\varphi^T \varphi)^{-1} \varphi^T t$$
$$=((10x235059).(235059x10))^{-1} (10x235059).(235059x1)$$
$$= \quad (10x10) \quad . \quad (10x1)$$
$$= \quad (10x1)$$

After doing all the computation RMS was used for testing the results.

The Gradient Descent Solution was relatively straightforward. For implementing the gradient descent solution at first the weights were initialized.

$$[W]_{10x1}$$

Then the weights were updated for each data point using the equation that was given in the project description 1.2. For sake of keeping this report brief and concise the equations that have already been given has not been mentioned. The learning rate decided how big each step should be and hence it was kept small. At last RMS values were calculated for the training set, validation set and testing sets.

1.2 Closed Form and Gradient Descent on GSC Dataset

The process of reaching the solution for GSC dataset was similar to the way it was done for Human Observed Data. The only difference came in the processing of data. The GSC data had 512 features for each of the samples. Working with such a huge number of features for nearly 800000 samples of same and different pairs would be daunting and with the limited RAM that is available it would be near to impossible. For this reason, the data was preprocessed and I worked with 100 features.

After the preprocessing of data, the training dataset had a shape of (200, 160000) for concatenated features and (100, 160000) for subtracted features. The validation data had a shape of (200,19999) for concatenated features and (100,19999) for subtracted features. The test data had a shape of (200,19999) for concatenated features and (100, 19999) for subtracted features.

1.3 Effect of Hyper-parameters and Performance

1.3.1 Effect of different number of Basis Function and regularization term

After increasing the number of basis functions there was no significant improvement in performance. In all iterations the RMS value of the training, validation and testing sets ranged from 40-50% for gradient descent solution and 0-5% for human observed datasets.

1.3.2 Effect of center of Gaussian radial basis function

There was no notable effect on choosing the centers of Gaussian radial basis function in a random way.

1.3.3 Effect of spread of spread of Gaussian radial basis function

Changing the variances to be 1/10 decreased the accuracy to some extent for all of the data sets. As an example, the results of Human Observed Concatenated features have been shown below.

```
E_rms Training   = 0.568733434231337
E_rms Validation = 0.678278713908991
E_rms Testing    = 0.790239848034986
```

1.3.4 Effect of Learning rate and La

After increasing the learning rate gradually to 0.2 the performance deteriorated to a little extent for all of the datasets but then remained constant no matter how much the learning rate was increased. Having a very big learning rate caused the computation to go out of bounds. No notable change was observed on changing the value of La as well. As an example, the end result is shown below for Human Observed concatenated features.

```
E_rms Training   = 0.6023
E_rms Validation = 0.6456
E_rms Testing    = 0.6889
```

1.4 General Performance of the Different Models

General Performance for Human Observed Concatenated Features using Closed Form.

```
E_rms Training   = 0.05768044755871531
E_rms Validation = 0.00554583050653763
E_rms Testing    = 0.00511553547539180
```

General Performance for Human Observed Concatenated Features using Gradient Descent.

```
E_rms Training   = 0.27069
E_rms Validation = 0.25912
E_rms Testing    = 0.23591
```

General Performance for Human Observed Subtracted Features using Closed Form.

```
E_rms Training   = 0.057779665546661654
E_rms Validation = 0.003818872974286936
E_rms Testing    = 0.003714708522479641
```

General Performance for Human Observed Subtracted Features using Gradient Descent.

```
E_rms Training   = 0.43266
E_rms Validation = 0.36752
E_rms Testing    = 0.35043
```

General Performance for GSC Concatenated Features using Closed Form

```
E_rms Training   = 0.4987388325281887
E_rms Validation = 0.4189130017090638
E_rms Testing    = 0.4267765618710845
```

General Performance for GSC Concatenated Features using Gradient Descent

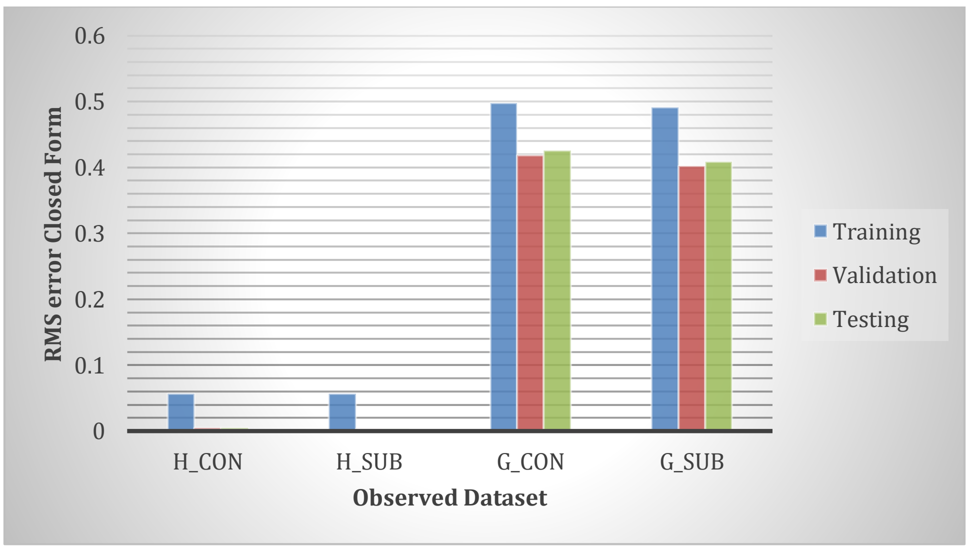
```
E_rms Training   = 0.6805
E_rms Validation = 0.4386
E_rms Testing    = 0.4279
```

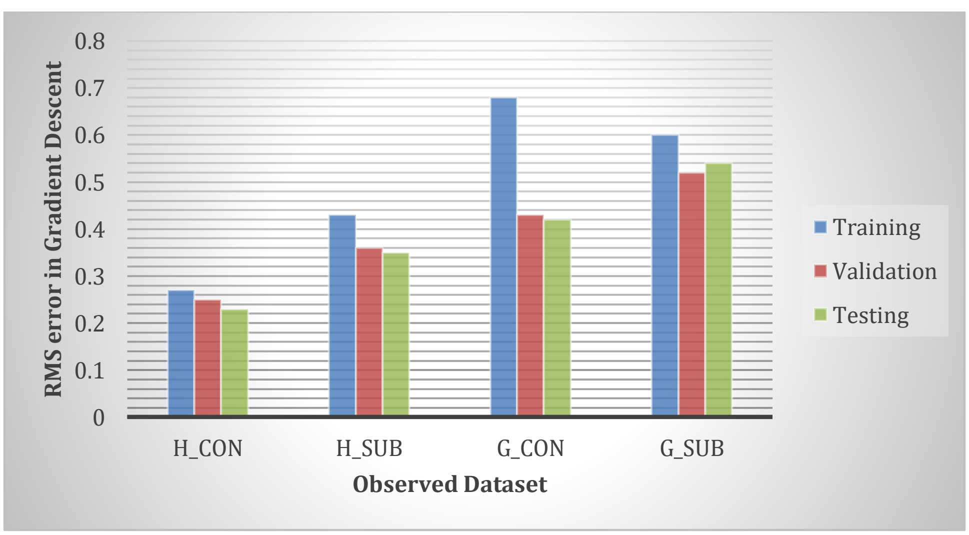
General Performance for GSC Subtracted Features using Closed Form

```
E_rms Training   = 0.4912013881428152
E_rms Validation = 0.4027950068771984
E_rms Testing    = 0.4092787775034313
```

General Performance for GSC Subtracted Features using Gradient Descent

```
E_rms Training   = 0.60355
E_rms Validation = 0.52785
E_rms Testing    = 0.54333
```





2 Logistic Regression

The sigmoid function was used in the logistic regression model. After calculations the loss function that was obtained is given below.

(-y * np.log(h) - (1 - y) * np.log(1 - h))

Here h is the target value that we obtained. For every point we updated the weights by taking the differential of this loss function.

2.1 General Performance of the Different Models

General Performance for Human Observed Subtracted Features

```
prediction accuracy of Training set:0.9966348874112457
prediction accuracy of Validation set:1.0
prediction accuracy of Testing set:1.0
```

General Performance for Human Observed Concatenated Features

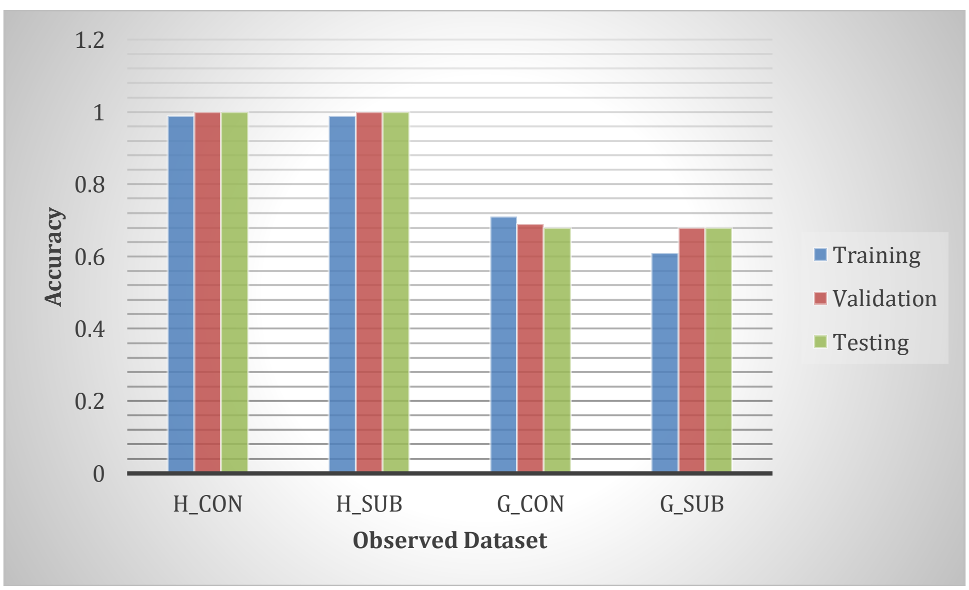
```
prediction accuracy of Training set:0.9966348874112457
prediction accuracy of Validation set:1.0
prediction accuracy of Testing set:1.0
```

General Performance for GSC Concatenated Features

```
prediction accuracy of Training set:0.70881875
prediction accuracy of Validation set:0.6896344817240863
prediction accuracy of Testing set:0.6764338216910846
```

General Performance for GSC Subtracted Features

```
prediction accuracy of Training set:0.6110875
prediction accuracy of Validation set:0.6834841742087104
prediction accuracy of Testing set:0.6812840642032102
```



3 Neural Networks

The same preprocessed dataset was used for neural networks. As the model was built from scratch, to avoid complexity, the number of hidden layer was 1 and the number of neurons in the hidden layer was 1000. For activation layer, sigmoid function was used. The model learned through forward and backward propagation and updated weights accordingly.

3.1 General Performance of the Different Models

General Performance for Human Observed Concatenated Features

```
Prediction Loss of Training Set: 0.24216909999999975
Prediction Loss of Validation Set: 0.25
Prediction Loss of Testing Set: 0.25
```

General Performance for Human Observed Subtracted Features

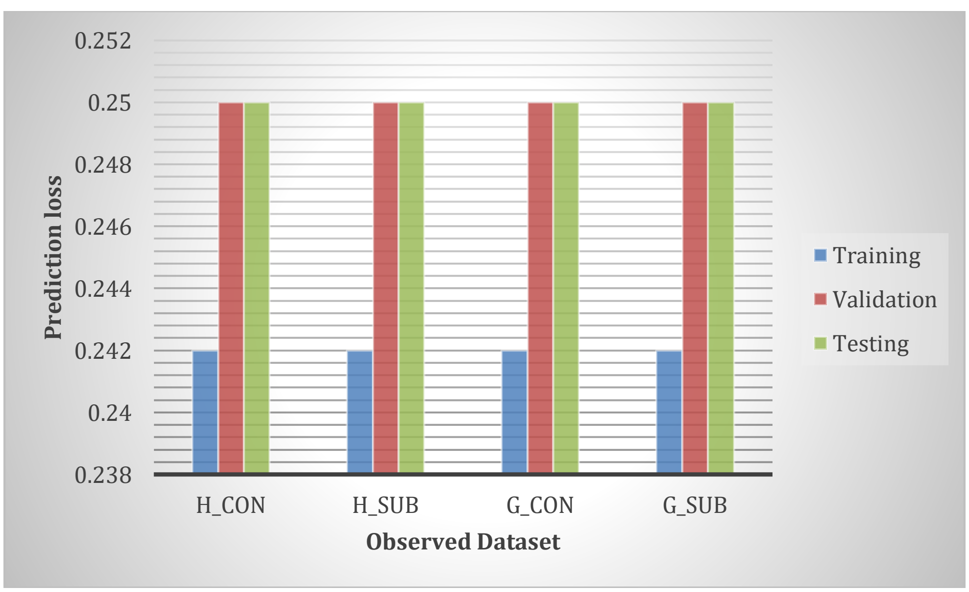
```
Prediction Loss of Training Set: 0.24216909999999975
Prediction Loss of Validation Set: 0.25
Prediction Loss of Testing Set: 0.25
```

General Performance for GSC Concatenated Features

```
Prediction Loss of Training Set: 0.24009999999999918
Prediction Loss of Validation Set: 0.25
Prediction Loss of Testing Set: 0.25
```

General Performance for GSC Subtracted Features

```
Prediction Loss of Training Set: 0.24009999999999918
Prediction Loss of Validation Set: 0.25
Prediction Loss of Testing Set: 0.25
```



4 Analysis of Results

After performing machine learning through different models it was seen that we achieved the highest accuracy when using the human observed features for closed form linear regression solution. Logistic regression also showed good performance. However, the GSC features could not show performance up to the mark. This might have happened as we worked with 100 features instead of 512 features to meet up space and time requirements. Neural network also could have showed a better performance if we had used more hidden layers.