**CSE 574 Introduction to Machine Learning**

**Project 2 Report**

The task was to find similarity between the handwritten samples of the known and the questioned writer by using linear and logistic regression as well as neural network.

The features are obtained from two different sources:

1. Human Observed features: Features entered by human document examiners manually

2. GSC features: Features extracted using Gradient Structural Concavity (GSC) algorithm.

The target values are scalars that can take two values f1: same writer, 0: different writers. Although the training target values are discrete we use linear regression to obtain real values which is more useful for finding similarity (avoids collision into only two possible values).

1. Human Observed features

There are total of 18 features for a pair of handwritten AND sample (9 features for each sample). The entire dataset consists of 791 same writer pairs and 293,032 different writer pairs(rows). Here we consider 791 same writer pairs and 791 different writer pairs so as to ensure that there is no bias and both classes are taken in equal probability.

Setting 1(concatenation)

Dataset size – 1582 rows and 21 columns (18 features ,2 image, 1 target)

Setting 2(subtraction)

Dataset size – 1582 rows and 12 columns (9 features ,2 image, 1 target)

2. Gradient Structural Concavity features

Gradient Structural Concavity algorithm generates 512 sized feature vectors for an input handwritten AND image. The entire dataset consists of 71,531 same writer pairs and 762,557 different writer pairs(rows). Here we consider 10000 same writer pairs and 10000 different writer pairs so as to ensure that there is no bias and both classes are taken in equal probability.

Setting 1(concatenation)

Dataset size – 20000 rows and 1027 columns (1024 features ,2 image, 1 target)

Setting 2(subtraction)

Dataset size – 20000 rows and 515 columns (512 features ,2 image, 1 target)

The pipeline for processing and running linear and logistic regression is as follows:

1. First, we read the files and create the 4 required datasets
2. Then we do partition the data
3. Next, we use gradient descent for linear regression and then tune the hyperparameters.
4. Lastly, we use logistic regression and predict a value between 0 and 1 for the given problem and then tune the hyperparameters as required.

**1 - Data Processing**

1. We have 4 datasets called matrixhod1.csv, matrixhod2.csv and matrixgsc1.csv, matrixgsc2.csv for Setting 1 and 2 respectively. We have generated these datasets from the same pair and different pairs. We read this dataset and separate the target and the features and divide these two datasets into training set, testing set, and validation set. The division is 80%, 10% and 10% for training set, testing set, and validation set respectively.

Training set is used to train the model and then the regression model that we created is tested on the testing set.

Validation Set

We create a validation set so that the model can be tested on unseen data and validation error is generated

**2 – Linear Regression Solution**

Why the need for Gradient Descent Solution?

We have used closed form solution or pseudo inverse method, so what is the need for gradient descent. In pseudo inverse method, we take the inverse of the covariance matrix.

This may not be possible, if the matrix in non-invertible. If that is the case, we will be unable to perform pseudo inverse method. Also calculating inverse of large matrix is very difficult and takes a lot of computations. For that reason, we use Gradient descent solution. Gradient descent is an efficient optimization algorithm that attempts to find a local or global minimum of a function.

Our linear regression function t has the form:

t = wT X

where w = (w1, w2..., wm­­) is a weight vector to be learnt from training samples and X is a vector of M features.

1. **Calculating Training Design Matrix XTR, Testing Design Matrix XTE and Validation Design Matrix XVA**

We calculate Training Design Matrix XTR, Testing Design Matrix XTR and Validation Design Matrix XVA as follows

1. **Training Design Matrix XTR, Testing Design Matrix XTE and Validation Design Matrix XVA**

Here are the steps for calculating the weights:

1. Initialize weights

We randomly initialize the weights as a vector of zeroes. We could use randomly initialize W by any value.

1. Update weights iteratively

We update the weights iteratively i.e. for each datapoint we update all the weights.

It is as follows:

w1= w1 +

.

.

wn= wn +

Here we are updating all the weights iteratively by adding the product of the learning rate and the partial derivative of the error function. The partial derivative is as follows:

2

So, the updated weights w1 …w10 will be as follows:

w1= w1 +

.

.

wn= wn +

1. Calculating error

ERMS=

When the model is trained using the training data and the model is functioning, we have no way of knowing whether the predicted value is correct. So, to test that out we use the error function.

In this project for linear regression we are using root mean squared error or ERMS. We calculate the ERMS for the testing, training and validation set

1. **Results & Observations**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | HOD Setting 1 | HOD Setting 2 | GSC Setting 1 | GSC Setting 2 |
| Learning Rate | 0.005 | 0.005 | 0.005 | 0.005 |
| Number of Epochs | 3000 | 3000 | 300 | 300 |
| ERMS | 0.49 | 0.505 | 0.46 | 0.484 |
| Accuracy | 56.96% | 56.31% | 66.65% | 61.4% |

Table 1.1 Results for Linear Regression

Observations-

As we can see from the above table which is for linear regression using gradient descent, the ERMS and accuracy for GSC dataset is much better as compared to Human observed dataset. This is mainly because the GSC dataset is richer in terms of data and has more features and more data rows to train on. Both the settings i.e. (concatenate and subtraction) give pretty similar values in terms of ERMS and accuracy.

**3 - Logistic Regression Solution**

What is Logistic Regression?

Logistic Regression is basically a binary classifier which estimates or predicts 0 or 1 values. Since we want to know whether the image has come from the same writer or the different writer we logistic regression. The equation for logistic regression is as follows:

t = wT X

1. **Calculating Training Design Matrix XTR, Testing Design Matrix XTE and Validation Design Matrix XVA**

The Calculating Training Design Matrix (X)TR, Testing Design Matrix (X)TE and Validation Design Matrix (X)VA also remain the same for logistic regression solution.

1. **Training Design Matrix XTR, Testing Design Matrix XTE and Validation Design Matrix XVA**

Here are the steps for calculating the weights:

1. Initialize weights

This step remains same as for linear regression using gradient descent solution.

1. Update weights iteratively

We update the weights iteratively i.e. for each datapoint we update all the weights.

Here we are updating all the weights iteratively by adding the product of the learning rate and the partial derivative of the error function. The error function for logistic regression changes to cross entropy.

The sigmoid function is given by:

Sigmoid = =

1. Calculating Cost

When the model is trained using the training data and the model is functioning, we have no way of knowing whether the predicted value is correct. So, to test that out we use the error function. The error function or cost function is called as cross entropy.

The cost function is as follows:

**(c) Results & Observations**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | HOD Setting 1 | HOD Setting 2 | GSC Setting 1 | GSC Setting 2 |
| Learning Rate | 0.03 | 0.03 | 0.03 | 0.03 |
| Number of Epochs | 2000 | 1000 | 1000 | 1000 |
| Accuracy | 56.20% | 58.86% | 59% | 66.84% |

Table 1.2 Results for Logistic Regression

Observations –

The GSC dataset is richer in terms of data and has more features and more data rows to train on, so the accuracy for logistic regression is much better using the GSC dataset .it was observed that Setting 2 i.e. (subtraction) provides better accuracy. Logistic regression provides much more accurate result as compared to linear regression using gradient descent.

References:

1. https://ml-cheatsheet.readthedocs.io/en/latest/logistic\_regression.html
2. https://en.wikipedia.org/wiki/Logistic\_regression