Programming Assignment 4: Linear Classification, Linear and Logistic Regression

Kavya Sethuram(ksetura@usc.edu); Rasika Guru (rguru@usc.edu); Roshani Mangalore (rmangalo@usc.edu)

1. Implementation of Linear Classification, Linear and Logistic Regression in python v2.7:

A. DataStructures used in the Implementation are:

- ***Lists:*** Python has a great built-in list type named "list". List literals are written within square brackets [ ]
- ***Dataframe*****:** Data Frame is a 2-dimensional labeled data structure and is the most commonly used panda object. We have generated Dataframe from the datafile using pandas.

B. Explanation:

Linear Classification – Perceptron Algorithm:

1. Start with data set X. Initialize random weight vectors as per the dimensionality given.
2. Add padding x0 to the dataset X
3. Calculate the dot product of the dataset with the weight vectors.
4. Check if the dot product is greater/lesser than zero.
5. If it is greater than zero and the classifying label (given in the input) is greater than zero then the classification holds good. The same applies if the product is less than zero.
6. Else re compute the weight vectors. For this choose a small Eta value say 0.0.1.
7. Weight vector would be sum/difference of the previous weight vector and product of eta and data set.
8. Repeat steps 3 to 7 for all the datapoints in the data set.
9. Output displays the weights for which the algorithm converges and the classifier for each data point.

Linear Classification – Pocket Algorithm:

1. Randomly initialize the weight vector, W

2. Add a padding value of 1 uniformly to all the input vectors, X

3. Calculate the activation function which is a product of the input vector and the transpose of weight vector, XW^T

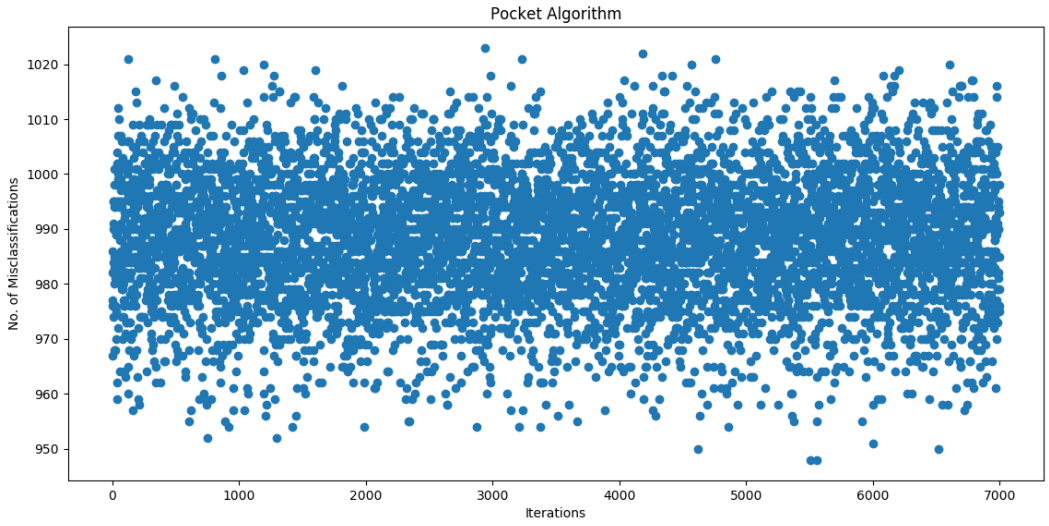
4. If the value of the function is greater than the threshold(zero in this case), we predict the output label as 1. Else it belongs to the negative class(Label = -1)

5. If the predicted class matches the actual class of the input vector, we proceed with the iteration

6. If the predicted class is greater or lesser than the actual class, we update the weight vectors accordingly.
    a. If the predicted class is +1; actual class is -1: W = W – (learning rate*X)
    b. If the predicted class is +1; actual class is +1: W = W + (learning rate*X)

7. The above process is repeated for 7000 iterations

Since pocket algorithm is not a promise problem, pure classification of all the input labels is not guaranteed. The number of misclassifications for each iteration is plotted.



Linear Regression Algorithm:

1. We can pack all response values for all observations into a n-dimensional vector called the response vector:

$$Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \cdots \\ \cdots \\ \cdots \\ \cdots \\ Y_n \end{pmatrix}$$

2. We can pack all predictors into a $n \times p+1$ matrix called the design matrix:

$$X = \begin{pmatrix} 1 & X_{11} & X_{12} & \cdots X_{1p} \\ 1 & X_{21} & X_{22} & \cdots X_{2p} \\ & & \cdots & \\ & & \cdots & \\ 1 & X_{n1} & X_{n2} & \cdots X_{np} \end{pmatrix}$$

Note the initial column of 1's. The reason for this will become clear shortly.

3. Compute Weights using the below Formula $W = (X^0X)^{-1}X^0Y$

X^0X and $(X^0X)^{-1}$ are $p+1 \times p+1$ symmetric matrices.

X^0Y is a $p+1$ dimensional vector.

4. The fitted values are

$$\hat{Y} = XW = X(X^0X)^{-1}X^0Y,$$

Logistic Regression Algorithm:

1. Randomly initialize the weight vector, W.

2. Add padding value of 1 uniformly to all the input vectors, X.

3. Calculate the gradient of the in sample error using the below formula, where N is the size of the input vector; y_i is the class label corresponding to $X^{(i)}$.

$$-1/N(\sum_{i=1}^{N} \frac{1}{1+e^{y_i W^T X^{(i)}}}).y_i.\ X^{(i)}$$

4. Update the weight vector as:

W = W- (learning rate * gradient of the in sample error)

5. Calculate the probability using the sigmoid function. Depending on the value of the probability assign the appropriate label values to the inputs.

Sigmoid Function: $\frac{e^s}{1+e^s}$

6. Calculate the accuracy as : (Number of inputs with accurately classified labels) / (Total no. of inputs)

C. Code Level Optimizations:

- Using NumPy comprehensions over explicit for loops in certain places, making code more compact and execution faster.
- Matrix operations are easy with NumPy arrays
- Modularizing parts of code into methods, making code more readable.

D. Challenges:

- Deciding on the convergence condition for Perceptron algorithm was tricky.
- Understanding the flow of logistic Regression with respect to the Implementation of the code was a little hard

2. Software Familiarization:

We have used the following packages to implement Linear Classification and Linear Regression

NumPy

- NumPy is a Numeric Python module. It provides fast mathematical functions.
- Numpy provides robust data structures for efficient computation of multi-dimensional arrays & matrices.
- We used numpy to read data files into numpy arrays and data manipulation.

Pandas

- Provides DataFrame Object for data manipulation
- Provides reading & writing data b/w different files.
- DataFrames can hold different types data of multidimensional arrays.

pylab*:* pylab is the library used to plot graphs. The word lists are represented in 2D space.

Existing Libraries:

1. Scikit-Learn (for Linear Classification)

    - It's a machine learning library. It includes various machine learning algorithms.
    - It uses the following inbuilt library to implement perceptron algorithm
      from sklearn.linear_model import Perceptron: sklearn.linear_model.Perceptron

      In scikit-learn, Perceptron is implemented as a transformer object to perform binary classification.

```
from sklearn import linear_model
# Build a classification task using 3 informative
features
X, y = make_classification(n_samples=1000,
n_features=10,
n_informative=3,
n_redundant=0,
n_repeated=0,
n_classes=2,
random_state=0,
shuffle=False)
per = linear_model.Perceptron()
```

2. Scikit-Learn (for Linear Regression)

- It's a machine learning library. It includes various machine learning algorithms.
- It uses the following inbuilt library to implement linear regression algorithm

```
from sklearn.linear_model import LinearRegression
```

```
from sklearn.linear_model import LinearRegression
import numpy as np
import pandas as pd


if __name__=="__main__":
    data=pd.read_csv("linear-regression.txt",sep=',',header=None)
    dataarray=np.array(data)

    Y_Train=np.array(dataarray)[:,2]
    X=np.array(dataarray)[:,0:2]

    #Add new column X0=1 for padding
    n, m = X.shape  # for generality
    X0 = np.ones((n, 1))
    X_Train = np.hstack((X0,X))
linreg = LinearRegression()
linreg.fit(X_Train, Y_Train)
score= linreg.score(X_Train,Y_Train)

print score*100
```

3. Scikit-Learn (for Logistic Regression)

- It's a machine learning library. It includes various machine learning algorithms.
- It uses the following inbuilt library to implement logistic regression algorithm

```
from sklearn.linear_model import LogisticRegression
```

```
from sklearn.linear_model import LogisticRegression
import numpy as np
import pandas as pd

if __name__ == "__main__":
    data = pd.read_csv("classification.txt", sep=',', header=None)
    dataarray = np.array(data)
    Y_Train = np.array(dataarray)[:,4]
    X = np.array(dataarray)[:, 0:3]

    # Add new column X0=1
    n, m = X.shape  # for generality
    X0 = np.ones((n, 1))
    X_Train = np.hstack((X0, X))

logreg = LogisticRegression()
logreg.fit(X_Train, Y_Train)
score=logreg.score(X_Train,Y_Train)

print score*100
```

3. Applications:

Linear Classification:

Linear Classification has been used to to classify an individual's response to a proposed loading dose regimen of Warfarin. The aim of the loading regimen is to bring the International Normalisation ratio into the required therapeutic range. This study proposes a method for pre-classifying an individual's response to the proposed loading regimen. The method takes into account other variables that may influence the response and with a high degree of accuracy classes the potential response as being either below therapeutic range, within or above the therapeutic range [References]: https://arxiv.org/ftp/arxiv/papers/1211/1211.2945.pdf

Linear Regression:

Multiple linear regression models can be used in the performance analysis of traffic rules. This study mainly discusses the overtaking and lane-changing problem in highway, the differences between traveling on the left and on the right lane. In order to promote traffic smooth effectively, overtaking model is constructed. The left lane changing rule considers its influence on human heart, and then multiple linear regression models. In the country with the cars driving on the left, this model is especially suitable for the near S shaped highway. By studying the joint influence of the curve radius of curvature, steering angle and the road friction coefficient on the driving speed of the curve and the obtained speed is within the speed limit. Therefore, after improving the overtaking model, the left lane vehicle also applies to the lane-changing rule. [Reference]:http://www.jocpr.com/articles/application-of-multiple-linear-regression-model-in-the-performance-analysis-of-traffic-rules.pdf

Logistic Regression:

Logistic Regression is used in Image Segmentation. This method is based on feeding artificial features to a framework of logistic regression classifier with ℓ_1 norm regularization and Markov Random Field prior, which has been studied, e.g., in hyperspectral image segmentation. The novelty of this approach stems from the use of a generic artificial feature set and the embedded feature selection property of the sparse logistic regression framework, which avoids application specific feature engineering. This method generates a large set of artificial features and passes them to a ℓ_1 regularized logistic regression classifier. Finally, a spatial prior imposes additional homogeneity. [Reference]: http://ieeexplore.ieee.org/document/6334177/

4. Individual Contribution

Perceptron Algorithm	Kavya Sethuram
Pocket Algorithm	Rasika Guru
Linear Regression	Roshani Mangalore
Logistic Regression	Rasika Guru and Roshani Mangalore