IE 506: Machine Learning - Principles and Techniques Jan-May 2023

Challenging Assignment : Multi Label Classification

Boddu Siva Venkata Sri Teja	**22N0459**

1 Introduction

Multi-label classification is a type of supervised learning problem in which an instance can be assigned multiple labels simultaneously. In contrast to traditional binary or multi-class classification, where each in- stance can only be assigned to one class, multi-label classification aims to predict the presence or absence of multiple labels for a given input. Our problem contains the data in which each data point is labeled with more than one labels. In our approach we solve the problem using ”one vs rest” approach.

2 Problem Statement:

We are given a text file of the data set where each line of the file contains the data point and correspoding labels assigned to the data point. This line is in the form of:
multi labels FeatureID:val FeatureID:val FeatureID:val ... FeatureID:val

1. our first objective is to preprocess the data and change it to suitable form for our model.
2. Our second objective is to Train an adaboost model using this data while using following accuracy and F1 score:
   F1 score= $\frac{1}{} * \sum_n$ $\frac{|y^i \cap y\hat{}_i|}{}$
   Accuracy= $\frac{1}{} * \sum_n$ $\frac{}{|y^i \cap y\hat{}_i|}$ $\frac{}{}$
3. We are to compute best hyper parameters using k fold cross validation.

3 Solution Approach:

We are to train a model which takes a data point as an input and predict the possible labels assigned to it. To encounter this problem our approach is to ”1 vs rest” method where we use Adaboost classifier (written code from scratch) for each class. We train binary classifiers corresponding to each label.It can be understood by the following procedure:

1. for label in labels:
   (a) input space:$D = (x^i, y^i)$, x^i $\mathbf{R^d}$, and y^i $\in \{+1, -1\}$ $\forall \in \{1, 2, \ldots N\}$ where y^i is 1 if label is present in label set of data point otherwise -1
   (b) initialize weight vector $w_i = \frac{1}{}$ $\forall \in \{1, 2, \ldots N\}$
   N
   (c) for rounds=1,2,3 . . . T :

- Train a weak classifier h(x) with examples weighted using current weights w^t :

$$\epsilon_t = \sum_{i=1}^{n} w_i^t \, \mathbf{I}(h_t(x^i) \neq y^i) \quad (1)$$

- Compute

$$\alpha_t = \frac{1}{2} * \ln \frac{1 - \epsilon_t}{\epsilon_t} \quad (2)$$

- update weight vector as follows:

$$w_i^{t+1} = w_i^t * \exp(-1 * \alpha_t * y^i * h_t(x^i)) \quad (3)$$

- normalize the weight vector:

$$w_i^{t+1} = \frac{w_i^{t+1}}{\sum_{i=1}^{N} w_i^{t+1}} \quad (4)$$

(d) Now we aggregate our week classifiers as follows:

$$\mathbf{H}(\mathbf{x}) = \mathbf{sign}\left(\sum_{i=1}^{T} \alpha_t h_t(x^i)\right) \quad (5)$$

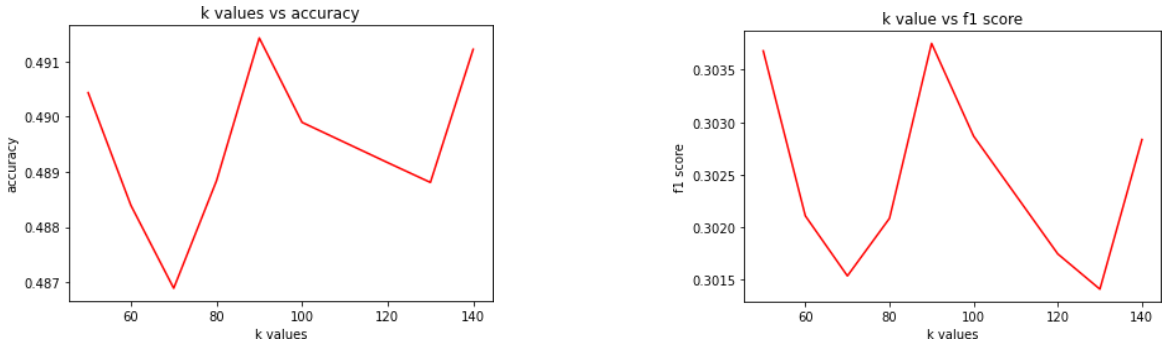
(e) H(X) is our prediction by our one class binary classifier for our data instance x^i

2. Now we have prediction of data point by k classfiers then we aggregate these predictions and create a set of predictions.

4 K fold cross validation for hyper parameter tunning:

In adaboost we train k week base classifiers and then aggregate them to preditct the label. So we are re- quired of best k hyperparameter that give us best results over test data. TO encounter this situation we use k fold cross validation. It is explained in the following procedure:

1. create two lists of f1 score and accuracy to store mean accuracy and mean f1 score correponding to each k value
2. for k in list of different k values:
3. create 2 empty lists of f1 score and accuracy
   (a) for seed in different seed values:
      - sample the data using seed value
      - split the data in train test split
      - train the model using train data
      - compute the predictions for test data
      - compute accuracy and f1 score using predictions and actual labels of the test data
      - store it in lists of f1 score and accuracy lists
   (b) store the mean f1 score and mean accuracy in lists corresponding to k value
4. best k value is the k who has maximum f1 score and accuracy (figure 1).



(a) K value vs accuracy (b) k value vs f1 score

Figure 1: K cross validation to find best k

5 Experiments:

As we mentioned above that best k value (obtained by tunning of hyperparameter using k cross valida- tion) is obtained 90. So now we proceed with value 90 of k.
Next procedure of code can be understood by the following procedure:

1. At first, we train our model using training data set(multilabel train data text file) and obtain classi- fiers and alpha values corresponding to each label
2. now using aggregation function over alphas and classifiers over test data (sample test data file) gives us prediction of each class label
3. now we predict labels of the data by creating the set of predicted labels corresponding to each data set.
4. now using function of accuracy and f1 score we compute f1 score and accuracy for predictio of test data set.

6 Code Introduction:

different tasks in code file can be explained using follwoing points:

- To input any text file which is similar to given files can be easily read by code file.
- code is written in such way that giving data file directly in the form of dataframe , returns the accu- racy corresponding to each individuals and combined predictions in the form of list of sets.
- Available functions of f1 score and accuracy takes input as numpy array.