Aggressive Sampling for Multi-class to Binary Reduction with Applications to Text

IMPLEMENTATION REPORT

Apoorva Manda
Dept. of Computer Science
Clemson University
Clemson, US
manda@g.clemson.edu

Aradhana Velidimalla
Dept. of Computer Science
Clemson University
Clemson, US
avelidi@g.clemson.edu

I. OVERVIEW

The paper "Aggressive Sampling for Multi-class to Binary Reduction with Applications to Text Classification" [1] addresses the multiclass-classification problem for large number of classes. In this paper an aggressive double sampling strategy is adopted on inter-dependent paired sub samples of text repository DMOZ composed of 100,000 classes, with an aim to reduce the curse of long-tailed distribution problem. This is achieved by obtaining the training set as a combination of oversampling of small classes and sub-sampling of large classes. The evaluation parameters considered for the proposed methodology were: utilization of memory, accuracy, training time, prediction time, predictive performance.

The reduction of doubly sampled multi- class to binary was carried out by implementing the following sub procedures:

1) Reduction of Multi-class to binary over pairs of examples
2) Aggressive Double Sampling
3) Prediction with Candidate Selection.

II. EXPERIMENTAL SETUP

DATASET

In this project the WIKIPEDIA dataset is being evaluated using the methodology proposed in the paper. The dataset used is pre-processed and is in .tf format (term frequency file). libSVM format is followed by each datafile. In data file, each line represents a sparse document vector in the following format: <label feature: value, feature: value …>
Where,
label represents the category to which the document vector belongs and is an integer.
The pair feature: value represents a non-zero feature with index = feature and value= Value. feature is an integer corresponding to a term and value is a double that corresponds to the term frequencies (weight) as shown in Figure1.

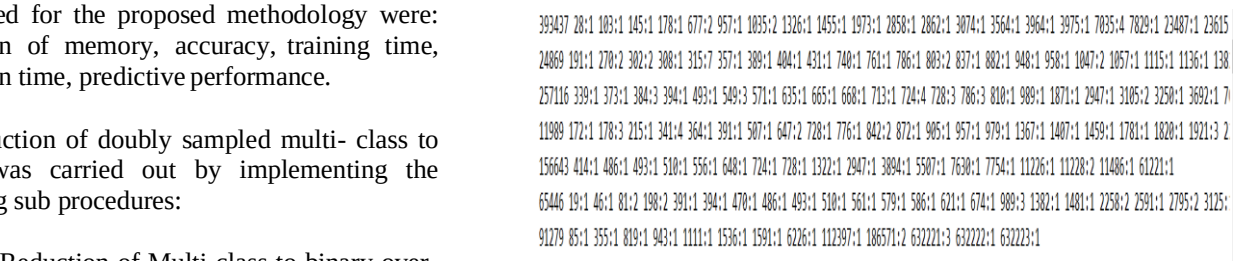


Figure1: Dataset in .tf format

The WIKIPEDIA dataset consists of remapped versions of the data in training and testing .tf files for computation.

EVALUATION MEASURES

The evaluation parameters we used over the test set are accuracy, F1 measure, Logarithmic Loss, Mean Squared Error and Mean Absolute Error.

F1 measure is defined as the harmonic average of precision and recall. The performance is measured with the following relation: The higher the F1 the better the performance.

Mean Absolute Error is the average of all absolute errors where absolute error is the difference between predicted value and actual value.

Mean Squared Error is the mean of the individual squared error. It provides information about the fit of the line.

Logarithmic loss is a measure of uncertainty of the predicted value on the basis of its variation from the actual label value.

The time taken for loading, pre-processing, binary reduction, learning of the algorithm, prediction along with the memory usages are also provided.

PLATFORM

In our experiment we used machines with the following specifications:
Operating System: Windows 10
System Type: 64-bit Operating System, x64-based processor
RAM:8 GB
Processor: Intel Core i5 / i7

III. IMPLEMENTATION

This project is being implemented in mainly 5 phases:

1) Loading of the Data
2) Pre-Processing of the data
3) Reduction
4) Algorithm learning
5) Prediction

The following are passed as argument vector parameters while running the file: Sample number, Sampling rate, number of candidates
Where,
Sample number denotes the average number of examples sampled per class.

Sampling rate is the rate with which the classes are chosen for sampling. A lower bound value for sampling rate is set as 1 / Size of class. In case if user enters a value lower than this then by default one class will be chosen.

Number of candidates: Number of candidate classes for prediction.

LOADING OF DATA:

The test and train files are loaded in this phase. The time for loading the dataset is also being calculated.

PRE-PROCESSING OF DATA:

In this phase, we make use of the TfidTransformer which is used to transform a count matrix to term-frequency times inverse document-frequency representation. It takes the parameters- matrix of terms and a Boolean value 'copy' and returns a vector of sparse matrix. Then the mean vector for each class and one global mean for the entire collection is calculated by making use of the csr_matrix and coo_matrix which are in-turn used to determine the centroids of the entire dataset and the class.

REDUCTION:

In this phase the multi class to binary reduction is being carried out. This is followed by the implementation of double sampling strategy. This strategy is executed in majorly 2 steps:

1)From the given class, a sub sample of examples is chosen at random with a probability p.

2)From these examples, adversarial classes are drawn uniformly.

The binary reduction produces binary labels and binary features. The binary features produced are using logarithmic functions like log $(1 + y_t)$, $\log(1+(L_s/f_T))$, centroid distances – d $(x^y$, centroid(y)), ranking functions like BM25 etc

ALGORITHM LEARNING:

In this phase the binary features and the binary_labels are passed as parameters and these are fit using the Linear Support Vector Classification algorithm. The best fit of the classification is returned by updating the weights using the binary labels and features so obtained.

PREDICTION:

The prediction is carried out by implementing the nearest neighbour classifier and brute force algorithm for fitting the class centroids in the feature space

IV. ANALYSIS & RESULTS

CHALLENGES IN DATASETS PROCUREMENT

Since the double sampling strategy mainly addresses the multiclass-classification problem for large number of classes, finding such open source datasets in field of Image/Object recognition, Human Activity recognition etc with more than 10,000 classes was a difficult task.

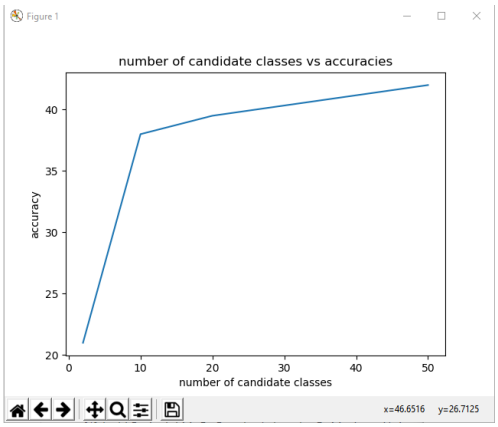
When searched for the same in Kaggle, UCI repositories, the maximum number of classes for the above-mentioned datasets were found to be 51. The challenge posed was that the datasets with such a smaller number of classes would give an extremely low accuracy as the feature selection from 51 classes would produce hardly any effective feature after aggressive double sampling. Also, such datasets with a smaller number of classes were not available in .tf format.

The datasets with large number of classes could only be found in the field of Text processing and could be procured from large data source repositories like Directory Mozilla Firefox, Wikipedia, Yahoo etc and we have chosen the WIKIPEDIA Dataset.

The WIKIPEDIA dataset chosen is the one containing testing, training datasets with more than 45k classes in each.

ANALYSIS OF VARIOUS PERFORMACE EVALUATORS

We have analysed the results (accuracies achieved) obtained by plotting a graph between accuracy and number of candidate classes considered as shown in the graph 1.



Graph1: Line graph between Candidates and accuracies.

The accuracy values obtained on different parametric values are can also be inferred from the Table1 below.

No. of samples	Sampling rate	No. of candidates	accuracy
1	0.001	2	21.95%
3	0.01	10	38%
5	0.1	20	39.5 %
12	0.1	50	42%

Table 1: Variations of the accuracy measures

The other parameters like time taken for loading, pre-processing, binary reduction, learning of the algorithm, accuracy, F1 score, Logarithmic loss, mean absolute error can be inferred from the figures below.

```
Time for Loading dataset: 20 Seconds
Time for preprocessing: 288 Seconds
Time for binary reduction: 1398 Seconds
Time for Loading learning: 2 Seconds
Accuracy: 0.38
F1 Score: 0.26
Logarithmic Loss: 4
Mean Absolute Error: 3.6
```

Figure 2

```
Time for Loading dataset: 17 Seconds
Time for preprocessing: 329 Seconds
Time for binary reduction: 3639 Seconds
Time for Loading learning: 4 Seconds
Accuracy: 0.21
F1 Score: 0.13
Logarithmic Loss: 6
Mean Absolute Error: 3.8
```

Figure 3

V. CONCLUSION

Upon examining the above test results, it can be inferred that with increase in number of candidates and number of samples the accuracy increases marginally.

The double sampling and binary reduction approach make it highly suitable for large class scenario by contributing significantly towards the performance in terms of memory usage and total runtime.

VI. REFERENCES

[1] Bikash Joshi et al., on "Aggressive Double Sampling for Multi-Class To Binary Reduction with Applications To Text Classification", Advances in Neural Information Processing Systems, 4159-4168.

[2] WIKIPEDIA DATASETS https://drive.google.com/open?id=1iAVpPOp9GbAkAM88FGEHTkEe4vYbVY-4