Modified Logarithmic Time Online Multiclass Train and Prediction Spring 2018

Instructor: Anna Choromanska
Author: Jiadong Chen and Jiawang Wang

I. ABSTRACT

Logarithmic depth trees can achieve the goal of obtaining train and test time complexity logarithmic in the number of classes. The central problem of constructing a logarithmic depth tree is to keep the tree balanced during the online training process. Based on the algorithm, we develop a new approach to keep the tree balanced instead of swapping orphan nodes. We give each tree node a new attribute called 'suspend' and realize the balanced tree by updating this attribute. There are totally two datasets: Aloi and Isolet used in our train and test experiment. Our extensive experiments demonstrate that our method to construct a logarithm tree is effective, flexible and plausible.

II. INTRODUCTION

By problem transformation techniques, people reduces the problems of multiclass classification to multiple binary classification problems. There are two major algorithms for this strategy: one-against-all algorithm and one-against-one algorithm. Compared with one-against-all algorithm with O(k) running time, LOMtree algorithm based on decision tree can reach O(log(k)) computational time per example for both training and testing. The main problem for decision tree is to find the best partition to each tree node. Thus,this multiclass classification problem is simplified as a binary problem for each node.

In the online machine learning setting, unlike batch learning techniques which generate the best predictor by learning on the whole training data set at once, data becomes available in a sequential order and is used to update our best predictor for future data at each step. In addition, when constructing a logarithmic depth tree, each data only traverses the tree once. The main problem caused by this setting is that classes contained in the node will change from time to time with expectation changing during data flows. Meanwhile, keeping the tree balanced is necessary to achieving O(log(k)).

The easiest way to solve this problem is to delete or add a class when it leaves or comes. However, in order to keep some important information and make it faster when the class comes to a node again. We use 'suspend' attribute in the node to collect these data.

The algorithm in Section 3.A describes the framework of our modified LOM algorithm. To solve the problem that some nodes are initially created but finally samples go to the other side, we keep some information in their parent nodes and set 'suspend'attribute.The objective and the way we keep the tree balanced are introduced in section 3.B. It is worth mentioning that the 'suspend'action which is critical in our algorithm shows in section 3.C. In section 4,we take an experiment on two different datasets: Isolet and Aloi and prove our two hypotheses.

III. Modified LOM (MLOM) Tree Algorithm

A. Framework

MLOM tree algorithm takes in online samples and construct a tree in process of training. It will also try to balance and purify itself.

Algorithm 1 MLOM algorithm training part

For each sample:

1: Register a class if the data belongs to a new class
2: Determine the way (left, right) where data will go next by the objective $J(h)$ as (1)
3: Train the node using this data
4: Update expectation and class information in this node
5: Give birth if second class arrives at this node
6: Suspend a class recursively if this class no longer goes to a child of this node
7: Recursively process each node all the way to leaves

B. Balance Analysis on Objective Function

The objective function can be written as:

$$J(h) = \sum_{i=1}^{k} \pi_i |P(h(x) > 0) - P(h(x) > 0|i)| \quad (1)$$

If we use uniform π_i for all classes in node, we would gain a balance w.r.t classes instead of a balance w.r.t samples.

This can lead to a perfect balance in practice, i.e., after we cast Aloi dataset on our MLOMtree, the root have 1000 classes and both its childs have exactly 500 classes, and even their childs have exactly 250 classes. This is quite a satisfactory achievement. However, this can slightly decelerate the process, because it takes O(lg n) to update the median of expectations of classes in current node when one of the expectation is changed, where n is number of active classes in current node. When it comes to update the mean, it takes O(1).

C. Suspending

Each node contains a list of names of suspended classes which means that these classes no longer exists in this node.

There are three situations when processing suspended class:

1.If this node loses(suspends) its last second class, delete all children.

2.If this node loses its last class, the node will not be deleted, and nothing special will be done. This will lead to a special temporary status: the node includes no class, and its parent and sibling have the same classes. This temporary status will not last long, unless expectation of samples arrives at its parent keep increase monotonically, which is unlikely for a pre-normalized dataset. However, this temporary status may cause problem for testing, because a leaf of the decision tree even might not give out a single class, and the node will be fixed in online test function.

3.If this node is not a leaf or a parent of a leaf, then search recursively each node until find a parent of a leaf.

IV. EXPERIMENT

In our experiment, we mainly verify two things: 1. Modified LOM algorithm can also achieve true logarithmic time computation in practice. 2. Modified LOM algorithm can reach the same level of accuracy as LOM algorithm on the whole. We totally use two datasets to address our hypothesis: Isolet and Aloi. The datasets were divided into training (90%) and testing (10%). The details of the datasets are provided in Table 1

We compare MLOMtree with LOM and one-against-all classifier (OAA) implemented in the Vowpal Wabbit leaning system. We report train time, per-example test time and accuracy.

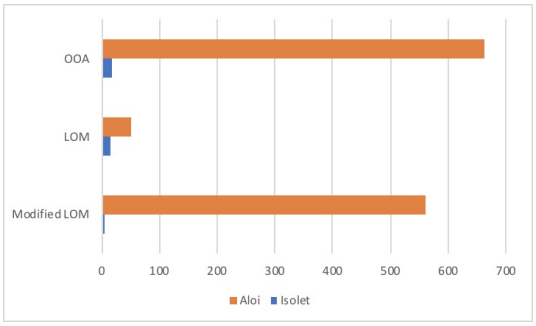

Figure 1. train time

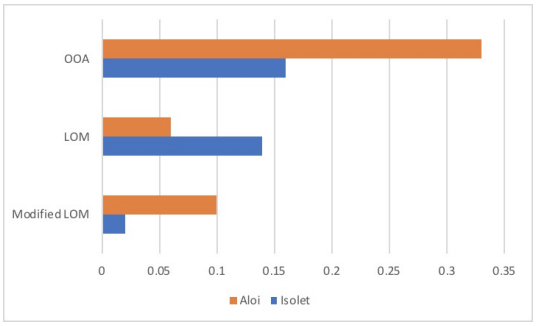

Figure 2. per-example test time

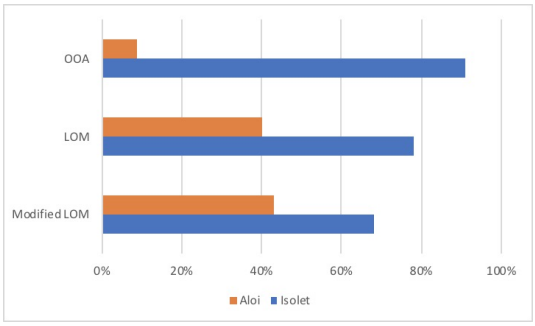

Figure 3. accuracy

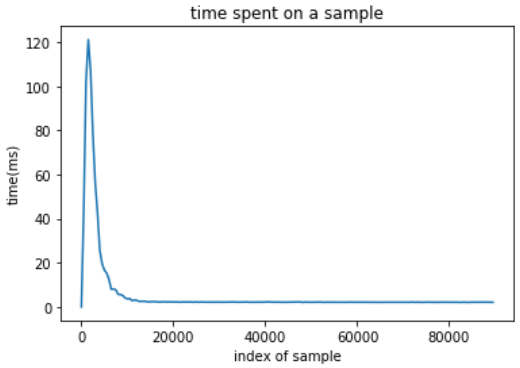

Figure 4. time spent on a sample for Aloi dataset

	Isolet	Aloi
Size	52.3MB	17.7MB
Features	617	128
Examples	7797	108K
Classes	26	1K

Table 1: Dataset sizes

In figure 1 and 2, we report respectively train time and per-example test time. Figure 3 is the accuracy for each dataset. These experiment results can verify our hypothesis 1 and 2. Comparing to LOM Tree algorithm, our running time can reach the same level.

From figure 1 we can see that, at the beginning of training, unfamiliar classes comes to nodes frequently, as well as a lot of 'flip flop'. Every time a node changes its mind on whether a class should go(left or right), this class would be suspended recursively. However, when the 'hard time' is over, the tree is quite balanced and nodes do not change their mind that frequently, thus time spending on a sample can keep O(log(k)).

V. CONCLUSION

Modified LOM Tree algorithm can also realize the same function which is to reduce a multiclass problem to a binary problem. The 'suspend' attribute and using the median of expectation guarantee balanced split which is critical to logarithmic tree algorithm. After analyzing and evaluating our hypothesis, we address that modified LOM Tree algorithm can reach logarithmic time and is also best available for multiclass classification problem.

REFERENCES

[1] Anna Choromanska and John Langford, Logarithmic time online multiclass prediction, CoRR, abs/1406.1822,2014

[2] R. Rifkin and A. Klautau. In defense of one-vs-all classification. J. Mach. Learn. Res., 5:101–141, 2004.

[3] J. Langford, L. Li, and A. Strehl. http://hunch.net/ vw, 2007.