

Group Project

December 5, 2023

Abstract

- 1 Introduction
- 2 Related work
- 3 Proposed algorithm: Instance Dependent Cost Online Classification

In this section, we design heuristics instance-dependent cost in online learning. To simplify cost matrix, we only consider the misclassification cost for each class (degree of freedom: $k^2 - k$ down to $k - 1$) and the misclassification cost for each instance (degree of freedom: $n \times (k^2 - k)$ down to $n \times (k - 1)$), where k is number of classes, n is number of instances.

3.1 The original structure of the proposed algorithm

For each prediction, if our classifier predict correctly, we continue to train the classifier by fitting this instance by the class-dependent cost; Otherwise, i.e. the current classifier can not predict this instance correctly, the instance is difficult to the classifier that it can not handle, our current classifier should pay more attention to this instance. We continue to train the classifier by fitting this instance by the heuristics instance-dependent cost.

Note that in algorithm 1 we take the prediction error into consideration when calculating the heuristics instance-dependent cost. The prediction error term focus more on the instance that p nears to 0.5 but is misclassified as the opposite class. For such instances, our current classifier is almost able to predict correctly, so we give them a higher prediction error cost. However, for the instance that p nears to 0 or 1 but is misclassified as the opposite class, there are two possible reasons: 1) our current classifier is still weak 2) the instance is a noise. For such instance, we give them a lower prediction error cost to prevent overfitting or learning from noise. By receiving the feedback from prediction, the model is

trained to strengthen the ability to correctly classify samples that are easy to classify incorrectly, thereby improving the overall performance of the classifier step by step.

Input: Input data, Input Labels

Output: Prediction

```

while Have more samples do
    datum, label <- next sample
    prediction label  $\hat{y}$  <- Predict by current classifier
    y <- the true label of this sample
    current ratio of Class y:  $CRC_y <- \frac{\# \text{ of current samples with label } y}{\# \text{ of all current samples}}$ 
    Class-dependent Cost:  $CDC_y <- \frac{1}{CRC_y}$ 
    if Prediction is correct then
        | train the classifier by fitting this instance with the
        | class-dependent cost
    else
        | prediction error <-  $\alpha e^{\beta(1-y(1-p)-(1-y)p)}$ 
        | Instance-dependent Cost:  $IDC <- CDC_y + \text{prediction error}$ 
        | train the classifier by fitting this instance with the heuristics
        | instance-dependent cost
    end
end

```

Algorithm 1: Heuristics Instance-dependent Cost Online Classification 1

3.2 Change the structure of the proposed algorithm

We have modified the structure of the original algorithm in two aspects, one is the cost of using feedback based on instance prediction results each timestep, and the other one is the modification in the definition of prediction error term.

For the first aspect, we do not distinguish between correctly classified and incorrectly classified instance, using the same definition of instance-dependent cost as the cost for each instance. Since we found that, if the classifier correctly classifies each instance, it will degenerate into class-dependent cost classifier, which shows no obvious difference between class-dependent cost classifier and instance-dependent cost classifier.

For the second aspect, the prediction error is redefined as:

$$\text{prediction error} = (1 - p_t)^\alpha, \quad (1)$$

where p_t is the classification probability as the true label, defined as:

$$p_t = \begin{cases} p, & \text{if } y=1 \\ 1-p, & \text{if } y=0 \end{cases} \quad (2)$$

In the above $y \in \{0, 1\}$ specifies the ground-truth class and $p \in [0, 1]$ is the model's estimated probability for the class with label $y = 1$.

Then the instance-dependent cost is redefined as

$$\text{instance-dependent cost} = (1 + \text{prediction error}) \times \text{class-dependent cost} \quad (3)$$

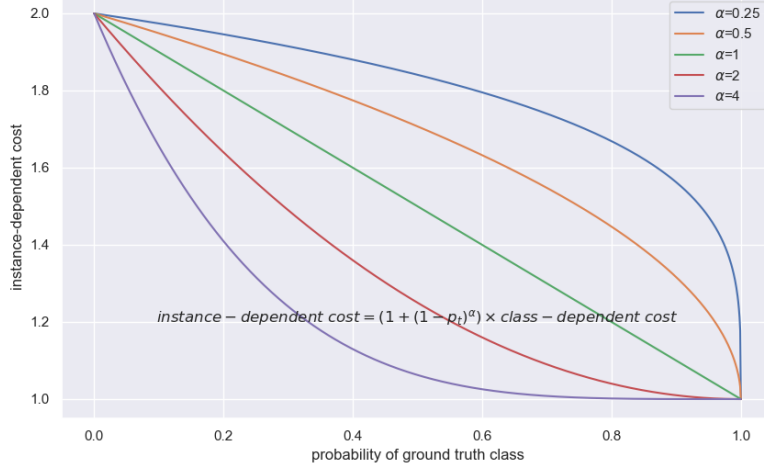


Figure 1: instance-dependent cost with different value of α

The instance-dependent cost is visualized for several values of $\alpha \in [0.25, 4]$ in 1. We note two properties of the instance-dependent cost. (1) When an instance is misclassified and p_t is small, the prediction error term nears to 1, which contributes a large instance-dependent cost. As p_t nears to 1, the prediction error goes to 0 and the instance-dependent cost for well-classified instance is down-weighted to class-dependent cost. (2) The hypothesis parameter α smoothly adjusts the rate at which hard instances are focused on more. Hence, compare with the former definition of prediction error focusing the instances with p_t nears to 0.5, the new definition of prediction error focuses on the instances with p_t nears to 0, which are considered as hard instances, and slightly adjusted by the hypothesis parameter α .

The algorithm with new structure is describe as follows:

4 Experiments

4.0.1 Experimental Setup

In this subsection, we illustrate our experimental dataset setting, parameter setting, and comparison methods.

For comparison method, we design a naive version of our method, which does not further fine-tuning on the human designed cost matrix. The human designed matrix is totally determined by tracked imbalance ratio .

Input: Input data, Input Labels

Output: Prediction

```
while Have more samples do
  datum, label <- next sample
  prediction label  $\hat{y}$  <- Predict by current classifier
  y <- the true label of this sample
  current ratio of Class y:  $CRC_y <- \frac{\# \text{ of current samples with label } y}{\# \text{ of all current samples}}$ 
  Class-dependent Cost:  $CDC_y <- \frac{1}{CRC_y}$ 
  if y is 1 then
    |  $p_t <- p$ 
  else
    |  $p_t <- 1 - p$ 
  end
  prediction error <-  $(1 - p_t)^\alpha$ 
  Instance-dependent Cost:  $IDC <- (1 + \text{prediction error}) \times CDC_y$ 
  train the classifier by fitting this instance with the heuristics
  instance-dependent cost
end
```

Algorithm 2: Heuristics Instance-dependent Cost Online Classification 2

Following existing works of online class imbalance learning, we select Hoeffding Tree as the base classifier and the number of base learners is set to 10. We set the interval between each evolution 100 data samples with a maximum generation of 5 in each evolution process. The number of individuals are set to 20 for all evolution processes. The metric we used for fitness evaluation is the difference between the maximum recall and the minimum recall.

One part of the datasets "contraceptive", "segment0", "yeast-0-2-5-6 vs 3-7-8-9" and "yeast1" come from Keel repository . Other datasets comes from synthesizing. All the dataset information are summarized in Table 1. Here IR means the imbalance ratio (ratio between instance number of maximum class and instance number of minimum class).

The metric we used is online GMean following , which give instantaneous GMean for each time step. The overall GMean is calculated by simply averaging the GMean on every time steps.

4.0.2 Overall Performance Comparison

We conduct the experiment on aforementioned datasets, the performances of our method and comparison method are summarized in Table 2.

5 Conclusion

dataset	IR	#instance	#feature	#class
synthesize1	5.0	4800	30	2
synthesize2	3.0	3000	10	3
synthesize3	3.0	4000	10	2
synthesize4	1.5	5000	10	2
synthesize5	3.0	4000	10	2
contraceptive	1.889	1473	9	3
segment0	6.015	2308	19	2
yeast-0-2-5-6 vs 3-7-8-9	9.141	1004	8	2
yeast1	2.459	1484	8	2

Table 1: Overview of used datasets

datasets	class-dependent cost	instance-dependent cost				
		$\alpha=0.25$	$\alpha=0.5$	$\alpha=1$	$\alpha=2$	$\alpha=4$
synthesize1	0.897	0.899	0.901	0.899	0.899	0.897
synthesize2	0.741	0.730	0.656	0.751	0.751	0.745
synthesize3	0.831	0.841	0.818	0.809	0.827	0.833
synthesize4	0.757	0.751	0.745	0.771	0.713	0.706
synthesize5	0.651	0.629	0.683	0.571	0.684	0.691
synthesize6	0.701	0.715	0.659	0.767	0.743	0.754
synthesize7	0.730	0.721	0.774	0.751	0.768	0.741
synthesize8	0.644	0.686	0.704	0.699	0.764	0.688

Table 2: Comparison with class-dependent cost in terms of overall GMean, the better algorithm is denoted as bold, synthesize benchmark

datasets	class-dependent cost	instance-dependent cost				
		$\alpha=0.25$	$\alpha=0.5$	$\alpha=1$	$\alpha=2$	$\alpha=4$
segment0	0.957	0.966	0.962	0.962	0.972	0.973
segment0-5-1tra	0.931	0.948	0.975	0.973	0.975	0.975
segment0-5-2tra	0.931	0.963	0.960	0.949	0.968	0.914
segment0-5-3tra	0.956	0.967	0.946	0.970	0.972	0.973
segment0-5-4tra	0.948	0.929	0.946	0.946	0.952	0.950
segment0-5-5tra	0.975	0.976	0.975	0.974	0.974	0.974
segment0-5-1tst	0.902	0.870	0.887	0.907	0.901	0.892
segment0-5-2tst	0.864	0.869	0.869	0.869	0.869	0.869
segment0-5-3tst	0.821	0.886	0.883	0.848	0.848	0.843
segment0-5-4tst	0.874	0.922	0.908	0.913	0.903	0.911
segment0-5-5tst	0.861	0.866	0.866	0.865	0.863	0.863

Table 3: Comparison with class-dependent cost in terms of overall GMean, the better algorithm is denoted as bold, segment benchmark

datasets	class-dependent cost	instance-dependent cost				
		$\alpha=0.25$	$\alpha=0.5$	$\alpha=1$	$\alpha=2$	$\alpha=4$
yeast1	0.598	0.661	0.655	0.636	0.643	0.671
yeast1-5-1tra	0.553	0.570	0.537	0.537	0.529	0.536
yeast1-5-2tra	0.617	0.623	0.601	0.610	0.619	0.598
yeast1-5-3tra	0.623	0.587	0.584	0.604	0.599	0.585
yeast1-5-4tra	0.566	0.570	0.676	0.653	0.650	0.651
yeast1-5-5tra	0.683	0.647	0.598	0.620	0.572	0.573
yeast1-5-1tst	0.602	0.608	0.612	0.625	0.625	0.637
yeast1-5-2tst	0.632	0.619	0.591	0.589	0.629	0.594
yeast1-5-3tst	0.455	0.471	0.433	0.447	0.449	0.431
yeast1-5-4tst	0.554	0.502	0.515	0.531	0.509	0.480
yeast1-5-5tst	0.492	0.625	0.616	0.591	0.581	0.578

Table 4: Comparison with class-dependent cost in terms of overall GMean, the better algorithm is denoted as bold, yeast benchmark