Instance-Level Cost-Sensitive Online Classification Algorithms for Class-Imbalanced Data Streams

Xian Shan [1, a] *, Jinyu You [1, b], Yu Xie [1, c], WenLei Xu [1, d], and Zongrui Li [2, e]

[1] * College of Science, China University of Petroleum (East China), Qingdao, 266580, China;

[2] School of Computer Science, China University of Petroleum (East China), Qingdao, 266580, China.

[a] 20120029@upc.edu.cn, [b] s21090088@s.upc.edu.cn, [c] z22090028@s.upc.edu.cn,

[d] 2309010223@s.upc.edu.cn, [e] 2107040114 @s.upc.edu.cn

Abstract. The importance of classification in machine learning is increasingly acknowledged in contemporary research and applications, such as disease detection, user analysis, etc. However, the efficacy of traditional classification algorithms is frequently affected by challenges such as class imbalance and the processing of large-scale dynamic data. To address these issues, inspired by the cost-sensitive learning strategy (Pinball loss) and instance-level loss function (Focal loss), this study constructs instance-level cost-sensitive loss functions by expanding the sparse and robust Hinge and Ramp loss. The new loss functions can better discern the difference between classes and samples. By integrating with the online SVM classification algorithm and using online gradient descent (OGD) algorithm, it can effectively deal with classification problems on class imbalanced data streams. Numerical experiments on UCI benchmark datasets validate their effectiveness.

Keywords: Classification; online learning; loss function; class imbalanced learning; cost-sensitive learning; instance level.

1. Introduction

In recent years, with the development of digital technology and the wide application of the Internet, data-driven decision-making has emerged as a key means to enhance economic benefits and reduce risks. As one of the core tasks of machine learning and data mining, classification has been widely used in fields like disease diagnosis and financial risk assessment [1]. However, traditional classification algorithms often show limitations when dealing with the large-scale, dynamically changing, and class imbalanced data streams that are commonly encountered nowadays [2].

On one hand, in this dynamic environment, the updating efficiency of classification algorithm becomes a key issue. Researchers put forward online learning technologies [3], whose dynamic updating ability enables them to effectively improve the computational efficiency of algorithms.

On the other hand, in class imbalanced datasets, the scale of some classes is significantly smaller than others, making it difficult to identify minority class samples. When class imbalance is severe, ignoring minority classes can often lead to high accuracy more simply. In addition, in practical applications such as medical diagnosis, the costs of false negative (FN) misclassification (missed diagnosis) and false positive (FP) misclassification (misdiagnosis) are often very different [4].

To address the class imbalance problem, researchers have focused on developing cost-sensitive learning (CSL) techniques that can accurately identify both majority and minority classes [5]. CSL addresses the class imbalance problem directly by adjusting the penalty parameters of algorithms or designing specific loss functions. The former imposes stricter penalties on misclassification of minority classes [6], while the latter assigns different error costs to each instance [7].

Although methods have been proposed to deal with data flow or class imbalance, most of them are not designed for dynamic class imbalance data flows. Building on previous studies, this paper innovatively improves the traditional models to solve the problem of class imbalance classification in data streams, and realizes the cost-sensitive online classification algorithm at the instance level. Finally, the effectiveness of the proposed method is verified through numerical experiments.

2. Instance-Level Cost-Sensitive Online Classification Algorithm

2.1 Related Research

In this study, a robust soft-margin kernel SVM is employed to search for decision functions [8] within a reproducing kernel Hilbert space H. This mapping transforms the features of samples ($x_t \in R^n$) to class labels ($y_t \in \{-1,1\}$). The objective function is shown as follows:

$$\min_{\omega,b} \ \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{m} l(\xi_i), \ \ s.t. \ \ y_i\left[\omega^T\phi(x_i)+b\right] \geq 1-\xi_i, \ \ \xi_i \geq 0, i=1,2,\cdots m. \tag{1}$$

Where ω, b represents the normal vector and the intercept of the hyperplane. $\|\omega\|^2/2$ is a regularization term, indicating the complexity of the model. C is the penalty parameter. $\phi: X \mapsto H$ is a mapping function. $\xi_i, i=1,...,m$ are the relaxation variables. The loss function $l(\cdot)$, such as Hinge and Ramp loss [9] (equation (2) and Fig. 1, $yf(x)$ indicates the degree of similarity between the predicted result and the actual category), is crucial for evaluating the degree of difference between the real and the predicted value and for measuring the quality of prediction by the classifier.

$$l_{Hinge}(f;x,y) = \max\left(1-yf(x),0\right), \ \ l_{Ramp}(f;x,y) = \max\left(\min\left(1-yf(x),1-s\right),0\right) \tag{2}$$

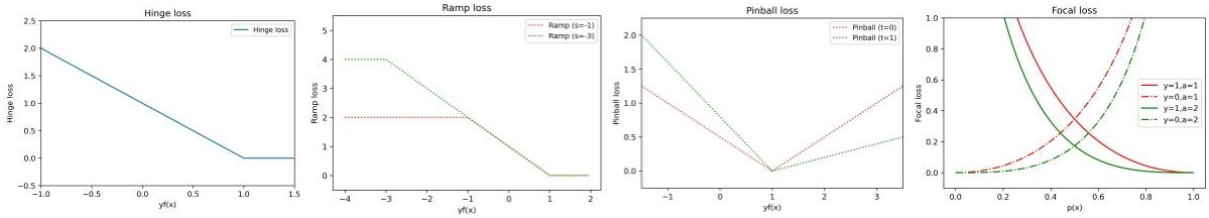

Fig. 1 Hinge, Ramp, Pinball and Focal loss (different colors represent different parameter values).

Hinge loss and Ramp loss are widely used in traditional classification algorithms. However, they lack resistance to class imbalance. In contrast, the classical Pinball loss [10] in regression problems imposes different degrees of punishment on different types of errors, which can achieve CSL by adjusting the parameter τ, as shown in equation (3) and Fig.1 ($I(\cdot)$ is an indicative function).

$$l_{Pinball}(f;x,y) = \tau^{I(yf(x)<1)}\left[-(1-\tau)\right]^{I(yf(x)\geq 1)}(1-yf(x)). \tag{3}$$

On the other hand, Focal loss [11] solves the model training problem from the perspective of sample distribution. By considering the difficulty of sample classification, the regulator parameter $a>0$ is introduced to make the loss function focus on the difficult samples, as shown in equation (4) and Fig.1 (the abscissa represents the estimated probability of predicting $y=1$, solid and dashed lines represent positive and negative samples respectively).

$$l_{Focal}(f;x,y) = -(1-f(x))^a y\log(f(x)) - f(x)^a(1-y)\log(1-f(x)). \tag{4}$$

When $a=0$, Focal loss do not consider the classification difficulty of samples. For accurately classified samples, the regulation factor ($(1-f(x))^a$ or $f(x)^a$) approaches 0 significantly, reducing the loss of easily separable samples. For samples with inaccurate classification, the regulation factor approaches 1, and the loss of difficult samples remains relatively unchanged. Therefore, the algorithm focuses more on difficult samples. In addition, difficult samples are not limited to minority class samples. Focal loss not only solves the problem of class imbalance, but also helps to improve the overall performance of the model.

2.2 Instance level Cost Sensitive Loss Function

Drawing on the above research, this paper contemplates establishing an instance-level CSL loss function. Firstly, asymmetric penalty coefficients are employed to assign different costs to different

classes, with special emphasis on minority classes. Subsequently, the classification difficulty of instances is incorporated into the modeling to further enhance the learning ability of the model.

CSL strategy first requires the selection of positive and negative classes. Generally, minority or important classes are defined as positive. Others are defined as negative. In this article, the terms "positive" and "minority" mean the same, and "negative" and "majority" are synonymous.

Next, the CSL strategy needs to define a penalty coefficient matrix $P=[p_{11},p_{12};p_{21},p_{22}]$, where p_{ij} represents the penalty coefficient when class i samples are classified into Class j. "1" represents the positive class and "2" represents the negative class. p_{12} and p_{21} represent the penalty coefficients of false negative (FN) and false positive (FP) misclassification respectively. Correct classification is not punished ($p_{11}=p_{22}=0$). For convenience and without losing generality, set $p_{21}=1$ and $\tau=p_{12}/p_{21}$. If $\tau>1$, the cost of FN misclassification is higher than that of FP. Otherwise, the cost of FP misclassification is greater.

The instance-level cost sensitive loss functions proposed in this paper are asymmetric focal Ramp (asfcRamp) and asymmetric focal Hinge (asfcHinge). The loss functions are presented in the following equations and Fig .2.

$$l_{asfcHinge}(f;x,y)=\tau^{I(FN)}\max\left((1-yf(x))^{a},0\right). \tag{5}$$

$$l_{asfcRamp}(f;x,y)=\tau^{I(FN)}\max\left(\min\left(1-s,(1-yf(x))^{a}(1-s)^{1-a}\right),0\right) \tag{6}$$

τ is the penalty coefficient (also the estimated ratio of negative class samples to positive class samples). $(1-yf(x))^{a}$ is the instance level regulator and $a>0$ is the regulator parameter.

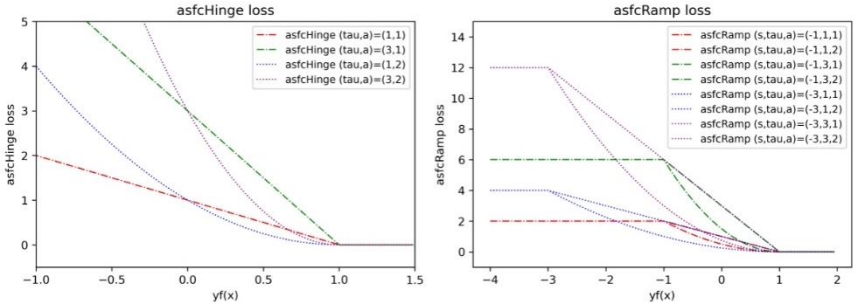

Fig .2 asfcHinge and asfcRamp loss.

2.3 Instance-level Cost-Sensitive Online Classification Algorithm

In traditional batch learning algorithms, the objective is typically to minimize the regularized empirical risk. This is done to increase the classification ability of the model while avoiding overfitting problems. It can be expressed as follows:

$$\min R_{reg-emp}[f]=\frac{\lambda}{2}\|f\|_{\mathrm{H}}^{2}+R_{emp}[f]=\frac{\lambda}{2}\|f\|_{\mathrm{H}}^{2}+C\sum_{i=1}^{m}l(f;x_i,y_i). \tag{7}$$

Where $\{(x_i,y_i)\}_{i=1}^{m}$ is the samples that have arrived. The regularization term $\frac{\lambda}{2}\|f\|_{H}^{2}$ represent the complexity of decision function $f\in\mathrm{H}$. H is the reproducing kernel Hilbert space.

Unlike batch learning, online learning algorithms employ instantaneous risk rather than empirical risk in objective functions. The online SVM is represented as the instantaneous risk minimization problem under regularization framework [12], and the objective function is defined as:

$$\min R_{reg-inst}[f]=\frac{\lambda}{2}\|f\|_{H}^{2}+R_{inst}[f]=\frac{\lambda}{2}\|f\|_{H}^{2}+l(f;x_t,y_t). \tag{8}$$

Where (x_t,y_t) is the latest arrived sample. The objective function represents the regularized instantaneous risk, comprising a loss function and a regularization term. This allows the model to be

updated with the latest data. f is updated with the commonly used OGD framework [13] as follows:

$$f^t = f^{t-1} - \eta_t z_t,\ \ \eta_t = \eta/\sqrt{t},\ \ z_t = \lambda f^{t-1} + \partial_f l(f;x_t,y_t)|_{f=f^{t-1}},$$
$$\partial_f l(f;x_t,y_t)|_{f=f^{t-1}} = l'(f;x_t,y_t)|_{f=f^{t-1}} \cdot kn(x_t,\cdot). \tag{9}$$

Where f^t is the decision function obtained after receiving the t th sample. $kn(\cdot,\cdot)$ is the kernel function, η is the learning rate. $\partial_f l(\cdot)$ is the subgradient of $l(\cdot)$ with respect to f. The update mode of the decision function is shown as follows.

Decision function update mode of asfcHinge loss:

$$f_q^t = f_q^{t-1} - \eta_t z_t = (1-\eta_t\lambda)f_q^{t-1} + I\left(yf_q^{t-1}(x)<1\right)\left(\eta_t \tau^{I(FN)} a\left[1-yf_q^{t-1}(x)\right]^{a-1} y\cdot kn\right) \tag{10}$$

Decision function update mode of asfcRamp loss:

$$f_q^t = f_q^{t-1} - \eta_t z_t = (1-\eta_t\lambda)f_q^{t-1} + I\left(s\le yf_q^{t-1}(x)<1\right)\left(\eta_t \tau^{I(FN)} a\left[1-yf_q^{t-1}(x)\right]^{a-1} y(1-s)^{1-a} kn\right) \tag{11}$$

The new algorithms have both class-level and instance-level cost sensitivity. This enables them to focus on minority class samples and difficult-to-classify samples simultaneously. In addition, the robustness and sparsity inherited from Ramp loss and Hinge loss also ensure good anti-interference performance and computational efficiency. Algorithm 1 summarizes an instance-level cost-sensitive online classification algorithm (using asfcRamp as an example).

Algorithm 1 Instance-level cost-sensitive online classification algorithm.

Input: Data stream $\{(x_t,y_t)\}$, penalty coefficient τ, regulator parameter a, truncate parameter s, learning rate η_0, regularization parameters λ.

Output: Final decision function f^m.

1: Set $t=0$. Initialize decision function f^0.

2: $t=t+1$. Receive sample (x_t,y_t) and calculate decision function $f^{t-1}(x_t)$.

3: Calculate z_t, which is the gradient of the regularized instantaneous risk.

4: Update decision function ($f^t = f^{t-1} - \eta_t z_t$).

5: If new samples arrive, the cyclic update process Step2-4 continues. Otherwise, the algorithm ends and outputs the final decision function f^m (assuming a total of m samples arrived).

3. Numerical Experiments

3.1 Experiment Settings

The numerical experiments in this section were conducted in a Python 3.10.9 environment on a Windows 10 platform with a 2.40 GHz Intel Core i5-1135G7 processor and 16 GB of RAM. Hinge, Ramp, CrossEntropy (CE), BalancedCrossEntropy (BCE) and Focal losses are employed for comparison with asfcHinge and asfcRamp. Hyperparameters include learning rate η (for all algorithms), regularization parameter λ (for all algorithms), truncation parameter s (for asfcRamp, Ramp), and penalty coefficient τ (for asfcHinge, asfcRamp), regulator parameter a (for Focal, asfcHinge, asfcRamp), balance factor b (for BCE). Optimal parameters are selected using grid search strategy: $\tau\in\{1,5,10,20,30,40,50,60,70,80,90,100\}$, $s\in\{-5,-3,-1\}$, $a\in\{1,2,3\}$, $b\in\{0.5,0.7,0.9\}$, $\lambda\in\{10^{-4},10^{-3},10^{-2}\}$, $\eta\in\{10^{-3},10^{-2},10^{-1}\}$. RBF kernel $kn(x_1,x_2)=\exp\left(-\|x_1-x_2\|^2\right)$ is used as the kernel function.

UCI benchmark datasets are used for experiments. The datasets' sample size m, number of features n, and imbalances ratio r are as: auto_mpg2 (m=392, n=7, r=3.96); Wholesale2channel (m=440, n=7, r=2.1); Wholesale2region1 (m=440, n=7, r=4.71); Wholesale2region2 (m=440, n=7, r=8.36); HCV2 (m=582, n=12, r=9.39); balance_scale (m=625, n=4, r=11.76); QSARb (m=1055, n=41, r=1.96); WQR2 (m=1599, n=11, r=6.37); DB2 (m=2549, n=16, r=3.88); letter2 (m=4639, n=16, r=4.88); WQW2_1 (m=4898, n=11, r=25.77); WQW2_45 (m=4898, n=11, r=3.62); WQW2_5 (m=4898, n=11, r=26.21); opticaldigits2 (m=5620, n=62, r=9.14); pendigit2 (m=10992, n=16, r=8.62); eegeye2 (m=14980, n=14, r=1.23). The imbalance ratio is the ratio of the number of negative and positive samples. All datasets are normalized (mean 0, standard deviation 1) to ensure scale consistency and divided into training set, validation set, and test set in a ratio of 6:2:2. Evaluation indexes used included accuracy (ACC), recall rate (RC), geometric mean (Gmean) and training time (Time). To ensure fair evaluation, a four-fold cross-validation method is adopted.

3.2 Experimental Results and Analysis

Fig .3 summarizes the winning times of algorithms. The 16 comparison experiments involved a total of 4 evaluation indicators, and finally produced 64 optimal and 64 sub-optimal performance respectively. asfcRamp and asfcHinge achieved optimal or suboptimal performance for 66 times (51.56%), of which 6 times (18.75%) are achieved on ACC, 26 times (81.25%) on Gmean, 29 times (90.63%) on RC and 5 times on Time (15.63%). New algorithms also achieve the lowest average order values on Gmean and RC, which means the best performance (Fig .4).

Index	Rank	asfcRamp	asfcHinge	CE	BCE	Focal	Ramp	Hinge	asfc proportion
ACC	1st & 2nd	3	3	1	1	7	10	7	18.75%
Gmean	1st & 2nd	13	13	0	2	1	2	1	81.25%
RC	1st & 2nd	15	14	0	3	0	0	0	90.63%
Time	1st & 2nd	0	5	1	2	3	12	9	15.63%
ALL	1st & 2nd	31	35	2	8	11	24	17	51.56%

Fig .3 Optimal and sub-optimal number on UCI datasets (Dark colors represent more wins, which means better algorithm performance.)

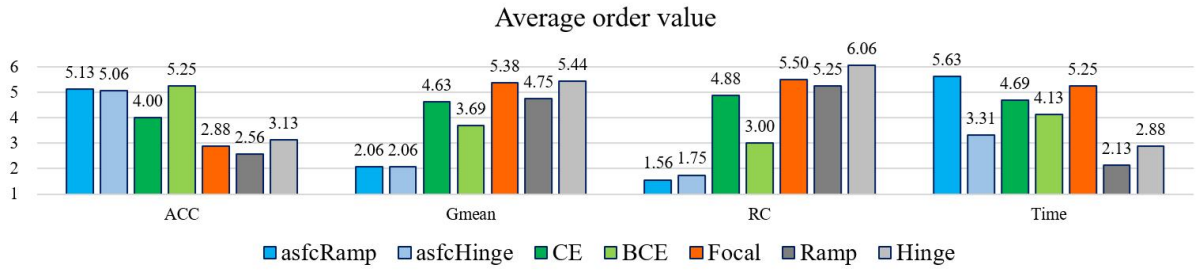

Fig .4 Average order values on UCI datasets.

Nemenyi nonparametric statistical test [14] is adopted to further detect significant performance differences between algorithms. Similar average order values usually indicate comparable performance. The critical difference (CD) is shown in Fig .5, where $CD = q_\alpha \sqrt{k(k+1)/6N}$, $k = 7$ is the number of algorithms, and $N = 16$ is the number of datasets. At the significance level $\alpha = 0.05$, $q_\alpha = 2.949$, then $CD = 2.2523$. When the difference between the average order values of algorithms exceeds CD, the Nemenyi test considers that there is a significant difference. Algorithms without significant differences are connected with black lines.

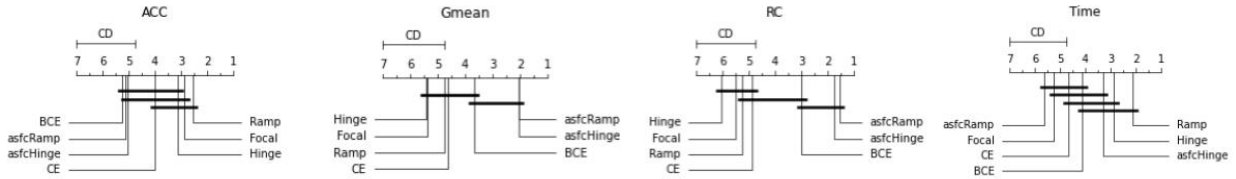

Fig .5 CD diagram in Nemenyi test results (the horizontal coordinate is the average order value).

For ACC, the difference between algorithms is not significant, but the order value of new algorithms are relatively high (not the worst), which indicates that the algorithm sacrifices some accuracy in order to achieve CSL. For Gmean and RC, there are significant differences among algorithms. The new algorithms perform best, and the gap between the new algorithms and the benchmark algorithms is obvious. This shows that CSL strategy and instance-level learning strategy significantly enhance the ability to identify all kinds of samples. For the index related to efficiency (Time), the classic Ramp and Hinge are more advantageous, and new algorithms are at medium level. This indicates that the instance-level loss functions have higher computational complexity, but the sparsity inherited from Ramp and Hinge makes the new algorithms still efficient.

4. Conclusion

In this paper, new instance-level cost-sensitive online classification algorithms are designed. By introducing an asymmetric penalty coefficient and a classification difficulty regulator, class-level and instance-level CSL are realized. Numerical experiments demonstrate that the new algorithms significantly enhance their ability to identify samples of different categories (i.e., cost sensitivity), while ensuring that accuracy and efficiency remain competitive. Future research will introduce advanced parameter optimization techniques. Moreover, by considering more loss functions and improving the algorithm structure, the algorithm performance will be further enhanced.

5. Acknowledgements

This study was supported by the National Natural Science Foundation of China (Grant No. 71901219).

6. References

[1] Zhaojie Hou, Jingjing Tang, Yan Li, Saiji Fu, Yingjie Tian. MVQS: Robust multi-view instance-level cost-sensitive learning method for imbalanced data classification. Information Sciences, 2024, 675.

[2] Priya S, Uthra RA. Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data. Complex Intelligent Systems, 2023, 9: 3499–3515.

[3] Jian L, Gao F, Ren P, et al. A noise-resilient online learning algorithm for scene classification. Remote Sensing, 2018, 10(11).

[4] Chen Z, Wang Z, Zhao M, et al. A new classification network for diagnosing alzheimer's disease in class-imbalance mri datasets. Frontiers in Neuroscience, 2022, 16.

[5] Wu Z, Lin W, Fu B, et al. A local adaptive minority selection and oversampling method for class-imbalanced fault diagnostics in industrial systems. IEEE Transactions on Reliability, 2020, 69(4): 1195–1206.

[6] Veropoulos K, Campbell C, Cristianini N. Controlling the sensitivity of support vector machines. Proceedings of International Joint Conference Artificial Intelligence, 1999.

[7] Fu S, Tian Y, Tang J, et al. Cost sensitive learning with modified stein loss function. Neurocomputing, 2023, 525: 57–75.

[8] Sch¨olkopf B, Smola AJ. Learning with Kernels: support vector machines, regularization, optimization, and beyond. MIT Press, 2002.

[9] Ozyildirim BM, Kiran M. Levenberg–marquardt multi-classification using hinge loss function. Neural Networks, 2021, 143: 564–571.

[10] Yang L, Dong H. Support vector machine with truncated pinball loss and its application in pattern recognition. Chemometrics and Intelligent Laboratory Systems, 2018, 177: 89–99.

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He and Piotr Dollár. Focal Loss for Dense Object Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 318-327.

[12] Shan X, Zhang Z, Li X, et al. Robust online support vector regression with truncated epsilon insensitive pinball loss. Mathematics, 2023, 11(3).

[13] Shalev-Shwartz, Shai. Online Learning and Online Convex Optimization. Found. Trends Mach. Learn., 2012, 4: 107-194.

[14] P. B. Nemenyi. Distribution-Free Multiple Comparisons. Ph.D. thesis. Princeton University, 1963.