2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE 2021)

A New Cost-sensitive SVM Algorithm for Imbalanced Dataset

Zheng Hengyu
School of Automation Science and Engineering of South China University of Technology
Guangzhou, China
13750056732@163.com

***Abstract*—Support Vector Machine(SVM) is a popular machine learning algorithm for its excellent generalization ability. However, similar to most of traditional algorithms, the proposal of SVM is based on an assumption that the dataset is nearly balanced, and when SVM is applied in imbalanced dataset, the result may be bias towards majority class which leads to poor performance. To solve this problem, a new cost-sensitive SVM algorithm based on samples density are proposed in this paper. In the proposed algorithm, samples' weights are depended on sample density estimated from Kernel Density Estimation(KDE) method, and furthermore, the samples' weights are modified to enlarge the weights of border samples and reduce the weights of noise samples based on Support Vector Data Description(SVDD) algorithm. The experiments result shows that the proposed algorithm could achieve satisfactory performance.**

Keywords-component; SVM;Imbalance Dataset;KDE;SVDD

I. Introduction

The classification problem of imbalance datasets is a popular topic of machine learning realm for its wide applications like credit card fraud detection[1], text classification[3], fault diagnoses[4]], and some medical application[4]. The difficult of solving imbalance problem is that in imbalance dataset, the sample size of one class is much bigger than the other and when we apply traditional classification algorithms to the dataset, the model tend to categorize the test samples to majority class. However, whether a classifier could recognize the minority class samples correctly is a key measure evaluation of classifier for it's important in some application like medical diagnosis.

There are mainly two strategies to solve imbalance problem: the data level and the algorithm level. In data level, the imbalance problem is relieved through changing the dataset. Over-sampling technique could synthetize some artificial samples of minority class to make dataset balanced, and one of the popular algorithms is SMOTE[5]. In recent years, some new oversampling algorithm based on SMOTE are proposed, including K-means SMOTE[6], WSMOTE[7] and BSMOTE[8] algorithm. On the contrary, under-sampling technique could reduce some of majority class samples to make dataset balanced, and among which Random Under-sampling is widely used. Some other under-sampling algorithms like SBC[9], Cbus[10] and ENN[11] are also practical. In algorithm level, the imbalance problem is relieved through modifying classification algorithms. Cost-sensitive learning and ensemble learning algorithm are two key schemes in algorithm level. Compared with majority samples, cost-learning algorithms pay more attentions to minority samples through increasing the error cost of minority samples, like CSSVM[12], IID3cs[13] and CS-RF[14] algorithm. Ensemble learning algorithms including Bagging[15] and Adaboost[16] solve imbalance problem through making information fusion of several classifiers, and sometimes, they would be combined with data sampling technique, like uNNBag[17], RUSBoost[18], SMOTEBoost[19].

Compared with other classifiers, SVM shows outstanding performance in imbalance problem for its separating hyperplane is depended on several support vectors. Based on SVM, a new cost-sensitive learning algorithm is proposed in this paper. In the algorithm, the cost weights of samples are depended on class density through KDE algorithm and a further weights' modifications based on SVDD is introduced to improve the classifier performance. Compared with some commonly used cost-sensitive SVM algorithm, the proposed algorithm shows better performance in generalization ability.

II. Support vector machine

SVM is one of the most popular machine learning algorithms proposed by Vapnik[20]. In SVM theory, the optimal separating hyperplane could be found through the tradeoff between the model complexity and learning ability given the limited training samples. Suppose that there is a binary classification problem, and the training dataset is denoted as $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_n, y_n)\}$, where $\boldsymbol{x}_i \epsilon \mathcal{R}^d, i = 1,2, \ldots, n$, is a d-dimensions sample, and $y_i \epsilon \{-1,1\}$ is the class label of $\boldsymbol{x}_i, i = 1,2 \ldots, n$. The mathematical model of SVM algorithm could be expressed as follow[21]:

$$\begin{aligned} &\min_{\omega,b,\xi} \ \frac{1}{2}\boldsymbol{\omega}^T\boldsymbol{\omega} + C\sum_{i=1}^{n}\xi_i \\ &s.t.\ y_i(\boldsymbol{\omega}^T\varphi(x_i) + b) \geq 1 - \xi_i, i = 1,2, \ldots, n \\ &\qquad \xi_i \geq 0, \qquad i = 1,2, \ldots, n \end{aligned} \quad (1)$$

where $\xi_i, i = 1,2, \ldots, n$ is the slack variable, $\varphi(\cdot)$ is the nonlinear function to map sample $\boldsymbol{x}_i, i = 1,2, \ldots, n$ from original feature space to a high-dimensional feature space and C is a regularization parameter used as a tradeoff between the cost of misclassification and generalization ability.

Through some mathematical transformations, the dual form of above optimization problem could be written as[22]:

$$\max_{\alpha} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

978-1-7281-8319-0/21/$31.00 ©2021 IEEE

$$s.t. \sum_{i=1}^{n} \alpha_i y_i = 0$$
$$0 \leq \alpha_i \leq C, i = 1,2,\ldots,n \quad (2)$$

where $\alpha_i, i = 1,2,\ldots,n$ is the Lagrange multiplier of instance $\boldsymbol{x}_i$, and $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is kernel function defined as $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(x_i) \cdot \varphi(x_j)$. After solving above problem, we could get the result $\alpha_i^*, i = 1,2,\ldots,n$, and if $\alpha_j^* > 0, i\epsilon\{1,2,\ldots,n\}$, the corresponding instance $\boldsymbol{x}_j$ is called support vector. Select one of support vector $\boldsymbol{x}_k(\alpha_k^* > 0)$ and we could get bias parameter b^* through Eq(3):

$$b^* = y_k - \sum_{\alpha_i^* \neq 0} y_i \alpha_i^* K(\boldsymbol{x}_i, \boldsymbol{x}_k) \quad (3)$$

And finally, given a test sample $\boldsymbol{x}_p$, the class label of $\boldsymbol{x}_p$ could be predicted as follow:

$$y_p = sgn\left(\sum_{\alpha_i^* \neq 0} y_i \alpha_i^* K(\boldsymbol{x}_i, \boldsymbol{x}_p) + b^*\right) \quad (4)$$

A. SVM for imbalance dataset

When dataset is imbalanced, the separating hyperplane of SVM would be bias towards minority class, which lead to a result that test samples are easier to be categorized to majority class. To overcome the disadvantage, CSSVM algorithm is proposed. In CSSVM algorithm, the regularization parameter C of minority samples and majority samples are different. Here, we denote the minority class as positive class, and the corresponding regularization parameter as C_+. And denote the majority class as negative class with regularization parameter C_-. The dual problem of Eq(2) could be rewritten as follow[23]:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$
$$s.t. \sum_{i=1}^{n} \alpha_i y_i = 0$$
$$0 \leq \alpha_i \leq C_+, i = 1,2,\ldots,n \ \&y_i = 1$$
$$0 \leq \alpha_i \leq C_-, i = 1,2,\ldots,n \ \&y_i = -1 \quad (5)$$

There is an empirical rule to decide the relationship of C_+ and C_-: $\frac{C_+}{C_-} = \frac{n_-}{n_+}$. Where n_- and n_+ are the numbers of majority and minority sample respectively. In CSSVM algorithm, there is no difference between the regularization parameter of every sample in the same class, which means that some noise sample may be overweight.

Another popular algorithm FSVM-CIL is proposed to reduce the influence of noise samples. Denote the training set as $S = \{(\boldsymbol{x}_1, y_1, s_1), (\boldsymbol{x}_2, y_2, s_2), \ldots, (\boldsymbol{x}_n, y_n, s_n)\}$. In the training set, $s_i, i = 1,2,\ldots,n$ are fuzzy memberships of every minority and majority samples and we define that the samples numbers of minority class and the majority class are n^+ and n^- ($n^+ + n^- = n$) respectively. A common way to decide the fuzzy membership is based on the distance between the sample and the class center[24]:

$$d_i = \begin{cases} \left\| \boldsymbol{x}_i - \frac{1}{n^+} \sum_{y_i=1} \boldsymbol{x}_i \right\| & y_i = 1 \\ \left\| \boldsymbol{x}_i - \frac{1}{n^-} \sum_{y_i=-1} \boldsymbol{x}_i \right\| & y_i = -1 \end{cases} \quad (6)$$

$$s_i = \begin{cases} 1 - \frac{d_i}{\max(d_i) + \delta} & y_i = 1 \\ 1 - \frac{d_i}{\max(d_i) + \delta} & y_i = -1 \end{cases} \quad (7)$$

the dual problem of FSVM-CIL could be written as：

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$
$$s.t. \sum_{i=1}^{n} \alpha_i y_i = 0$$
$$0 \leq \alpha_i \leq s_i C_+, i = 1,2,\ldots,n \ \&y_i = 1$$
$$0 \leq \alpha_i \leq s_i C_-, i = 1,2,\ldots,n \ \&y_i = -1 \quad (8)$$

Compared with CSSVM, FSVM-CIL algorithm could decrease the weights of noise samples, for they are always far away from class center. However, in FSVM-CIL algorithm, samples' weights are reliable to class center, and in some sample distribution which is quite different from normal distribution or in an extreme case that the samples are distributed like a circular shape, FSVM-CIL may get poor performance. What's more, some samples which are located at the border of different class are important in SVM algorithm for they are potential to be support vectors, but in FSVM-CIL, they may be assigned a small membership and be neglected.

III. PROPOSED METHOD

Designing the fuzzy membership according to class center distance is a reasonable way, but it may not work well in some situation. In our proposed algorithm, we pay more attentions to sample density and sample location. For the perspective of sample distribution, we assign larger weight of samples with high density through KDE algorithm and for the perspective of SVM algorithm, we enlarge the weight of border samples and decrease the influence of noise through support vector data description algorithm.

A. Estimate Samples' Weights through Density

Kernel Density Estimation(KDE) is a popular data analysis algorithm, and because it could directly estimate the probability density function from observational data without any reliance to priori knowledges, KDE is suitable to be used to evaluate whether a specific sample is important to the corresponding class.

If we observe one dimensional dataset $S = \{x_1, x_2, \ldots, x_n\}$ from the same but unknown data distribution, we could get the estimated probability density function $f(x)$ from Eq(9):

$$f(x) = \frac{1}{n} \sum_{t=1}^{n} K_h(x - x_i) \quad (9)$$

where $K_h(\cdot)$ is kernel function and h is the bandwidth of kernel function. A commonly used kernel function is Gaussian kernel function:

$$K_h(x - x_i) = \frac{1}{\sqrt{2\pi h^2}} e^{-\frac{(x-x_i)^2}{2h^2}} \tag{10}$$

Reference[25] have proposed a method to get the optimal value of h:

$$h = n^{-\frac{1}{5}} \cdot s \tag{11}$$

where n is the sample number of dataset and s is the standard deviation of dataset. Suppose that the dimension of observed samples is d, and we have a dataset $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}, \boldsymbol{x}_i \in R^d$. The probability density function $f(\boldsymbol{x})$ could be estimated as:

$$f_H(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{K_H}(\boldsymbol{x} - \boldsymbol{x}_i) \tag{12}$$

Similarly, $\boldsymbol{K_H}(\cdot)$ is kernel function and $\boldsymbol{H}$ is bandwidth matrix. $\boldsymbol{H}$ could be get from Eq(13):

$$\boldsymbol{H} = n^{-\frac{1}{d+4}} \cdot \boldsymbol{S} \tag{13}$$

where n is the number of samples and $\boldsymbol{S}$ is the covariance matrix of samples.

In binary classification problem, assume the dataset is $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_n, y_n)\}$, $\boldsymbol{x}_i \in R^d, y_i \in \{-1,1\}$ and the number of samples of minority class and majority class are n^+, n^-. Based on KDE algorithm, we could estimate the probability density $f(\boldsymbol{x}_i)$ of samples $\boldsymbol{x}_i$ through:

$$f(\boldsymbol{x}_i) = \begin{cases} \dfrac{1}{n^+}\displaystyle\sum_{y_i=1} \boldsymbol{K}_{\boldsymbol{H}_1}(\boldsymbol{x}_i - \boldsymbol{x}_j), & y_i = 1 \\ \dfrac{1}{n^-}\displaystyle\sum_{y_i=-1} \boldsymbol{K}_{\boldsymbol{H}_2}(\boldsymbol{x}_i - \boldsymbol{x}_j), & y_i = -1 \end{cases} \tag{14}$$

$\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ are bandwidth matrix of minority samples and majority samples respectively. Furthermore, for the simplicity of parameters tunning, a normalization procedure is needed:

$$w_i^p = \begin{cases} \dfrac{f(\boldsymbol{x}_i)}{\sum_{y_i=1} f(\boldsymbol{x}_i)}, & y_i = 1 \\ \dfrac{f(\boldsymbol{x}_i)}{\sum_{y_i=-1} f(\boldsymbol{x}_i)}, & y_i = -1 \end{cases} \tag{15}$$

B. Weight's Modifications Based on SVDD

Support Vector Data Description(SVDD)[22] is one of a variants of SVM algorithm. Different from SVM, SVDD is a one class classifier to find a minimum hypersphere which could enclose most of samples from the same class in feature space. Given a dataset $S = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$, the mathematical expression of SVDD algorithm is as follow:

$$\min_{C,R} \; R^2 + C\sum_{i=1}^{n} \xi_i$$

$$s.t. \; \|\varphi(\boldsymbol{x}_i) - \boldsymbol{c}\| \le R^2 + \xi_i$$

$$\xi_i \ge 0, i = 1,2, \dots, n \tag{16}$$

The definition of C, $\varphi(\cdot)$ are the same with SVM. The dual problem of SVDD could be written as:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_j) - \sum_{i=1}^{n}\sum_{i=1}^{n} \alpha_i\alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$s.t. \; 0 \le \alpha_i \le C$$

$$\sum_{i=1}^{n} \alpha_i = 1 \tag{17}$$

Through similar calculation with SVM, we could get $\alpha_i^*, i = 1,2, \dots, n$. The radius R is the distance between support vector to the center $\boldsymbol{c}$ which we could get from:

$$\boldsymbol{c} = \sum_{i=1}^{n} \alpha_i^* \varphi(\boldsymbol{x}_i) \tag{18}$$

In a binary imbalance classification problem, given a dataset $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_n, y_n)\}$, $\boldsymbol{x}_i \in R^d, y_i \in \{-1,1\}$, we apply SVDD algorithm to all minority and majority samples respectively, and we could get the center and radius of minority class: $\boldsymbol{c}_{min}$, R_{min} and majority class: c_{maj}, R_{maj}. According to samples' location, we divide them into four sets and they are noise samples set S_{Noise}, normal samples set S_{Normal}, border samples set S_{Border} and overlap samples set $S_{Overlap}$. We regard the samples that locate out of the hypersphere of the belonging class as noise samples, and apart from S_{Noise}, the other three sample sets are divided according to the projection distance d^p of samples. Take a minority sample $\boldsymbol{x}_k (y_k = 1)$ as example, and here we introduce the process to get the projection distance d_k^p of $\boldsymbol{x}_k$:Through SVDD algorithm, we could get center of hypersphere of minority samples $\boldsymbol{c}^+$ and majority samples $\boldsymbol{c}^-$, and we could get a vector from $\boldsymbol{c}^+$ to $\boldsymbol{c}^-$:

$$\overrightarrow{\boldsymbol{c}^+\boldsymbol{c}^-} = \boldsymbol{c}^- - \boldsymbol{c}^+ = \sum_{\alpha_i^- \neq 0} \alpha_i^- \varphi(\boldsymbol{x}_i^-) - \sum_{\alpha_i^+ \neq 0} \alpha_i^+ \varphi(\boldsymbol{x}_i^+) \tag{19}$$

The vector from $\boldsymbol{c}^+$ to $\boldsymbol{x}_k$ could be expressed as:

$$\overrightarrow{\boldsymbol{x}_k^+} = \varphi(\boldsymbol{x}_k) - \boldsymbol{c}^+ = \varphi(\boldsymbol{x}_k) - \sum_{\alpha_i^+ \neq 0} \alpha_i^+ \varphi(\boldsymbol{x}_i^+) \tag{20}$$

The projection distance of $\overrightarrow{\boldsymbol{x}_k}$ in the direction of $\overrightarrow{\boldsymbol{c}^+\boldsymbol{c}^-}$ could be calculated as:

$$d_k^p = \frac{\overrightarrow{\boldsymbol{c}^+\boldsymbol{c}^-} \cdot \overrightarrow{\boldsymbol{x}_k^+}}{\left\|\overrightarrow{\boldsymbol{c}^+\boldsymbol{c}^-}\right\|} \tag{21}$$

where, some details of above formular are as follow:

$$\overrightarrow{\boldsymbol{c}^+\boldsymbol{c}^-} \cdot \overrightarrow{\boldsymbol{x}_k^+} = \sum_{\alpha_i^- \neq 0} \alpha_i^- K(\boldsymbol{x}_k, \boldsymbol{x}_i^-) - \sum_{\alpha_i^+ \neq 0} \alpha_i^+ K(\boldsymbol{x}_k, \boldsymbol{x}_i^+)$$

$$- \sum_{\alpha_i^+ \neq 0} \sum_{\alpha_j^- \neq 0} \alpha_i^+ \alpha_j^- K(\boldsymbol{x}_i^+, \boldsymbol{x}_j^-)$$

$$+ \sum_{\alpha_i^+ \neq 0} \sum_{\alpha_j^+ \neq 0} \alpha_i^+ \alpha_j^+ K(\boldsymbol{x}_i^+, \boldsymbol{x}_j^+) \quad (22)$$

$$\|\overline{\boldsymbol{c}^+ \boldsymbol{c}^-}\|^2 = \sum_{\alpha_i^- \neq 0} \sum_{\alpha_j^- \neq 0} \alpha_i^- \alpha_j^- K(\boldsymbol{x}_i^-, \boldsymbol{x}_j^-) - 2 \sum_{\alpha_i^+ \neq 0} \sum_{\alpha_j^- \neq 0} \alpha_i^+ \alpha_j^- K(\boldsymbol{x}_i^+, \boldsymbol{x}_j^-) + \sum_{\alpha_i^+ \neq 0} \sum_{\alpha_j^+ \neq 0} \alpha_i^+ \alpha_j^+ K(\boldsymbol{x}_i^+, \boldsymbol{x}_j^+) \quad (23)$$

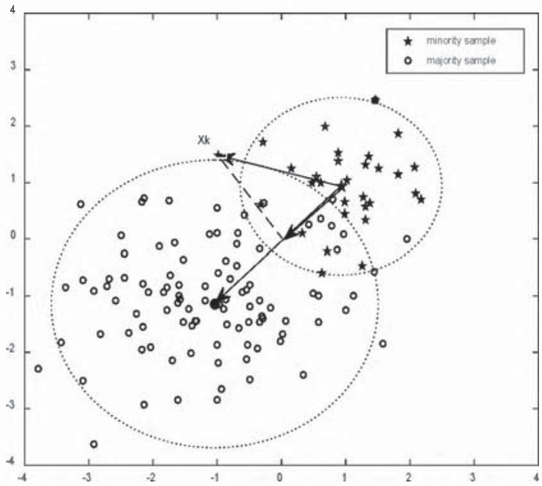

Fig. 1: The process projection distance calculation

Similarly, if $\boldsymbol{x}_k$ is a majority sample we could get the projection distance through vector $\overrightarrow{\boldsymbol{x}_k^-}$ and $\overline{\boldsymbol{c}^- \boldsymbol{c}^+}$. It's obvious that a larger d_k^p of sample $\boldsymbol{x}_k$ means that the sample locates closer to the class border and assigning a higher weight to the sample may be conducive to improving SVM performance. S_{Normal}, S_{border} and $S_{overlap}$ are divided as follow. If $d_k^p \leq 0$, it means that $\boldsymbol{x}_k$ may not be a support vector, and the sample should be belong to normal set S_{Normal}. If $0 < d_k^p < (1+\delta)(\frac{R_{min} * \|\overline{c^+ c^-}\|}{R_{min} + R_{maj}})$, it means that $\boldsymbol{x}_k$ may contribute to the hyperplane and should be belonging to border set S_{border}. The left samples which have so large d_k^p that it may cause overfitting problem are assigned to overlap set $S_{overlap}$. In above division, δ is a key parameter to decide a sample should be belonging to S_{Border} or $S_{Overlap}$, and a suitable value should be set to enlarge the border samples' weights and avoid overfitting problem, and in this paper, the value is 0.2. We apply different strategies to these four different sets and for simplicity, here we take a minority sample $\boldsymbol{x}_i (y_i = 1)$, and a majority sample $\boldsymbol{x}_j (y_j = -1)$ as example to illustrate the process to modify the sample weight of $\boldsymbol{x}_i (w_i)$ and $\boldsymbol{x}_j (w_j)$:

$$w_i = \begin{cases} w_i^p * e^{-d_i / R_{min}} & , if \ \boldsymbol{x}_i \epsilon S_{Noise} \\ w_i^p & , if \ \boldsymbol{x}_i \epsilon S_{Normal} \\ w_i^p * (1 + \eta e^{\frac{d_i^p}{R_{min}}}) & , if \ \boldsymbol{x}_i \epsilon S_{Border} \\ w_i^p (1 + \eta e^{\frac{d_i^p}{R_{min}}})^{-1} & , if \ \boldsymbol{x}_i \epsilon S_{Overlap} \end{cases} \quad (24)$$

where, η is a free parameter to control how much the border samples should be enlarged and d_i is the distance between $\boldsymbol{x}_i$ and hyperplane center $\boldsymbol{c}^+$, which could be calculated through:

$$d_i = (K(\boldsymbol{x}_i, \boldsymbol{x}_i) - 2 \sum_{\alpha_i^+ \neq 0} \alpha_i^+ K(\boldsymbol{x}_i, \boldsymbol{x}_i^+) + \sum_{\alpha_i^+ \neq 0} \sum_{\alpha_j^+ \neq 0} \alpha_i^+ \alpha_j^+ K(\boldsymbol{x}_i^+, \boldsymbol{x}_j^+))^{\frac{1}{2}} \quad (25)$$

For samples in noise set S_{Noise}, their weights should decay according to the distance to center $\boldsymbol{c}^+$; for samples in normal set S_{Normal}, no more modification is applied, because they may contribute little to the hyperplane; for samples in border set S_{Border}, which are important to the hyperplane, we enlarge their weight through multiplying a coefficient larger than 1; for samples in overlap set $S_{overlap}$, their weights need to be limited to avoid overfitting problem. Similarly, for majority sample $\boldsymbol{x}_j$, the modification process is as follow:

$$w_j = \begin{cases} w_j^p * e^{-d_j / R_{maj}} & , if \ \boldsymbol{x}_j \epsilon S_{Noise} \\ w_j^p & , if \ \boldsymbol{x}_j \epsilon S_{Normal} \\ w_j^p * (1 + \eta e^{\frac{d_j^p}{R_{maj}}}) & , if \ \boldsymbol{x}_j \epsilon S_{Border} \\ w_j^p (1 + \eta e^{\frac{d_j^p}{R_{maj}}})^{-1} & , if \ \boldsymbol{x}_j \epsilon S_{Overlap} \end{cases} \quad (26)$$

Where the distance d_j is the distance between $\boldsymbol{x}_j$ and hyperplane center $\boldsymbol{c}^-$.

Finally, in the proposed algorithm, the dual problem of SVM could be written as:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

$$s.t. \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq w_i C_+, i = 1,2, \dots, n \ \& y_i = 1$$

$$0 \leq \alpha_i \leq w_i C_-, i = 1,2, \dots, n \ \& y_i = -1 \quad (27)$$

Then we could get the decision function after solving the quadratic programming problem similar to SVM.

Here is the summary of the proposed algorithm given an imbalance dataset:

1. Estimate the sample density through KDE method, and get the samples' weights based on density.

2. Classify the dataset into four set: S_{Noise}, S_{Normal}, S_{Border}, $S_{Overlap}$ through SVDD algorithm.
3. Modify the samples' weights according to corresponding set they are belonging to.

IV. EVALUATION MEASURES

In binary imbalance classification problem, we care more about the classification accuracy of minority samples, so the classification accuracy of total test samples is not appropriate to reflect the performance of classifier. Denote the minority class as positive class and the majority class as negative class. Confuse Matrix shown in Table I is widely used to reflect the performance of classifier in imbalance problem.

TABLE I. CONFUSE MATRIX

Confusion Matrix		Predicted Class	
		Positive	*Negative*
Actual Class	*Positive*	TP	FP
	Negative	FN	TN

TP and TN are the correctly classified positive samples and negative samples respectively. FP is the number of positive misclassified and FN is the number of negative samples misclassified. From Confuse Matrix, we could get the precision(P) and recall(R) for minority samples. Then, we could get one of the popular evaluation measure $F1$ through calculation the harmonic mean of P and R[26]:

$$P = \frac{TP}{TP + FN} \tag{28}$$

$$R = \frac{TP}{TP + FP} \tag{29}$$

$$F1 = \frac{2 * P * R}{P + R} \tag{30}$$

Another widely used evaluation measure is *G-means*[27], which is the geometric mean of the accuracy of minority class and majority class:

$$G - means = \sqrt{\frac{TP}{TP + FP} * \frac{TN}{FN + TN}} \tag{31}$$

In our experiments, we mainly evaluate the performance of the classifier through $F1$ and *G-means*,

V. EXPERIMENTS

In order to compare the performance of the proposed algorithm with the widely used algorithm, we test the proposed algorithm in several UCI datasets. In these datasets, the number of minority samples is n^+ and the number of majority samples is n^-. The imbalance rate $r = \frac{n^-}{n^+}$ is used to describe the imbalance degree of a dataset. The brief information of the dataset is shown in Table II.

TABLE II. INFORMATION OF SELECTED DATASET

id	*dataset name*	*attributes*	$n_1 + n_2$	*r*
1	glass0	9	214	1.82
2	breast	9	277	2.05
3	haberman	3	306	2.78
4	abalone18_13	8	245	4.83
5	yeast05679_4	8	528	9.35
6	glass2	9	214	11.59
7	balanceB	4	625	11.65

In experiments, we compare our proposed algorithm with SVM algorithm and two popular SVM variants for imbalance dataset: CSSVM, FSVM-CIL and to verify effectiveness of our improvement, we divide the proposed algorithm into two algorithms, which have been demoted as DSVM1-CIL and DSVM2-CIL in the below table. The difference between these two algorithms is that in DSVM2-CIL algorithm, samples' weights have been modified through SVDD while samples' weights in DSVM1-CIL algorithm have not. The selected kernel function is RBF function. We would use grid search algorithm to find the satisfactory value of regularization parameter C and bandwidth r, and the searching set are $\{1,2,4,\dots,1024\}$ and $\{0.1,0.2,\dots,1.0\}$. For our proposed algorithm, a suitable value of η needs to be selected to avoid overfitting problem, and we search the optimal value of η in the searching set $\{0.05, 0.10, \dots, 0.5\}$. C_+ and C_- are set according samples number of minority and majority class All the results get from the average of two times of 5-fold cross-validation and they are shown in Table III and Table IV.

TABLE III. COMPARARISON OF F1

ID	*SVM*	*CSSVM*	*FSVMCIL*	*DSVM1CIL*	*DSVM2CIL*
1	0.734	0.740	0.724	0.743	0.752
2	0.467	0.537	0.501	0.557	0.568
3	0.327	0.506	0.504	0.523	0.537
4	0.195	0.335	0.349	0.336	0.352
5	0.466	0.490	0.507	0.477	0.489
6	0.434	0.457	0.511	0.491	0.509
7	0.689	0.621	0.588	0.644	0.663
Avg.	0.473	0.527	0.526	0.539	0.553

TABLE IV. COMPARARISON OF GMEANS

ID	*SVM*	*CSSVM*	*FSVMCIL*	*DSVM1CIL*	*DSVM2CIL*
1	0.807	0.817	0.801	0.813	0.826
2	0.602	0.668	0.634	0.685	0.693
3	0.462	0.650	0.650	0.672	0.683
4	0.270	0.610	0.613	0.604	0.625
5	0.615	0.790	0.795	0.799	0.812
6	0.634	0.857	0.859	0.882	0.894
7	0.853	0.880	0.873	0.920	0.923
Avg.	0.606	0.753	0.746	0.768	0.779

In F1 measure, we could see that DSVM2CIL algorithm get the best performance in glass0(0.752), breast(0.568), haberman(0.537), abalone18_13(0.352) dataset; SVM get the best performance in balanceB(0.689)) dataset; FSVMCIL get the best performance F1 in yeast05679_4(0.507) and glass2(0.511) dataset. In *G-means* measure, DSVM2CIL get the best performance in all seven datasets. Compared with DSVM1CIL algorithm, DSVM2CIL algorithm get higher F1 and *G-means* in the tested datasets and shows better generalization ability, which have verified that samples weights' modifications process is effective to improve SVM performance. However, from the perspective of the average performance, DSVM1CIL algorithm, which get average result of 0.539 in F1 and 0.768 in *G-means* measure still get satisfactory performance compared with CSSVM(0.527 in F1 and 0.753 in *G-means*) and

FSVM-CIL(0.526 in F1 and 0.746 in *G-means*) algorithm and it means that it's feasible to decide samples' weights through density. The simulation results show that the proposed algorithm could improve SVM performance in imbalance dataset effectively.

VI. CONCLUSION

In this paper, to improve the performance of SVM algorithm in imbalance dataset, a new cost-sensitive SVM algorithm have been proposed. In the algorithm, samples' weights are estimated according to sample density, and furthermore, a weights' modifications process based on SVDD have been used to enlarge the weight of border samples and reduce the influence of noise. Results from tested datasets show that the proposed algorithm is practical.

ACKNOWLEDGMENT

This work is supported in part by Key Laboratory of Autonomous Systems and Networked Control, College of Automation Science and Engineering, South China University of Technology, in part by Science and Technology Research Earmark of Guangzhou, Guangdong Province, China under Grant 201707010068.

REFERENCES

[1] Man Leung Wong, Kruy Seng, Pak Kan Wong. Cost-sensitive ensemble of stacked denoising autoencoders for class imbalance problems in business domain[J]. Expert Systems with Applications. 2020, 141(1), 112918

[2] Padurariu Cristian, Breaban Mihaela Elena. Dealing with Data Imbalance in Text Classification[J]. Procedia Computer Science. 2019, 159, 736-745.

[3] Gecheng Chen, Zhiqiang Ge. SVM-tree and SVM-forest algorithms for imbalanced fault classification in industrial processes[J]. IFAC Journal of Systems and Control. 2019, 8, 100052.

[4] Wei Xiushen, Mu Xin, Yang Yang. Application of secondary ensemble learning in medical data mining [J]. Computer science and exploration, 2014,8 (09): 1113-1119

[5] Chawla N V, Bowyer K W, Hall L O, et al. SMOTE: Synthetic Minority Over-Sampling Technique[J]. Journal of Artificial Intelligence Research, 2002, 16(1), pp. 321-357

[6] Georgios Douzas, Fernando Bacao, Felix Last. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE[J]. Journal of Information Sciences, 2018, 465, 1-20.

[7] Prusty M R , Jayanthi T , Velusamy K . Weighted-SMOTE: A modification to SMOTE for event classification in sodium cooled fast reactors[J]. Progress in Nuclear Energy, 2017, 100(2017), pp. 355-364.

[8] WAN K J, ADRIAN A M, CHEN K H, et al. A hybrid classifier combining borderline-SMOTE with AIRS algorithm for estimating brain metastasis from lung cancer: A case study in China Taiwan ［J］. Computer Methods & Programs in Biomedicine, 2015, 119(2) : 63-76.

[9] WAN K J, ADRIAN A M, CHEN K H, et al. A hybrid classifier combining borderline-SMOTE with AIRS algorithm for estimating brain metastasis from lung cancer: A case study in China Taiwan ［J］. Computer Methods & Programs in Biomedicine, 2015, 119(2) : 63-76.

[10] Wei-Chao Lin, Chih-Fong Tsai, Ya-Han Hu, Jing-Shang Jhang. Clustering-based undersampling in class-imbalanced data[J]. Information Sciences, 2017, 409-410, pp. 17-26.

[11] Tomek I . An Experiment with the Edited Nearest-Neighbor Rule[J]. IEEE Transactions on Systems Man & Cybernetics, 2007, SMC-6(6):448-452.

[12] Veropoulos, K., Campbell, C. and Cristianini, N. "Controlling the Sensitivity of Support Vector Machines". Proceedings of the International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 1999, pp. 55-60.

[13] Guo Bingnan, Wu Guangchao. Classification of network loans based on improved cost sensitive decision tree [J]. Computer applications, 2019,39 (10): 2888-2892

[14] Yin Hua, Hu Yuping. A cost sensitive random forest algorithm [J]. Journal of Wuhan University (Engineering Edition), 2014,47 (05): 707-711

[15] Breiman L. Bagging Predictors[J]. Machine learning, 1996, 24(2), pp. 123-140

[16] Freund Y. Boosting a weak learning algorithm by majority[J]. Information and Computation, 1995, 121(2):, pp. 256-285

[17] Błaszczyński, Jerzy, Stefanowski J . Neighbourhood sampling in bagging for imbalanced data[J]. Neurocomputing, 2015, 150, pp. 529-542

[18] Seiffert C , Khoshgoftaar T M , Hulse J V , et al. RUSBoost: A Hybrid Approach to Alleviating Class Imbalance[J]. IEEE Transactions on Systems Man and Cybernetics - Part A Systems and Humans, 2010, 40(1), pp. 185-197.

[19] Chawla, N.V., Lazarevic, A. and Hall, O. SMOTE- Boost: improving prediction of the minority class in boosting: knowledge discovery in databases. Proceeding of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases. 2003, pp. 107-119

[20] Vapnik, V.N. and Lerner, A.Y.. Recognition of patterns with help of generalized portraits. Avtomat.[J]. Telemekh, 1963, 24(6), pp.774-780.

[21] Li Hang. Statistical learning method[J]. Tsinghua University Press, 2012.

[22] Wang Fei, Feng Zao, Zhu Xuefeng. Research on pipe blockage state identification based on CSSVM with FOA parameters optimization [J]. Journal of electronic measurement and instrumentation, 2020,34 (07): 168-176

[23] Ju Zhe, Cao Junzhe, Gu Hong. Fuzzy support vector machine algorithm for imbalanced data classification [J]. Journal of Dalian University of technology, 2016,56 (05): 525-531

[24] Rivera Velázquez, Salgadougarte G, Isaías Soto, et al. Kernel density estimators[M]. Multivariate Density Estimation: Theory, Practice, and Visualization. John Wiley & Sons, Inc. 2008: 75-88.

[25] Tax D M J, Duin R P W. Support vector data description[J]. Machine learning, 2004, 54(1): 45-66.

[26] Zhou Zhihua. Machine learning [M]. Beijing: Tsinghua University Press, 2016.

[27] Xu Yuge, Lai Chunling, Luo Fei. Fault diagnosis of sewage system based on Bagging ensemble algorithm for imbalanced classification [J]. Journal of South China University of Technology (NATURAL SCIENCE EDITION), 2018,46 (08), pp. 107-11