Searchlight-scanned Over-sampling for Class Imbalance Problem

Yi Sun, Lijun Cai, Bo Liao, Wen Zhu, and JunLin Xu

◆

1 INTRODUCTION

CLass imbalance problem with complex distributions attracts more and more interests in data mining and machine learning. In one binary classification, class imbalance occurs when the number of samples in one class is obviously smaller than the other one, respectively called minority (class) and majority (class). Especially for some applications like stream data mining [2] [3], credit fraud detection [4], [5], steganography detection [6], the class imbalance problem becomes particularly prominent. Meanwhile, abnormal data distributions (like outliers, class overlapping or disjuncts [1]) make this problem more complex, called as complex distributions.

To cope with the class imbalance problem, many techniques have been proposed mainly involving algorithm-level methods [7], [8], [9] and data-level methods [10], [11], [12],or their combination [13], [15], [16]. For algorithm-level methods, the cost-sensitive learning [6], [17] assigns higher penalties to minority samples that being of misclassification. Or the ensemble learning [18], [19] makes use of Bagging and Boosting [20] for imbalanced data classification.

Different from algorithm-level methods, data-level methods generally rebalance the distribution between classes by decreasing the number of majority samples [21], [22], [23] or increasing the number of minority samples [24], [25], [26], respectively called as under-sampling methods and over-sampling methods. Obviously, under-sampling methods tend to loss some information from the original data when compared to over-sampling methods. This study focus on the over-sampling one. The most straightforward over-sampling technique is to repeat the minority for several times. But it only takes the magnitude of class size into consideration which would lead to over-fitting. Thus, many other over-sampling methods have been proposed, mainly involving interpolation-based methods [27], [28] and structure-preserving methods [29], [30], [31]. The interpolation-based technique often generates new samples between the line segment of one seed minority and one of its neighbours [27], [28]. Differently, the structure-preserving technique often generates new samples by preserving the global covariance structure of minority class.

To cope the class imbalance problem with complex distributions, some cluster-incorporated over-sampling methods have been proposed. Those methods generally group minority samples into different clusters first, and then respectively generate synthetic samples in different clusters like [32]. Beside, the technique in [33] designs one evolutionary ensemble framework for the cluster-incorporated over-sampling method, representing selected variables in a chromosome mainly involving the number of clusters, the number of synthetic samples and the selection of clustering methods.

However, except cluster-incorporated over-sampling methods, no one pure over-sampling method has been specially proposed for class imbalance problem with complex distributions. To fill the gap, this study proposes one searchlight-scanned over-sampling (SCOS) technique which innovatively regards the data filling of minority area in imbalance problem as the searchlight scanning of objective area in real life. We first define the searchlight structure in $\mathbb{R}^n$. Then, we compute this structure for each minority sample. Finally, we fill each searchlight structure with synthetic samples.

The proposed method is evaluated on several real-world imbalance datasets, and outperforms current state-of-art over-sampling techniques. Main contributions of this paper can be summarized as follows.

1) We propose one novel searchlight-scanned over-sampling method for imbalance data classification, that inspired by the scene of objective area scanning in real life.
2) We find one problem that a pair of points in the same class tends to distribute in one continuing area when their line segment does not go through the area of other classes, providing a new view on the judgement of relationship between two same-class points.
3) We use a series of searchlight structures to scan the minority area, being of natural robustness to disjuncts and good robustness to class overlapping and noises.
4) We define one searchlight structure with one vertex, one base unit vector, one restricted scalar value of inner product and one radius, which proofed to be the intersection of an Euclidean ball and a cone.

The paper is organized as follows. Section 2 reviews

- *Y. Sun, L. Cai and J. Xu are with the College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China. E-mail: id.yisun@gmail.com; ljcai@hnu.edu.cn; 18273118685@163.com.*
- *B. Liao and W. Zhu are with the Key SCOSratory of Computational Science and Application of Hainan Province, Hainan Normal University, Haikou 571158, China. E-mail: dragonbw@163.com; syzhuwen@163.com.*

(Corresponding authors: Lijun Cai and Bo Liao.)

related works, states the motivation, and defines the searchlight structure. And Section 3 introduces the proposed oversampling technique. Section 4 shows the experimental results. Discussion and conclusion are respectively included in Sections 5 and 6.

2 RELATED WORKS AND MOTIVATION

2.1 Related works

To cope with the class imbalance problem, many oversampling methods have been proposed and proven to be effective that mainly including interpolation-based methods and structure-preserving methods. SMOTE [27] is one well-known interpolation-based method which linearly generates new samples between the seed minority and one of its k nearest neighbours. To clarify the borderline between classes, B-SMOTE [36] only selects minority samples near to the borderline to generate synthetic samples. To show the different importance of borderline minority samples, ADASYN [28] assigns higher weights for minority samples that with larger number of majority in its k nearest neighbours. Obviously for interpolation-based methods, some synthetic samples tend to root into the majority area when the line segment between the seed minority and another selected minority goes through this area, bringing negative impacts for the classifier. Especial for noises, almost all their synthetic samples would root into the majority class.

Slightly different, (MWMOTE) [32] assigns higher weights for minority samples with nearer Euclidean distance to the majority class. Additionally to fit complex distributions, MWMOTE does not select k nearest neighbours but same-cluster samples to linearly generate new samples. But the clustering algorithm highly depends on the distance factor and only takes minority samples into consideration that ignoring the existence of majority samples. Thus, two minority samples, that divided apart by majority samples, maybe wrongly grouped together when their distance is near enough. Or two minority samples, that sparsely distributed, maybe wrongly grouped apart when their distance is far enough.

For structure-preserving methods, INOS [31] generates synthetic samples to maintain the main covariance structure of minority class. Moreover, INOS uses a small percentage of synthetic samples from ADASYN to protect key original minority samples. Similarly, MDO [30] first identifies suitable minority samples that with enough minority in their nearest neighbours; and then for those suitable minority, MDO generates synthetic samples with the same Mahalanobis distances from the majority. Obviously, the covariance of minority class or the Mahalanobis distance does not care complex distributions (like outliers, class overlapping or disjuncts [1]) at all, so leading to the extra data cleaning in INOS [31] or several noises and overlapped samples in MDO [30].

2.2 Motivation

In real life, some domains, like armed defence, prison or detention center, always use multi searchlights to jointly scan objective areas at night, similar to the over-sampling process in class imbalance problem. First from the perspective of purpose, the purpose of searchlight-used domain is to scan objective areas while the over-sampling procedure is to fill minority area with synthetic samples. Second from the characteristic of structure, the minority and majority area respectively corresponding to the objective area and the barrier (like walls or buildings).

To further display their similarities, as seen in Fig. 1. (a), minority samples distribute over two areas that can be treated as two objective areas to be scanned, thus two searchlights needed. As seen in Fig. 1. (d), we put the searchlight on the center of one minority area and rotate it for a circle to scan the whole minority area. But three questions asked: (1) how much searchlights are needed? (2) how to scan the minority area with irregular distribution? for example, not a circle. (3) the scan process of rotation in $\mathbb{R}^2$ fails in $\mathbb{R}^3$ or higher dimensions. so how to adapt to any dimensions?

Intuitively in real life, the light cone is first started from one launch point, then passes through the objective area, last stopped by the surface of barrier. Thus to answer above questions, for each minority sample, we compute one searchlight structure first started from the inner minority area, then passes through this seed minority and its nearby area, last stopped by the majority class. Thus for the first question, we do not need the number of searchlights, but compute one searchlight structure for each minority sample. For the second question, the searchlight structure only scans the local nearby area of each minority, that not suffering from the integral irregular distribution of minority class. For the third question, we will give a definition of searchlight structure in $\mathbb{R}^n$ in the next sub-section. The theory of proposed method will be first demonstrated with emulational datasets that involving different complex distributions and evaluated with real-world datasets in the experimental section.

2.3 searchlight structure

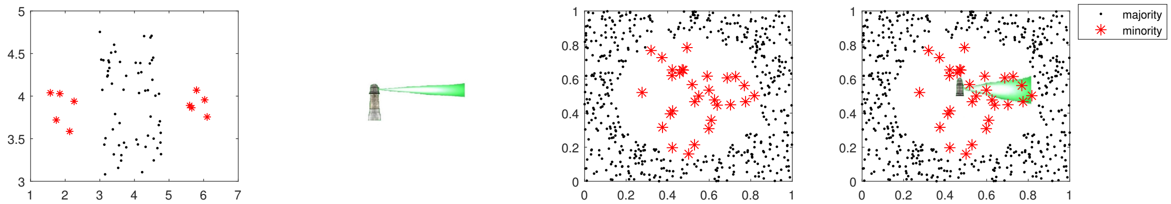
This subsection gives a new mathematical structure, called the searchlight structure. First, we analyse its possible components, then define its mathematical representation , last discuss its geometrical characteristics.

Intuitively, in real life, the light cone consists of one launch point, one scan direction, one cone angle, and one distance of propagation. Except the cone angle, other three components can be easily defined in $\mathbb{R}^n$. To define the cone angle, as seen in Fig. 2, we use the inner product of one unit borderline vector and one unit base vector to replace the cone angel, where the unit base vector denotes the scan direction. Thus in $\mathbb{R}^n$, searchlight structure consists of four corresponding components as one vertex, one base unit vector, one restricted scalar value of inner product, and one radius.

To this end, we suppose the vertex being at the origin for convenient discussion, and define the searchlight structure as:

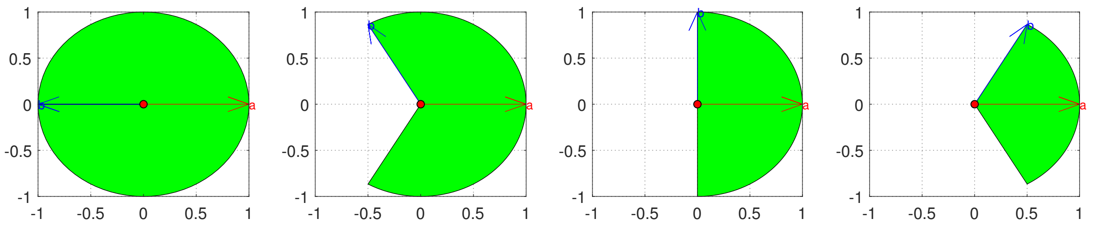
$$\mathbb{S} = \left\{ \boldsymbol{x} \middle| \left\langle \frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}, \boldsymbol{a} \right\rangle \geq \rho \ \& \ \|\boldsymbol{x}\|_2 \leq r \right\} \bigcup \left\{ \boldsymbol{0} \right\} \tag{1}$$

where $\left\langle , \right\rangle$ denotes the inner product operation of two vectors such as $\left\langle \boldsymbol{c}, \boldsymbol{d} \right\rangle = \boldsymbol{c} \cdot \boldsymbol{d}$, $\boldsymbol{a}$ is the base unit vector

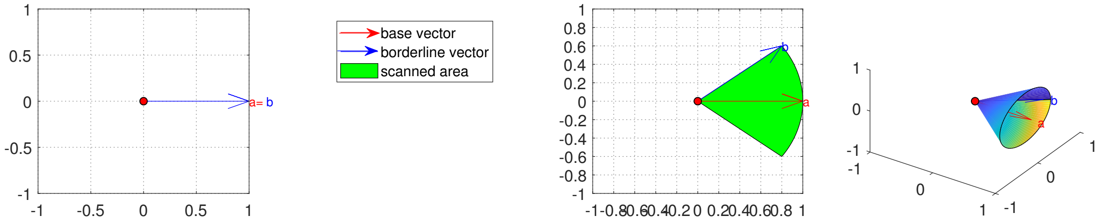


(a) two minority areas (b) searchlight scanning in real world (c) one imbalanced data (d) scan the minority area

Fig. 1: Motivation of proposed method. (a) two minority areas, (b) searchlight scanning in real world, (c) one imbalanced data, (d) scan the minority area. Intuitively, the whole minority area in (c) can be scanned when putting the searchlight on the center location and rotating it for a circle; two minority areas in (a) maybe need two searchlights. The motivation in this paper is that the method of minority over-sampling is similar to the scene of searchlight scanning in real world.



(a) p=a • b=-1 (b) p=a • b=-0.5 (c) p=a • b=0 (d) p=a • b=0.5



(e) p=a • b=1 (f) p=0.8 in 2-D data space (g) p=0.8 in 3-D data space

Fig. 2: Unit searchlight structures. (a)-(e) with different inner products in $\mathbb{R}^2$, (f)-(g) unit searchlight structures in $\mathbb{R}^2$ and $\mathbb{R}^3$ with $\boldsymbol{a}\boldsymbol{\cdot}\boldsymbol{b}=0.8$, $\|\boldsymbol{a}\|_2=1$ and $\|\boldsymbol{b}\|_2=1$

that $\|\boldsymbol{a}\|_2=1$, and $\rho\in[0,1]$ is the restricted scalar value of inner product, r is the radius. Obviously, this structure is the intersection of an Euclidean ball $\mathbb{B}$ and one unknown structure $\mathbb{C}$:

$$\mathbb{C}=\left\{\boldsymbol{x}\middle|\left\langle\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2},\boldsymbol{a}\right\rangle\geq\rho\right\}\bigcup\left\{\mathbf{0}\right\} \tag{2}$$

$$\mathbb{B}=\left\{\boldsymbol{x}\middle|\ \|\boldsymbol{x}\|_2\leq r\right\} \tag{3}$$

$$\mathbb{S}=\mathbb{C}\bigcap\mathbb{B} \tag{4}$$

To this end, we discuss the geometry characteristics of $\mathbb{C}$; in brief, whether $\mathbb{C}$ is a cone or a convex cone in $\mathbb{R}^n$.

Definition 2.1 (page 25 in [35]). A set $\mathbb{C}$ is a cone, only for every $\boldsymbol{x}\in\mathbb{C}$ and $\theta\geq0$, have $\theta\boldsymbol{x}\in\mathbb{C}$.

Theorem 2.1. Given a set $\mathbb{C}=\left\{\boldsymbol{x}\middle|\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}\boldsymbol{\cdot}\boldsymbol{a}\geq\rho\right\}\bigcup\left\{\mathbf{0}\right\}$ where $\rho\in[-1,1]$ is one scalar and $\boldsymbol{a}$ is one referred unit vector, the set $\mathbb{C}$ is a cone.

Proof: Since $\boldsymbol{x}\in\mathbb{C}$, we have $\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}\boldsymbol{\cdot}\boldsymbol{a}\geq\rho$. We consider the following two cases:

Case 1: $\theta=0$: Have $\theta\boldsymbol{x}=\mathbf{0}\in\mathbb{C}$.

Case 2: $\theta>0$, compute

$$\frac{\theta\boldsymbol{x}}{\|\theta\boldsymbol{x}\|_2}\boldsymbol{\cdot}\boldsymbol{a}=\frac{\theta\boldsymbol{x}}{\theta\|\boldsymbol{x}\|_2}\boldsymbol{\cdot}\boldsymbol{a}=\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}\boldsymbol{\cdot}\boldsymbol{a}\geq\rho \tag{5}$$

have $\theta\boldsymbol{x}\in\mathbb{C}$.

Hence, $\theta\boldsymbol{x}\in\mathbb{C}$ in both cases, $\mathbb{C}$ is a cone $\rho\in[-1,1]$. ☐

Definition 2.2 (page 25 in [35]). A set $\mathbb{C}$ is a convex cone if it is convex and a cone, which means that for any $\boldsymbol{x_1},\boldsymbol{x_2}\in\mathbb{C}$ and $\theta_1,\theta_2\geq0$, we have

$$\theta_1\boldsymbol{x_1}+\theta_2\boldsymbol{x_2}\in\mathbb{C} \tag{6}$$

Theorem 2.2. Given a set $\mathbb{C}=\left\{\boldsymbol{x}\middle|\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}\boldsymbol{\cdot}\boldsymbol{a}\geq\rho\right\}\bigcup\left\{\mathbf{0}\right\}$ where $\rho\in[0,1]$ is one scalar and $\boldsymbol{a}$ is one referred unit vector, the set $\mathbb{C}$ is a convex cone.

Proof: We first make $f(\boldsymbol{x})=\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_2}\boldsymbol{\cdot}\boldsymbol{a}$. Since $\boldsymbol{x_1},\boldsymbol{x_2}\in\mathbb{C}$, we have $f(\boldsymbol{x_1})=\frac{\boldsymbol{x_1}}{\|\boldsymbol{x_1}\|_2}\boldsymbol{\cdot}\boldsymbol{a}\geq\rho$ and $f(\boldsymbol{x_2})=\frac{\boldsymbol{x_2}}{\|\boldsymbol{x_2}\|_2}\boldsymbol{\cdot}\boldsymbol{a}\geq\rho$. We consider the following two cases:

Case 1: $\theta_1 \boldsymbol{x_1} + \theta_2 \boldsymbol{x_2} = \mathbf{0} \in \mathbb{C}$.

Case 2: $\theta_1 \boldsymbol{x_1} + \theta_2 \boldsymbol{x_2} \neq \mathbf{0}$, compute

$$
\begin{aligned}
f(\theta_1 \boldsymbol{x_1} + \theta_2 \boldsymbol{x_2}) &= \frac{\theta_1 \boldsymbol{x_1} + \theta_2 \boldsymbol{x_2}}{\left\|\theta_1 \boldsymbol{x_1} + \theta_2 \boldsymbol{x_2}\right\|_2} \cdot \boldsymbol{a} \\
&= \frac{\theta_1 \left\|\boldsymbol{x_1}\right\|_2 \dfrac{\boldsymbol{x_1}}{\left\|\boldsymbol{x_1}\right\|_2} + \theta_2 \left\|\boldsymbol{x_2}\right\|_2 \dfrac{\boldsymbol{x_2}}{\left\|\boldsymbol{x_2}\right\|_2}}{\left\|\theta_1 \boldsymbol{x_1} + \theta_2 \boldsymbol{x_2}\right\|_2} \cdot \boldsymbol{a} \\
&= \frac{\theta_1 \left\|\boldsymbol{x_1}\right\|_2 \dfrac{\boldsymbol{x_1}}{\left\|\boldsymbol{x_1}\right\|_2} \cdot \boldsymbol{a} + \theta_2 \left\|\boldsymbol{x_2}\right\|_2 \dfrac{\boldsymbol{x_2}}{\left\|\boldsymbol{x_2}\right\|_2} \cdot \boldsymbol{a}}{\left\|\theta_1 \boldsymbol{x_1} + \theta_2 \boldsymbol{x_2}\right\|_2} \\
&= \frac{\theta_1 \left\|\boldsymbol{x_1}\right\|_2 f(\boldsymbol{x_1}) + \theta_2 \left\|\boldsymbol{x_2}\right\|_2 f(\boldsymbol{x_2})}{\left\|\theta_1 \boldsymbol{x_1} + \theta_2 \boldsymbol{x_2}\right\|_2} \\
&\geq \frac{\theta_1 \left\|\boldsymbol{x_1}\right\|_2 + \theta_2 \left\|\boldsymbol{x_2}\right\|_2}{\left\|\theta_1 \boldsymbol{x_1} + \theta_2 \boldsymbol{x_2}\right\|_2} \times \rho \\
&\geq \rho
\end{aligned}
\tag{7}
$$

have $\theta_1 \boldsymbol{x_1} + \theta_2 \boldsymbol{x_2} \in \mathbb{C}$. Noting: the last step of proof is draw from the Triangle Inequality for norm ($\left\|\boldsymbol{x} + \boldsymbol{y}\right\|_2 \leq \left\|\boldsymbol{x}\right\|_2 + \left\|\boldsymbol{y}\right\|_2$).

Hence, $\theta_1 \boldsymbol{x_1} + \theta_2 \boldsymbol{x_2} \in \mathbb{C}$ in both cases; $\mathbb{C}$ is a convex cone when $\rho \in [0, 1]$. □

In generally, if $\rho \in [-1, 1]$, the searchlight structure is the intersection of an Euclidean ball and a cone; if $\rho \in [0, 1]$, the searchlight structure is the intersection of an Euclidean ball and a convex cone.

3 SEARCHLIGHT-SCANNED OVER-SAMPLING FOR IMBALANCE PROBLEM

In this section, we propose a novel method, called searchlight-scanned over-sampling (SCOS), for the class imbalance problem with complex distributions. The overall SCOS algorithm is described in Algorithm S1. First, SCOS computes relationships between any two minority samples. Then, based on those relationships, SCOS computes one searchlight structure for each minority sample. Finally, SCOS generates new synthetic samples in those searchlight structures.

3.1 Direct-interlinked relationship between same-class points

In the first step, SCOS first describes direct-interlinked relationship between two same-class points, then computes it. As seen in Fig. 5. (a), two minority points are considered of direct-interlinked relationship when their line segment does not go through the majority area. For example, the line segment between point A and E does not go through the majority area, so denoting A and E are direct-interlinked or A is direct-interlinked to E. This relationship is symmetrical that E is also direct-interlinked to A. Whereas judging whether the line segment of two points goes through the majority area is difficult. Because the majority area is unknown that only majority points are known. Thus to compute the relationship between two points, as seen in Fig. 5. (b), SCOS selects one majority point from all majority to makes a minimum value of inner product of two unit vectors that directed from this majority point to two minority points. Finally, SCOS sets one threshold for it. Two points are direct-interlinked when their minimum value is larger than the threshold. For example, as seen in Fig. 5. (c), the seed minority point B owns four direct-interlinked minority points whose minimum inner-product values are larger than the threshold -0.8910 ($cos(153^o) = -0.8910$). It is easily understood since the minimum inner-product value of two non direct-interlinked points will be equal to -1 when the majority area is fully filled with majority points; or near to -1 when the majority area is filled with enough majority points.

In detail, for any two minority samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, we compute the inner product of two unit vectors:

$$
D_k(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_k) = \left\langle \frac{\boldsymbol{x}_i - \boldsymbol{z}_k}{\left\|\boldsymbol{x}_i - \boldsymbol{z}_k\right\|_2}, \frac{\boldsymbol{x}_j - \boldsymbol{z}_k}{\left\|\boldsymbol{x}_j - \boldsymbol{z}_k\right\|_2} \right\rangle \tag{8}
$$

where $\boldsymbol{z}_k$ is the k-th sample in the majority set. When $\left\|\boldsymbol{x}_i - \boldsymbol{z}_k\right\|_2 = 0$ or $\left\|\boldsymbol{x}_j - \boldsymbol{z}_k\right\|_2 = 0$, set $D(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_k) = 0$. Then pick the minimum inner product:

$$
M(\boldsymbol{x}_i, \boldsymbol{x}_j) = \min_{1 \leq k \leq m} D_k(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_k) \tag{9}
$$

where m is number of majority samples. Finally, we obtain the relationship between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$:

$$
I(\boldsymbol{x}_i, \boldsymbol{x}_j) = \begin{cases} 1, & if\ M(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq \delta \\ 0, & else \end{cases} \tag{10}
$$

where δ is one threshold ranging in [-1, 1]. $I(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1$ denotes two minority direct-interlinked, $I(\boldsymbol{x}_i, \boldsymbol{x}_j) = 0$ denotes not direct-interlinked.

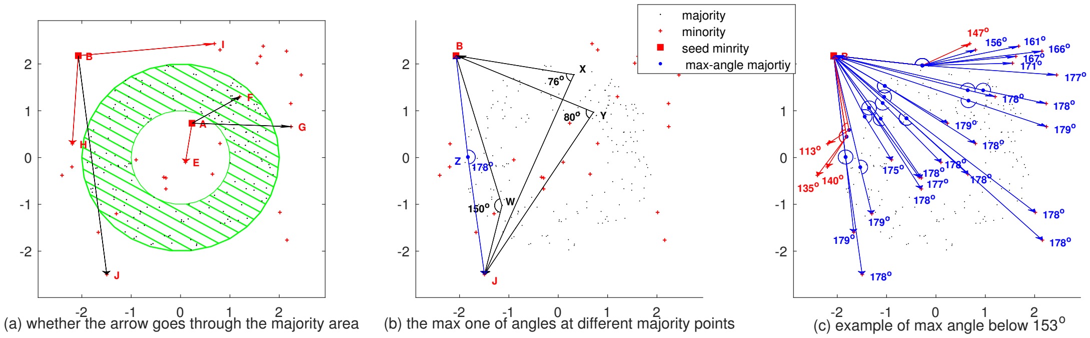
In the class imbalance problem, the direct-interlinked relationship is important. Take interpolation-based methods [27], [28], [32] for example, most of them generate one synthetic point between the line segment of one seed minority and other one selected minority. If these two minority are direct-interlinked, the new synthetic point does not fall in the majority area at all; otherwise, would fall in the majority area, so causes noises or class overlapping.

3.2 searchlight structure

Based on the direct-interlinked relationship, in the second step, SCOS computes the base unit vector, the vertex and the radius, and sets the restricted scalar value of inner product as one parameter of threshold for the searchlight structure defined in Eq. 1. Thus, the essence of searchlight structure involves into three aspects.

3.2.1 Base unit vector

In the first aspect, SCOS first explains why the vertical scan help the searchlight structure to cover more boundary area of minority class, then attends to compute one base unit vector that nearly vertical to the borderline. As seen in Fig. 4, for the convenient explanation, the borderline between classes is linear; x=0.25 is used to divide the minority area into the boundary one and non-boundary one; in $\mathbb{R}^2$, the searchlight structure is one sector. In brief, we attempt to display which scan direction makes the area of sector that falling in the boundary minority area maximum. As seen in Fig. 4. (a), the scanned boundary area is of the max size in the first graph that denoting the vertical scan. As seen in Fig. 4. (b), the scanned boundary area is of the max size in the



(a) whether the arrow goes through the majority area (b) the max one of angles at different majority points (c) example of max angle below 153°

Fig. 3: Relationship description of same-class points. (a) whether the arrow goes through the majority area, (b) the max one of angles at different majority points, (c) example of max angle below 153^o. In (a), the green slash denotes the majority area, the red arrow denotes the line between a pair of same-class points does not go through the majority area, the black arrow denotes the line between a pair of same-class points goes through the majority area. In (b), the blue denotes the max one of angles at different majority points. In (c), the red denotes the max angle of two minority points is smaller than the threshold 153^o, the blue denotes the max angle is larger than the threshold 153^o.

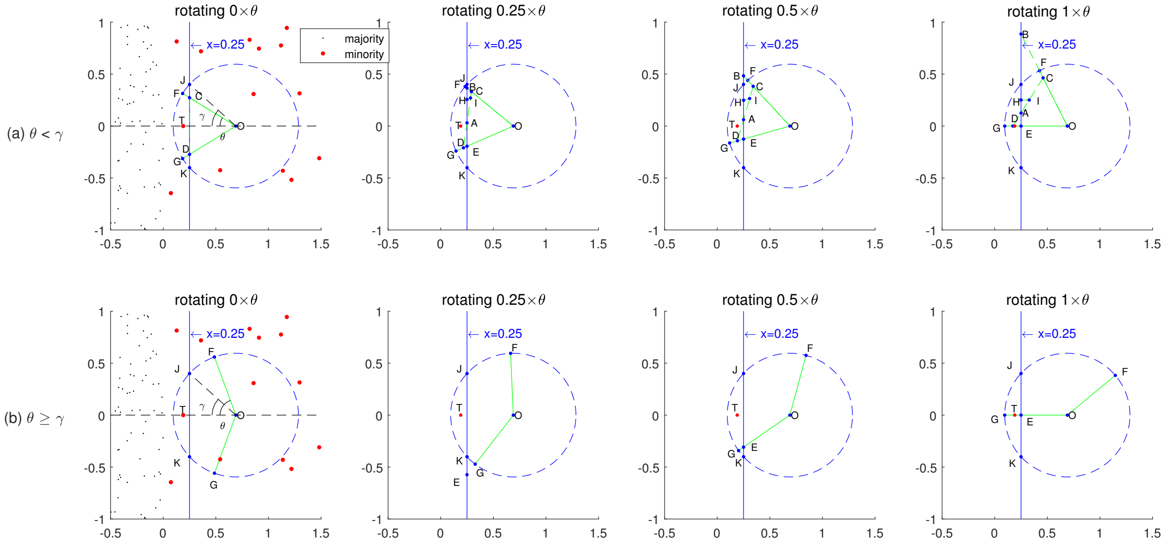


Fig. 4: Why vertical scan products the maximum scanned boundary area. Where point T is the seed minority point to be scanned; x=0.5 is a line that divides the minority area into the boundary one and non-boundary one; S_{OGF} is one fixed scanned sector (four components of search light structure: vertex O, angle $2 \times \theta$ corresponding to restricted scalar value of inner product, radius OG, rotated scan in each column corresponding to the base unit vector); the area in S_{GEJ} denotes the part of scanned sector that falling in the boundary area; in some special cases, it is S_{GDCF} in first graph in (a), S_{GECF} in second graph in (a), or S_{KJK} in first two graphs in (b). In both (a) and (b), individually rotate the sector, it individually leaves the boundary area. (a) $\theta < \gamma$; obviously S_{GDCF} in first graph owns the max falling area in the boundary; to see this, draw a line HI with HA=EA and IA=DA, S_{ADE} is the new joining area, S_{ACB} in second graph or S_{ACFJ} in followed graphs is the leaving area, $S_{ADE} = S_{AIH}$ and $S_{AIH} \in S_{ACB}$ or S_{ACFJ}, so the leaving area is larger than the new joining area in that case (notice: whether the net-loss area gradually increases is not confirmed owing to our limited knowledge), (b) $\theta \geq \gamma$; obviously S_{KJK} in first two graphs owns the max falling area in the boundary. Intuitively, the vertical scanning to the borderline area products the maximum scanned boundary area.

first two graphs that respectively denoting the vertical scan and nearly vertical scan. To this end, SCOS considers the vertical scan help the searchlight structure covering more boundary area of minority. Because the vertex of searchlight structure is unknown, one inverse of base unit vector is priorly estimated that vertically pointed to minority area from the majority area. As seen in Fig. 5. (a) for the seed minority A, SCOS draws an arrow from the center of its k nearest majority neighbours B to A as $\boldsymbol{c}$, denoting the nearly vertical direction to the borderline. And uses the inverse and unit of $\boldsymbol{c}$ as the base unit vector.

In detail, for the seed minority sample $\boldsymbol{x}$, compute its k nearest majority neighbours :

$$S_{knn} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_k\} \tag{11}$$

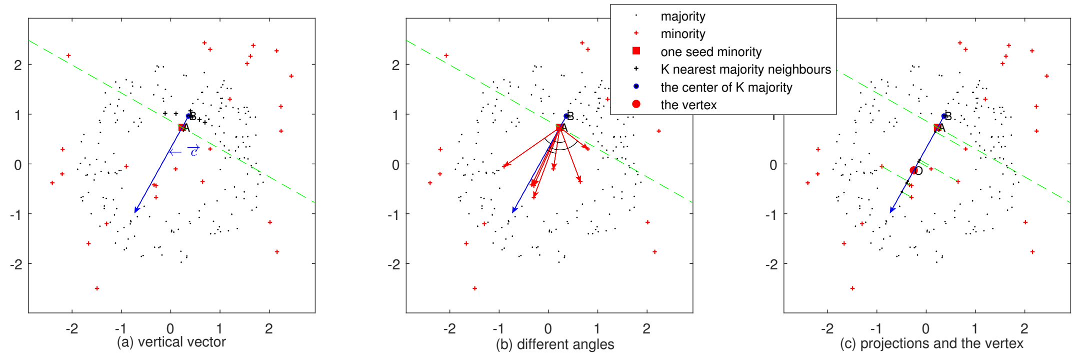


Fig. 5: Vertex computation. (a) vertical vector $\boldsymbol{c}$, (b) different angles of direct-interlinked minority, (c) projections and the vertex. Where the blue arrow denotes the vertical vector, green slash denotes one line that vertical to $\boldsymbol{c}$; red arrow denotes the vector that started from the seed minority point and pointed to the direct-interlinked minority. For one seed minority point A, first compute the center point B of its K majority neighbours, then draw an arrow from B to A as the vertical vector $\boldsymbol{c}$, next compute the projection on $\boldsymbol{c}$ that only positive projections are remained, last use the mean of those positive projections as the vertex of searchlight structure as O.

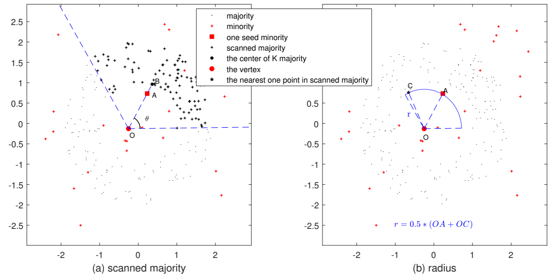


Fig. 6: Radius computation. (a) scanned majority, (b) radius. Take the seed point A for example, first compute scanned majority samples under one vertex O, one vertical vector $\boldsymbol{a}$ and one scanned angle $2\times\theta$; then select the nearest one point C from scanned majority sample for reference, last compute the radius; in this case, radius is the mean of OC and OA.

where $\boldsymbol{z}_k$ is the k-th nearest majority sample to the seed one $\boldsymbol{x}$. Then compute its center:

$$\bar{\boldsymbol{z}} = \frac{1}{k}\sum_{i=1}^{k}\boldsymbol{z}_i \tag{12}$$

Compute the nearly vertical vector, and normalize it:

$$\boldsymbol{c} = \frac{\boldsymbol{x}-\bar{\boldsymbol{z}}}{\|\boldsymbol{x}-\bar{\boldsymbol{z}}\|_2} \tag{13}$$

Obtains the base unit vector:

$$\boldsymbol{a} = -c \tag{14}$$

3.2.2 Vertex

In the second aspect, SCOS finds the vertex on the vertical vector that with one deeper location than the seed minority sample, making the searchlight structure started from the inner minority area first and then pass through the seed one. As seen in Fig. 5. (b) and (c), SCOS firstly computes the projects of between $\boldsymbol{a}$ and vectors that from the seed one to other direct-interlinked minority points; then only remains the positive projects and uses their mean as the location of vertex.

In detail, for the seed minority sample $\boldsymbol{x}$, SCOS records its direct-interlinked minority samples as $\{\boldsymbol{x}_1, \boldsymbol{x}_2, .., \boldsymbol{x}_m\}$; where $I(\boldsymbol{x}, \boldsymbol{x}_m) = 1$, m is the number of direct-interlinked minority samples. Then obtain the vector that pointed to the direct-interlinked minority from the seed minority:

$$\boldsymbol{d}_i = \boldsymbol{x}_i - \boldsymbol{x} \tag{15}$$

where $\boldsymbol{x}_i$ is i-th sample in $\{\boldsymbol{x}_1, \boldsymbol{x}_2, .., \boldsymbol{x}_m\}$. And compute its project on $\boldsymbol{c}$

$$p_i = \left\langle \boldsymbol{d}_i, \boldsymbol{c} \right\rangle \tag{16}$$

$$J(p_i) = \begin{cases} 1, & if\ p_i > 0 \\ 0, & else \end{cases} \tag{17}$$

Next, compute the mean of positive projections:

$$\bar{p} = \frac{1}{\sum_{i=1}^{m} J(p_i)} \sum_{i=1}^{m} J(p_i) \times p_i \tag{18}$$

Finally, compute the vertex of searchlight structure:

$$v = \bar{p} \times \boldsymbol{c} + \boldsymbol{x} \tag{19}$$

3.2.3 Radius

In the third aspect, SCOS computes the radius for the searchlight structure. To make the searchlight structure not cover the majority area, as seen in Fig. 6.(a), SCOS firstly computes scanned majority samples; then for the vertex, computes its nearest scanned majority point as seen in Fig. 6.(b). Last compute the radius that referring lengths of the nearest scanned majority and the seed minority to the vertex. For example, $r = (OC + OA)/2$ with the case $OC \geq OA$.

In detail, SCOS sets the restricted scalar value of inner product as τ ranging in[-1,1]. Then, find scanned majority samples that satisfying:

$$\left\langle \frac{\boldsymbol{z}_k - \boldsymbol{v}}{\|\boldsymbol{z}_k - \boldsymbol{v}\|_2}, \boldsymbol{c} \right\rangle \geq \tau \tag{20}$$

where $\boldsymbol{z}_k$ is the k-th sample in the majority set, $\boldsymbol{v}$ is the vertex of searchlight structure. Next, find the nearest majority point from those scanned majority samples, denoting it as $\boldsymbol{g}$; last compute the radius:

$$r = min(L(\boldsymbol{v},\boldsymbol{x}), L(\boldsymbol{v},\boldsymbol{g})) + 0.5(L(\boldsymbol{v},\boldsymbol{g}) - L(\boldsymbol{v},\boldsymbol{x})) \quad (21)$$

where $min(,)$ means the minimum one from two values, $L(,)$ is the Euclidean distance that $L(\boldsymbol{v}_1,\boldsymbol{v}_2) = \|\boldsymbol{v}_1 - \boldsymbol{v}_2\|_2$. Obviously for non-overlapped seed minority, the seed one is nearer to the vertex that r is the mean of two lengths; for overlapped seed minority, the nearest majority point is nearer to the vertex that r is the length of the nearest majority point to minus the half gap of two length. Thus, making the searchlight structure cover nearly half local boundary area in normal cases, and not cover the majority area in overlapping cases.

3.3 Data generation of minority class

Since the searchlight structure has been computed, SCOS generates the new synthetic sample:

$$new = \boldsymbol{v} + (\xi \times r) \times \overrightarrow{d} \quad (22)$$

where $\overrightarrow{d}$ is one rand unit vector that satisfying $\left\langle \overrightarrow{d}, \boldsymbol{a} \right\rangle \geq \tau$ (Notice: in the high dimension, randomly generating the condition-satisfied unit vector is difficult. To solve this problem, on track is to generate one random unit vector first, then add it to $\boldsymbol{a}$, next normalize to one unit vector again; this procedure can be repeated for several times until one condition-satisfied unit vector d generated. In experience, SCOS uniformly generates a series of condition-satisfied unit vectors first, then randomly pick up one for use).

And ξ is one rand scalar in [0,1], to make more synthetic samples near to the boundary, SCOS randomly generates it as:

$$\xi = \begin{cases} rand(), & if\ rand() > 0.5 \\ 1, & else \end{cases} \quad (23)$$

where rand() is one rand number in [0,1], here, two rand values are included; one is for the half probability setting $\xi = 1$, another one is for setting to $\xi = rand()$.

4 EXPERIMENTAL RESULTS

This section compares the performance of the proposed one with other state-of-art over-sampling methods including SMOTE [27], B-SMOTE1 [36], B-SMOTE2 [36], ADASYN [28], MWMOTE [32], INOS [31], AMDO [37] and GDO [26]. First for intuitive comparison, we visualize over-sampling methods on 2D emulational datasets. Next for the further comparison, we test over-sampling methods on real-world benchmark datasets that collected from UCI machine learning repository [38] and [39]. Finally, we discuss their classification performance.

4.1 Visual comparison in $\mathbb{R}^2$

This section plots synthetic data of over-sampling methods on 2D emulational datasets for visual comparison, respective named Ring, Linear, TwoBall in Fig. 7 and 8, S1 and S2 (in the supplementary material). Black points denote majority samples, red points denote minority samples and blue crosses denote synthetic minority samples. Among those datasets, Ring is added with several noises, Linear is added with several overlapped points. For equal comparison, $N_{maj} - N_{min}$ synthetic data are generated for each over-sampling method, where $N_{maj}$is number of majority class and N_{min} is the numbers of minority class.

As seen in Fig. 7, MWMOTE and SCOS are robuster to noises than other methods. Because MWMOTE only over-sampling in clusters, but fails to two noises when they distribute too close. As seen in Fig. 8, AMDO and SCOS is robuster to overlapped points than other methods. Because AMDO only selects minority samples that with enough minority in their nearest neighbours but fail to generates much synthetic samples in the boundary area.

4.2 Comparison on real-world datasets

For imbalance data classification, accuracy is not one appropriate performance measurement because of its bias towards the majority class. Thus, recall, f-measure, g-mean and AUC (area under curve) are used in this paper.

$$\begin{aligned} precision &= \frac{TP}{TP + FP} \\ recall &= \frac{TP}{TP + FN} \\ f - measure &= \frac{2 \times recall \times precision}{recall + precision} \\ g - mean &= \sqrt{\frac{TP \times TN}{(TP + FN) \times (TN + FP)}} \end{aligned} \quad (24)$$

where TP, TN, FN and FP are the number of true positives, true negatives, false negatives and false positives.

The basic information of real-world datasets is shown in Table S1 that collected from UCI repository [38] and [39]. We only care binary classifications so only two classes included in each dataset. Before experience, those different species of dataset are preprocessed by the standardized z-scores. We select two base classifiers respectively as SVM and neural network (NN, 10 hidden nodes), and use a stratified 2-fold cross validation for 35 times, resulting in 70 runs for each dataset. Using the SVM as the classifier, Table 1 shows its average performance of g-mean over all datasets when; average performances of precision, recall, f-measure and auc are shown in Table S2-S5 (in the supplementary materials). Using NN as the classifier, average performances of precision, recall, f-measure, g-mean and auc are shown in Table S6-S10.

As shown in Table 1, each table cell includes the mean and standard deviation of 70 runs, and the rank of technique in a bracket. Where the best rank in each row is highlighted as bold. For example, we run SMOTE on Biomed diseased (the second dataset) for 70 times that resulting in 70 g-mean values, then record their mean and standard deviation as 0.8578 and 0.0312. This mean value is the 5-th best, thus assign its rank as 5 in a bracket. Except selected over-sampling methods, we also reserve the result on original data for comparison, calling corresponding method as Ori in which the classifier is directly trained on original imbalanced dataset.

For a rough comparison, as shown in Fig. 9, we count the time for each method that achieves the best rank over

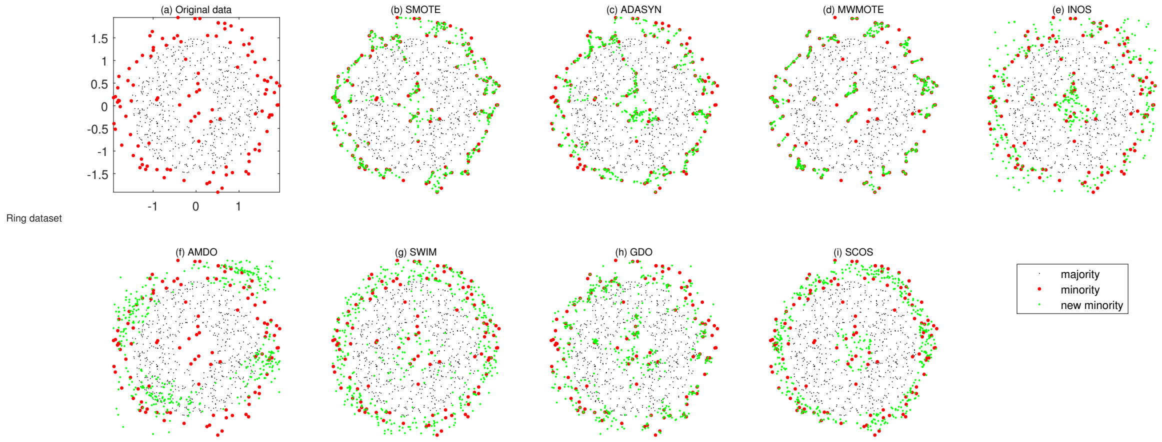


Fig. 7: Synthetic data on Ring dataset. Adding it with several noises.

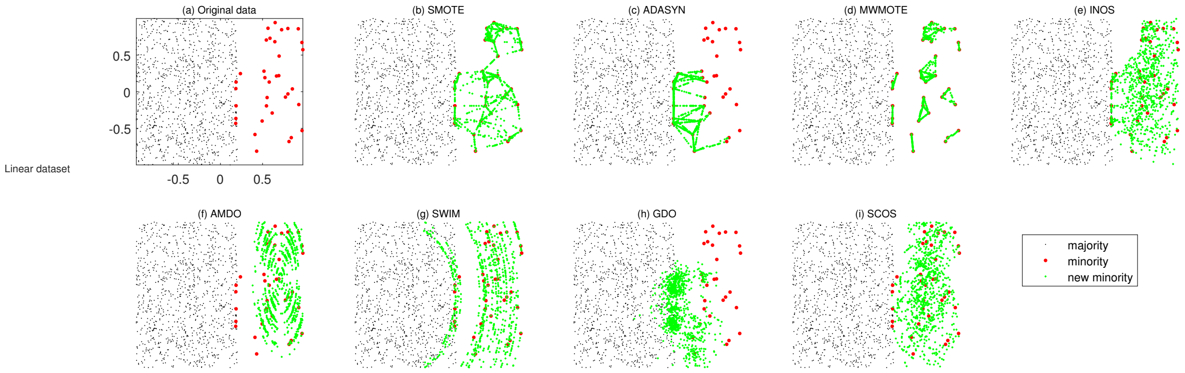


Fig. 8: Synthetic data on Linear dataset. Adding it with several overlapped points.

all datasets. For example, in g-mean, SCOS achieves the best rank over 21 datasets; its best-rank count is 21. Additionally, as shown in Table 2 and 4, we compute the corresponding mean ranks over all datasets. Where the best mean rank in a row is highlighted as bold. Besides, we use the Friedman test to judge whether the difference exists among all techniques; and use the posthoc Bonferroni-Dunn test to judge whether the difference exists between the proposed technique and one of other technique when the the proposed one obtains the best mean rank. Using the Friedman test, $reject$ denotes the difference exists among all method, $accept$ denotes not. Using the posthoc Bonferroni-Dunn test, the gap between two mean ranks that larger than the critical value (CD=1.67), denoting the difference exists between two methods. For example on g-mean with SVM, using the Friedman test, the actual value is 169.58 that larger than the table-lookup value 15.51, thus the difference exists among all techniques; using the posthoc Bonferroni-Dunn test, the mean rank of proposed one is 1.85, the mean rank of SMOTE is 5.45, their gap is 3.6 that larger than 1.67, thus the difference exists between two methods and label one symbol † on the top right of 5.45.

After the posthoc Bonferroni-Dunn test, there exists some methods that their gaps is smaller than CD. Thus, as show in Table 3 and 5, we use the Wilcoxon paired signed-rank test to judge whether the difference exists between the proposed technique and one of remained techniques. $p-value$ below 0.05 denotes the difference exists, and highlight it as bold.

4.3 Performance analysis

From above results, over-sampling methods improve the mean rank of recall but degrade the mean rank of precision when compared to Ori. It is straightforward since more samples are classified as the minority class both including the minority and majority than Ori. Thus, the good recall means many minority being truly classified and the bad precision means some majority being wrongly classified. Good f-measure, g-mean and auc mean a good balance between truly classified minority and wrongly classified majority.

From both mean ranks of SVM and NN classifiers, the one proposed here, SCOS obtains the best mean rank on recall since many synthetic samples are generated near to the borderline between classes that more minority are truly

TABLE 1: SVM: average g-mean

Dataset	Ori	SMOTE	ADASYN	MWMOTE	INOS	AMDO	SWIM	GDO	SCOS
Survival ¡5yr	0.1121±0.1561(9)	0.5147±0.0627(4)	0.5088±0.0635(6)	**0.5337±0.0695(1)**	0.5233±0.0641(3)	0.5022±0.0732(8)	0.5128±0.0654(5)	0.5029±0.0680(7)	0.5260±0.0573(2)
Biomed diseased	0.8300±0.0375(9)	0.8578±0.0312(5)	**0.8627±0.0288(1)**	0.8618±0.0340(2)	0.8556±0.0324(6)	0.8352±0.0403(8)	0.8526±0.0339(7)	0.8614±0.0362(3)	0.8598±0.0325(4)
Cancer wpbc ret	0.5169±0.0998(9)	0.6308±0.0702(5)	0.6367±0.0618(3)	0.6212±0.0629(6)	0.6315±0.0522(4)	0.5283±0.0886(8)	0.6515±0.0562(2)	0.6004±0.0766(7)	**0.6520±0.0508(1)**
Diabetes absent	0.6256±0.0362(9)	0.7004±0.0273(6)	0.7069±0.0272(2)	**0.7072±0.0277(1)**	0.6937±0.0333(7)	0.6880±0.0286(8)	0.7027±0.0272(5)	0.7027±0.0273(4)	0.7053±0.0278(3)
Hepatitis normal	0.6178±0.1184(9)	0.6719±0.0894(6)	0.6729±0.0855(5)	0.6707±0.0823(7)	0.6856±0.0782(4)	0.6185±0.1185(8)	**0.7348±0.0692(1)**	0.7106±0.0753(3)	0.7313±0.0613(2)
Housing MEDV¿35	0.6863±0.0708(9)	0.8298±0.0570(6)	0.8392±0.0551(5)	0.8110±0.0621(7)	0.8403±0.0536(4)	0.7193±0.0615(8)	**0.8689±0.0314(1)**	0.8543±0.0457(3)	0.8620±0.0410(2)
Spectf 0	0.6939±0.0492(9)	0.7497±0.0514(4)	0.7567±0.0420(2)	0.7519±0.0488(3)	0.7450±0.0442(5)	0.7226±0.0453(8)	0.7392±0.0402(6)	0.7389±0.0463(7)	**0.7702±0.0360(1)**
Iris setosa	0.9911±0.0101(6.5)	0.9911±0.0101(6.5)	0.9911±0.0101(6.5)	0.9911±0.0101(6.5)	0.9913±0.0101(4)	0.9934±0.0096(2)	0.9925±0.0098(3)	0.9903±0.0101(9)	**1.0000±0.0000(1)**
Vowel 3	0.6622±0.0781(9)	0.8153±0.0544(5)	0.8219±0.0554(3)	0.8135±0.0599(6)	0.8105±0.0526(7)	0.7812±0.0770(8)	0.8207±0.0504(4)	0.8223±0.0529(2)	**0.8262±0.0485(1)**
Vowel 8	0.0000±0.0000(9)	0.7828±0.0553(4)	0.7966±0.0496(2)	0.7620±0.0672(7)	0.7755±0.0522(6)	0.3220±0.2201(8)	0.7784±0.0478(5)	0.7936±0.0447(3)	**0.8064±0.0423(1)**
Waveform 0	0.7866±0.0270(9)	0.8382±0.0219(7)	0.8466±0.0217(4)	0.8408±0.0203(5)	0.8389±0.0217(6)	0.8331±0.0211(8)	0.8649±0.0137(2)	0.8519±0.0189(3)	**0.8676±0.0128(1)**
BreastTissue24	0.7458±0.0919(9)	0.8282±0.0607(7)	0.8368±0.0541(3)	0.8282±0.0546(6)	0.8315±0.0554(4)	0.8290±0.0543(5)	0.8370±0.0453(2)	0.8280±0.0521(8)	**0.8442±0.0513(1)**
BreastTissue3	0.0000±0.0000(9)	0.6242±0.1052(5)	0.6373±0.0903(2)	0.6137±0.1045(6)	0.5885±0.1416(7)	0.0066±0.0550(8)	**0.6649±0.0803(1)**	0.6254±0.0883(4)	0.6363±0.0823(3)
Ecoli2	0.7931±0.0447(9)	0.8413±0.0377(6)	0.8577±0.0370(3)	0.8408±0.0417(7)	0.8420±0.0370(5)	0.8274±0.0415(8)	**0.8778±0.0303(1)**	0.8679±0.0335(2)	0.8514±0.0364(4)
Ecoli3	0.7171±0.0691(9)	0.8585±0.0454(7)	0.8771±0.0384(2)	0.8587±0.0409(6)	0.8613±0.0416(5)	0.8427±0.0510(8)	**0.8815±0.0393(1)**	0.8738±0.0400(3)	0.8681±0.0421(4)
Glass5	0.4526±0.1947(8.5)	0.7620±0.1497(5)	0.7586±0.1514(6)	0.7574±0.1535(7)	0.8110±0.1339(4)	0.4526±0.1947(8.5)	**0.8868±0.0620(1)**	0.8516±0.1026(3)	0.8797±0.0899(2)
Glass7	0.8736±0.0746(9)	0.8921±0.0650(6)	0.8965±0.0632(4)	0.8832±0.0665(7)	0.8977±0.0604(3)	0.8799±0.0662(8)	**0.9178±0.0383(1)**	0.8961±0.0653(5)	0.9160±0.0442(2)
ImageSegmentation7	0.9947±0.0021(7.5)	0.9955±0.0024(5)	0.9958±0.0026(4)	0.9947±0.0021(7.5)	0.9965±0.0023(2)	0.9955±0.0023(6)	0.9962±0.0026(3)	0.9796±0.0158(9)	**0.9969±0.0025(1)**
ImageSegmentation5	0.6704±0.0221(9)	0.8707±0.0164(5)	0.8907±0.0100(2)	0.8590±0.0162(7)	0.8671±0.0163(6)	0.6941±0.2721(8)	0.8727±0.0129(4)	**0.8948±0.0092(1)**	0.8812±0.0134(3)
LibrasMovement11	0.3422±0.1884(9)	0.6670±0.1251(5)	0.6657±0.1269(6)	0.6611±0.1241(7)	0.6734±0.1385(4)	0.3491±0.1959(8)	**0.8378±0.0890(1)**	0.6895±0.1330(3)	0.7486±0.1277(2)
LibrasMovement15	0.6378±0.0958(9)	0.7785±0.0980(5)	**0.7995±0.0838(1)**	0.7820±0.0982(4)	0.7134±0.0983(7)	0.6588±0.0927(8)	0.7677±0.0907(6)	0.7930±0.0920(2)	0.7898±0.0923(3)
Pageblocks35	0.4748±0.0717(9)	0.8976±0.0288(3)	0.9086±0.0232(2)	0.8930±0.0280(4)	0.8750±0.0266(6)	0.8286±0.0447(7)	0.7643±0.0704(8)	**0.9211±0.0157(1)**	0.8832±0.0266(5)
StatlogVehicleSilhouettes3	0.9076±0.0154(9)	0.9377±0.0133(4)	0.9349±0.0168(7)	0.9363±0.0112(6)	0.9383±0.0111(3)	0.9366±0.0110(5)	**0.9542±0.0110(1)**	0.9314±0.0173(8)	0.9434±0.0125(2)
StatlogVehicleSilhouettes2	0.5244±0.0574(9)	0.7603±0.0262(3)	**0.7682±0.0250(1)**	0.7572±0.0277(5)	0.7497±0.0234(6)	0.6957±0.0299(8)	0.7641±0.0187(2)	0.7433±0.0248(7)	0.7589±0.0217(4)
WallFollowingRobotNavigation4	0.4709±0.0544(9)	0.8902±0.0130(4)	0.8630±0.0236(7)	0.8892±0.0147(6)	0.8921±0.0123(3)	0.8895±0.0137(5)	**0.9079±0.0082(1)**	0.8430±0.0174(8)	0.9057±0.0099(2)
Yeast789	0.3086±0.0822(9)	0.7076±0.0380(3)	0.6970±0.0375(5)	0.7017±0.0375(4)	0.6928±0.0440(6)	0.6070±0.0856(8)	0.6843±0.1250(7)	0.7083±0.0425(2)	**0.7117±0.0362(1)**
Yeast56	0.6096±0.0664(9)	0.8848±0.0258(5)	0.8827±0.0235(6)	0.8802±0.0301(7)	0.8853±0.0265(4)	0.8334±0.0341(8)	**0.9038±0.0164(1)**	0.8886±0.0197(3)	0.8890±0.0217(2)
DMEAntiVirus	0.9534±0.0280(6.5)	0.9534±0.0280(6.5)	0.9534±0.0280(6.5)	0.9534±0.0280(6.5)	0.9534±0.0276(4)	0.8839±0.2484(9)	0.9563±0.0285(2)	0.9555±0.0272(3)	**0.9861±0.0082(1)**
ParkinsonsDC	0.6837±0.0424(6.5)	0.6837±0.0424(6.5)	0.6837±0.0424(6.5)	0.6837±0.0424(6.5)	0.6812±0.0393(9)	0.6838±0.0420(4)	0.6993±0.0312(2)	0.6911±0.0419(3)	**0.7414±0.0292(1)**
GLRCWL1	0.6276±0.1187(6)	0.6276±0.1187(6)	0.6276±0.1187(6)	0.6276±0.1187(6)	0.6257±0.1173(9)	0.6276±0.1187(6)	0.6604±0.1182(2)	0.6280±0.1184(3)	**0.7421±0.0954(1)**
GLRCWL2	0.3062±0.2366(7)	0.3062±0.2366(7)	0.3062±0.2366(7)	0.3062±0.2366(7)	0.3114±0.2334(3)	0.3062±0.2366(7)	0.4713±0.2186(2)	0.3088±0.2332(4)	**0.5790±0.1397(1)**
GLRCNBI1	0.6408±0.1414(6)	0.6408±0.1414(6)	0.6408±0.1414(6)	0.6408±0.1414(6)	0.6389±0.1582(9)	0.6408±0.1414(6)	0.7000±0.1300(2)	0.6448±0.1458(3)	**0.7535±0.1039(1)**
GLRCNBI2	0.3327±0.2235(7)	0.3327±0.2235(7)	0.3327±0.2235(7)	0.3327±0.2235(7)	0.3351±0.2272(4)	0.3327±0.2235(7)	0.5043±0.1418(2)	0.3390±0.2206(3)	**0.5708±0.1120(1)**
Colon 1	0.5868±0.1884(6)	0.5868±0.1884(6)	0.5868±0.1884(6)	0.5868±0.1884(6)	0.5892±0.1868(3)	0.5868±0.1884(6)	0.6574±0.1546(2)	0.5850±0.1873(9)	**0.7760±0.1007(1)**
Leukemia 1	0.7587±0.0886(6)	0.7587±0.0886(6)	0.7587±0.0886(6)	0.7587±0.0886(6)	0.7562±0.0898(9)	0.7587±0.0886(6)	0.8935±0.0869(2)	0.7601±0.0900(3)	**0.9538±0.0361(1)**
Metas 1	0.2743±0.1334(7)	0.2743±0.1334(7)	0.2743±0.1334(7)	0.2743±0.1334(7)	0.2751±0.1337(3)	0.2743±0.1334(7)	0.3529±0.1236(2)	0.2751±0.1335(4)	**0.4975±0.0810(1)**
DrivFace1	0.7484±0.0885(6)	0.7484±0.0885(6)	0.7484±0.0885(6)	0.7484±0.0885(6)	0.7231±0.0989(9)	0.7484±0.0885(6)	0.8960±0.0543(2)	0.7521±0.0878(3)	**0.9258±0.0520(1)**
DrivFace3	0.7214±0.0820(6)	0.7214±0.0820(6)	0.7214±0.0820(6)	0.7214±0.0820(6)	0.7063±0.0824(9)	0.7214±0.0820(6)	0.8919±0.0641(2)	0.7237±0.0830(3)	**0.9163±0.0543(1)**
ARBT6	0.0000±0.0000(6)	0.0000±0.0000(6)	0.0000±0.0000(6)	0.0000±0.0000(6)	0.0000±0.0000(6)	0.0000±0.0000(6)	**0.2227±0.3407(1)**	0.0000±0.0000(6)	0.0292±0.1059(2)
ARBT5	0.0000±0.0000(5.5)	0.0000±0.0000(5.5)	0.0000±0.0000(5.5)	0.0000±0.0000(5.5)	0.0000±0.0000(5.5)	0.0000±0.0000(5.5)	0.0000±0.0000(5.5)	0.0000±0.0000(5.5)	**0.4306±0.0947(1)**

A stratified k-fold cross validation (k=2 in experience) is used for 35 times that 70 (2 × 35) runs are conducted. Thus for each table cell, the mean and standard deviation of corresponding performance on 70 runs are first recorded and then its rank among all methods is followed in one bracket. The best rank for each row is highlighted as bold.

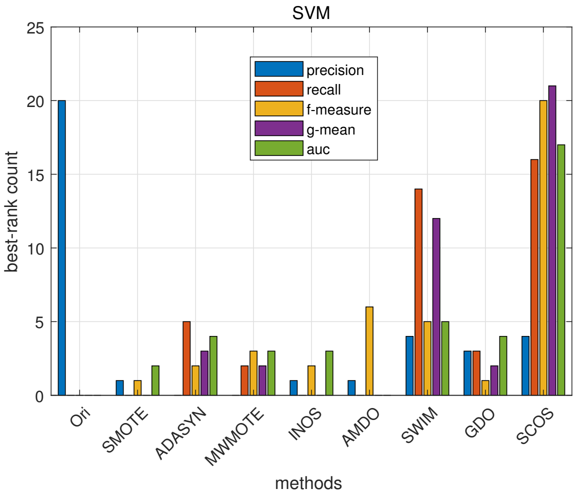


Fig. 9: Best-rank counts of different method over all real-world datasets.

TABLE 2: SVM: Mean ranks on all datasets

Measurement	Friedman test	Ori	SMOTE	ADASYN	MWMOTE	INOS	AMDO	SWIM	GDO	SCOS
precision	87.85(reject)	**2.94**	4.21	5.91	4.46	4.72	3.51	7.00	6.25	5.99
recall	175.80(reject)	7.99†	5.66†	4.41†	5.80†	5.44†	7.00†	2.49	4.16†	**2.05**
f-measure	87.22(reject)	7.55†	4.88†	5.22†	5.04†	4.66†	5.75†	4.21†	5.36†	**2.33**
g-mean	169.58(reject)	8.00†	5.45†	4.55†	5.74†	5.34†	7.00†	2.76	4.31†	**1.85**
auc	84.39(reject)	7.28†	4.30†	4.63†	4.81†	4.89†	6.69†	5.04†	4.75†	**2.63**
the Friedman Test: F=15.51, (n=9-1,alpha=0.05)										
the Bonferroni-Dunn test: critical values=1.67, (k=9,alpha=0.05)										

The best mean rank is highlighted as bold.

[reject] Using the Friedman Test with F=15.51 under (n=9-1,alpha=0.05) as one of the statistic test. If the actual value is larger than 15.51, reject the original hypothesis that significant difference exists among those methods.

[†] Using the Bonferroni-Dunn test with critical values=1.67 under (k=9, alpha=0.05) as one of the statistic test. If the gap of mean rank between two methods is larger than 1.67, reject the original hypothesis that significant difference exists between two methods and marked with †. There, we only consider the measurement in which the proposed method is of the best mean rank , and compare it with any one of other methods.

TABLE 3: SVM: Wilcoxon signed rank test (alpha=0.05)

recall		g-mean	
Ours vs.	p-V	Ours vs.	p-V
SWIM	0.4853	SWIM	**0.0091**

p-value under 0.05 is highlighted as bold which means the proposed one outperforms the compared one when using the Wilcoxon rank test (alpha=0.05)).

TABLE 4: NN: Mean ranks on all datasets

Measurement	Friedman test	Ori	SMOTE	ADASYN	MWMOTE	INOS	AMDO	SWIM	GDO	SCOS
precision	92.54(reject)	**3.31**	4.10	4.69	4.05	4.76	4.14	8.19†	5.72	6.04
recall	208.40(reject)	8.76†	4.47†	4.09†	5.76†	5.63†	7.72†	2.69	3.76	**2.11**
f-measure	112.85(reject)	7.49†	**3.08**	3.65	3.56	4.30	5.97†	7.17†	5.63	4.15
g-mean	160.54(reject)	8.51†	3.50	3.98†	4.59†	4.35†	7.17†	5.92†	4.83†	**2.15**
auc	113.88(reject)	7.40†	3.81	3.49	4.74	3.84	6.74†	6.96†	4.81	**3.21**
the Friedman Test: F=15.51, (n=9-1,alpha=0.05)										
the Bonferroni-Dunn test: critical values=1.67, (k=9,alpha=0.05)										

The best mean rank is highlighted as bold.

[reject] Using the Friedman Test with F=15.51 under (n=9-1,alpha=0.05) as one of the statistic test. If the actual value is larger than 15.51, reject the original hypothesis that significant difference exists among those methods.

[†] Using the Bonferroni-Dunn test with critical values=1.67 under (k=9, alpha=0.05) as one of the statistic test. If the gap of mean rank between two methods is larger than 1.67, reject the original hypothesis that significant difference exists between two methods and marked with †. There, we only consider the measurement in which the proposed method is of the best mean rank , and compare it with any one of other methods.

TABLE 5: NN: Wilcoxon signed rank test (alpha=0.05)

recall		g-mean		auc	
Ours vs.	p-V	Ours vs.	p-V	Ours vs.	p-V
SWIM	0.5923	SMOTE	**0.0026**	SMOTE	0.7588
GDO	**0.0045**			ADASYN	0.5578
				MWMOTE	0.1248
				INOS	0.1428
				GDO	0.5119

p-value under 0.05 is highlighted as bold which means the proposed one outperforms the compared one when using the Wilcoxon rank test (alpha=0.05)).

classified than other methods. Meanwhile, SCOS obtains the worse mean rank on precision than all other methods except SWIM that more majority are wrongly classified. In SWIM algorithm, although many synthetic samples are generated nearer to the borderline when the seed one is near to the borderline since using the same Mahalanobis distance from the majority class. Such a use may generate some overlapped synthetic samples when the Mahalanobis distance does not meet the true distribution of borderline of majority class.

For other over-sampling methods, on the one hand, SMOTE, MWMOTE, INOS and AMDO obtain better mean rank of precision than recall. In those four over-sampling methods, AMDO performs worst on the mean rank of recall. Because AMDO only picks the minority with enough minority as its nearest neighbours that ignoring many minority nearer to the borderline, that resulting in less synthetic samples nearer to the borderline. On the other hand, ADASYN and GDO obtain better mean rank of recall than precision. In those two over-sampling methods, GDO performs better on the mean rank of recall. Because, GDO uses a Gaussian distribution model for each seed minority that many synthetic samples are generated nearer to the borderline than ADASYN in which only line segment is used to generate synthetic samples.

In generally, SCOS obtains the best mean ranks of g-mean and auc on both classifiers, and the best mean rank of f-measure on SVM. Which means its good balance between truly classified minority and wrongly classified majority; and better classification performance than other over-sampling methods on SVM.

5 DISCUSSION

5.1 Parameter settings

1) $\delta = -0.7660$: it is one threshold , ranged in [-1,1], to be set for the judgement of direct-interlinked relationship between two samples. For example, given two minority $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, if their minimum value of inner product satisfying $M(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq \delta$, they are direct-interlinked.

For one seed minority point, if no one point is direct-interlinked to it, SCOS considers it as the noise. As seen in Fig. S3, different settings of δ product different noise cleaning results. Obviously, smaller δ caused weak robustness to noises; and larger δ earns strong robustness to noises, but wrongly detecting some overlapped or borderline points as noises at the same time. As seen in Fig. S4, different g-mean and auc results are plotted with varying δ on several picked datasets. In experience, we set $\delta = -0.7660$. This is straightforward since it makes the judgement of direct-interlinked relationship more adaptive for different densities of majority distribution.

2) $k = 7$: It means k-nearest neighbours of majority points for the seed minority one. We use k to compute a referred direction from the center of those neighbours to the seed one. This referred direction is intended to start from the majority area and pointed to the minority area. In other words, we expect the center locating deeper in the majority area than the seed minority one.

As seen in Fig. S5, different settings of k product different robustness to the overlapped point. Small k like $k = 1$ or $k = 2$ makes the corresponding center near to seed overlapped one, so bringing much uncertainty of referred vector. And large k makes the the corresponding center deeper in the majority area, giving one referred vector pointed to the minority area. Except overlapped points, other normal minority points does not highly depend on this parameter k. For example, $k = 1$ can just product one referred vector with the good verticality to the borderline. As seen in Fig. S6, different g-mean and auc results are plotted with varying k on several picked datasets. To considering overlapped points, in experience, we set $k = 7$.

3) $\rho = 0.5$: This parameter, ranged in [-1,1], controls the size of cone angle of searchlight structure. As seen in Fig. 2, when $\rho = -1$, the searchlight structure is one Euclidean ball; when $\rho = 0$, is one half Euclidean ball; when $\rho = 1$, it is a line segment. As seen in Fig. S7, different g-mean and auc results are plotted with varying ρ on several picked datasets. Obviously, ρ values larger than 0.8 product decrease performance on g-mean and auc. In experience, we set $\rho = 0.5$.

5.2 Robustness to complex distributions

Although minority samples may distribute over multi areas, calling this scene as disjuncts, their searchlight structures just follow them into corresponding areas. Because we use direct-interlinked minority samples to compute the vertex of searchlight structure, which always makes this structure located in the same area as the seed minority sample. Thus, SCOS is naturally robust to disjuncts.

In the discussion of parameter k, SCOS is robust to the overlapped minority sample when the center of its k nearest majority locates in the deeper majority area that making the direction pointed to the minority area. In many cases, for the seed minority one, its k nearest majority just locates deeper, producing one deeper center into the majority area; thus robust to many overlapped minority samples.

Noises are generally surrounded by majority samples. Thus, their line segments to any one of other minority must go through the majority area, so no minority sample is direct-interlinked to them. Thus, SCOS is robust to most noises, especially for the scene that the majority area is fully filled with points.

5.3 Drawbacks

First, the fixed threshold ρ may fail to give an accurate relationship when the density of majority area is sparse. Because the judgement of relationship is based on the hypothesis that majority area is filled with enough points. Second, for the overlapped minority, SCOS may give one inappropriate direction that pointed to the majority area when the center of its nearest majority does not locate deeper in the majority area. In other words, many of its nearest majority samples do not locate deeper in the majority area than the seed overlapped one.

5.4 Computation efficiency

We run all experience on Matlab 2017b, Windows 10, 64 bits, Core i9 CPU, RAM 32.0 GB. Table 6 shows the time consuming of different over-sampling methods. For SCOS, cost time of five datasets exceed 10 seconds respectively as ParkinsonsDC (dimension: 754, minority and majority number: 192 and 564) with 17.03 seconds, DrivFace1 (dimension: 6399, minority and majority number: 27 and 579) with 18.87 seconds, DrivFace3 (dimension: 6399, minority and majority number: 33 and 573) with 19.69 seconds, ARBT6 (dimension: 8265, minority and majority number: 12 and 578) with 21.31 seconds, ARBT5 (dimension: 8265, minority and majority number: 31 and 559) with 24.54 seconds. Obviously, as the dimension and the number of minority samples increase, SCOS cost more time. Because for high dimensions, it is time-consuming to generate one condition-satisfied unit vector in Eq. 22. For other methods, AMDO and SWIM cost more than 10 seconds on some high dimension datasets. Especially, INOS and GDO cost more than 100 seconds on some high dimension datasets.

TABLE 6: Comparison of computation time (second) of different methods for real-world data sets

Dataset	SMOTE	ADASYN	MWMOTE	INOS	AMDO	SWIM	GDO	SCOS
Survival ¡5yr	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.02
Biomed diseased	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
Cancer wpbc ret	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.03
Diabetes absent	0.00	0.00	0.01	0.02	0.00	0.00	0.01	0.23
Hepatitis normal	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01
Housing MEDV¿35	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.04
Spectf 0	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.11
Iris virginica	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
Vowel 8	0.00	0.00	0.00	0.03	0.00	0.00	0.01	0.03
Vowel 5	0.00	0.00	0.00	0.02	0.00	0.00	0.01	0.03
Waveform 1	0.00	0.00	0.01	0.03	0.00	0.00	0.01	0.94
BreastTissue3	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
BreastTissue25	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
Ecoli5	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01
Ecoli3	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.02
Glass7	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.01
Glass2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02
ImageSegmentation1	0.00	0.01	0.01	0.18	0.01	0.00	0.03	3.67
ImageSegmentation7	0.00	0.01	0.01	0.27	0.01	0.00	0.03	3.67
LibrasMovement11	0.00	0.00	0.00	0.15	0.00	0.00	0.01	0.08
LibrasMovement8	0.00	0.00	0.00	0.12	0.00	0.00	0.01	0.08
Pageblocks35	0.01	0.07	0.12	18.30	0.10	0.00	0.09	1.17
StatlogVehicleSilhouettes1	0.01	0.02	0.04	1.41	0.02	0.00	0.01	0.52
StatlogVehicleSilhouettes2	0.01	0.02	0.04	1.23	0.02	0.00	0.01	0.55
Wine2	0.00	0.01	0.01	0.00	0.00	0.00	0.00	0.03
Yeast679	0.01	0.01	0.02	2.82	0.02	0.00	0.02	0.18
Yeast567	0.01	0.02	0.03	2.35	0.02	0.00	0.02	0.25
DMEAntiVirus	0.03	0.07	0.11	1.29	0.05	0.03	0.27	1.34
ParkinsonsDC	0.02	0.16	0.26	3.98	0.07	0.03	0.88	17.03
GLRCWL1	0.02	0.01	0.06	0.31	0.01	0.02	0.07	0.14
GLRCWL2	0.02	0.03	0.06	0.44	0.01	0.02	0.09	0.14
GLRCNBI1	0.02	0.01	0.06	0.32	0.01	0.02	0.07	0.14
GLRCNBI2	0.02	0.03	0.06	0.44	0.01	0.03	0.09	0.14
Colon 1	0.01	0.03	0.07	0.82	0.21	0.21	0.27	0.20
Leukemia 1	0.02	0.06	0.11	5.11	2.80	2.42	1.46	0.46
Metas 1	0.02	0.19	0.13	16.66	2.44	7.99	7.37	2.88
DrivFace1	0.10	0.09	0.18	138.43	22.39	17.69	141.89	18.87
DrivFace3	0.08	0.14	0.21	174.23	23.35	17.67	138.77	19.69
ARBT6	0.24	0.06	0.03	394.77	0.03	17.49	308.92	21.31
ARBT5	0.03	0.07	0.03	380.54	0.04	17.40	288.37	24.54

6 CONCLUSION

In this paper, we present one novel over-sampling method (SCOS) to address the class imbalance problem with complex distributions, like disjuncts, class overlapping and noises. Inspired by the scene of objective area scanning in real life, we use a series of searchlight structures to scan the minority area and fill them with synthetic samples, where the searchlight structure is modelled with four components including one base unit vector, one vertex, one radius and one restricted scalar value of inner product. To compute the searchlight structure, SCOS tactfully treats the majority area as the barrier of buildings; and makes the light cone first launched from the minority area, then through the nearby area of seed minority and last stopped by the majority area. Moreover, to cover more boundary area, SCOS products a nearly vertical illumination on the surface of borderline majority samples. Additionally, this study gives one new finding that a pair of samples in the same class tends to distribute over a continuing area when their line segment does not go through the area of other classes, providing a new view on the judgement of relationship between two same-class points.

Visualization results on emulational datasets show the satisfied capability of SCOS to complex distributions. Classification results on real-world datasets show the superior learning performance of proposed method when compared with selected stat-of-the-art over-sampling methods. In the future, some works will be attached to the better robustness when meeting different densities of majority area, and different distributions of class overlapping.

ACKNOWLEDGMENTS

REFERENCES

[1] C. Seiffert, T. M. Khoshgoftaar, J. V. Hulse, and A. Folleco, “Dan empirical study of the classification performance of learners on imbalanced and noisy software quality data,” *Information Sciences.*, vol. 259, pp. 571– 595, 2014.

[2] S. Wang, L. L. Minku, and X. Yao, “Resampling-based ensemble methods for online class imbalance learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 5, pp. 1356–1368, May 2015.

[3] H. Zhang, W. Liu, and Q. Liu, “Reinforcement online active learning ensemble for drifting imbalanced data streams,” *IEEE Trans. Knowl. Data Eng.*, pp. 2020, doi: 10.1109/TKDE.2020.3 026 196.

[4] A. D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, “Credit card fraud detection: A realistic modeling and a novel learning strategy,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. PP, pp. 1–14, 2017.

[5] H. Lin, G. Liu, J. Wu, Y. Zuo, X. Wan, and H. Li, “Fraud detection in dynamic interaction network,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 10, pp. 1936–1950, Oct. 2020.

[6] J. Jia, L. Zhai, W. Ren, L. Wang, and Y. Ren, “An effective imbalanced jpeg steganalysis scheme based on adaptive cost-sensitive feature learning,” *IEEE Trans. Knowl. Data Eng.*, pp. 2020, doi: 10.1109/TKDE.2020.2 995 070.

[7] A. Tayal, T. F. Coleman, and Y. Li, “Rankrc: Large-scale nonlinear rare class ranking,” *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 12, pp. 3347–3359, Dec. 2015.

[8] J. Hu, H. Yang, M. R. Lyu, I. King, and A. M. So, “Online nonlinear auc maximization for imbalanced data sets,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 882–895, Apr. 2018.

[9] C. Huang, C. C. Loy, and X. Tang, “Discriminative sparse neighbor approximation for imbalanced learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1503–1513, May 2018.

[10] J. Mathew, C. K. Pang, M. Luo, and W. H. Leong, “Classification of imbalanced data by oversampling in kernel space of support vector machines,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4065–4076, Sep. 2018.

[11] C. T. Li, T. Y. Liu, Y. Y. Lin, C. N. Fang, Y. K. Wang, G. Wang, N. R. Pal, and C. H. Chuang, “Minority oversampling in kernel adaptive subspaces for class imbalanced datasets,” *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 5, pp. 950–962, May 2018.

[12] X. Zhang, D. Ma, L. Gan, S.Jiang, and G. Agam, “Cgmos: Certainty guided minority oversampling,” in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Indianapolis, IN, USA, 2016.

[13] R. Razavi-Far, M. Farajzadeh-Zanjani, B. Wang, M. Saif, and S. Chakrabarti, “Imputation-based ensemble techniques for class imbalance learning,” *IEEE Trans. Knowl. Data Eng.*, pp. 2019, doi: 10.1109/TKDE.2019.2 951 556.

[14] A. Sen, M. Islam, K. Murase, and X. Yao, “Binarization with boosting and oversampling for multiclass classification,” *IEEE Trans. Cybern.*, vol. 46, no. 5, pp. 1078–1091, May 2016.

[15] C. Liu and P. Hsieh, “Model-based synthetic sampling for imbalanced data,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 8, pp. 1543–1556, Aug. 2020.

[16] E. R. Q. Fernandes, A. C. P. L. F. de Carvalho, and X. Yao, “Ensemble of classifiers based on multiobjective genetic sampling for imbalanced data,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1104–1115, Jun. 2020.

[17] B. Cao, Y. Liu, C. Hou, J. Fan, B. Zheng, and J. Yin, “Expediting the accuracy-improving process of svms for class imbalance learning,” *IEEE Trans. Knowl. Data Eng.*, pp. 2020, doi: 10.1109/TKDE.2020.2 974 949.

[18] M. Z. Jan, J. C. Munoz, and M. A. Ali, “A novel method for creating an optimized ensemble classifier by introducing cluster size reduction and diversity,” *IEEE Trans. Knowl. Data Eng.*, pp. 2020, doi: 10.1109/TKDE.2020.3 025 173.

[19] S. Ren, W. Zhu, B. Liao, Z. Li, P. Wang, K. Li, M. Chen, and Z. Li, “Selection-based resampling ensemble algorithm for nonstationary imbalanced stream data learning,” *Knowledge-Based Systems.*, vol. 163, pp. 705–722, Jan. 2019.

[20] S. Datta, S. Nag, and S. Das, “Boosting with lexicographic programming: Addressing class imbalance without cost tuning,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 5, pp. 883–897, May 2020.

[21] A. Manukyan and E. Ceyhan, “Classification of imbalanced data with a geometric digraph family,” *J. Mach. Learn. Res.*, vol. 17, pp. 1–40, Jan. 2016.

[22] Y. S. anf K. Tang, S. W. L. L. Minku and, and X. Yao, “Online ensemble learning of data streams with gradually evolved classes,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 6, pp. 1532–1545, Jun. 2016.

[23] Q. Kang, X. Chen, S. Li, and M. Zhou, “A noise-filtered undersampling scheme for imbalanced classification,” *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4263–4274, Dec. 2017.

[24] L. Abdi and S. Hashemi, “To combat multi-class imbalanced problems by means of over-sampling techniques,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 238–251, Jan. 2016.

[25] A. Pourhabib, B. K. Mallick, and Y. Ding, “Absent data generating classifier for imbalanced class sizes,” *J. Mach. Learn. Res.*, vol. 16, pp. 2695–2724, Jan. 2015.

[26] Y. Xie, M. Qiu, H. Zhang, L. Peng, and Z. Chen, “Gaussian distribution based oversampling for imbalanced data classification,” *IEEE Trans. Knowl. Data Eng.*, pp. 2020, doi: 10.1109/TKDE.2020.2 985 965.

[27] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: Synthetic minority over-sampling technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[28] H. He, Y. Bai, E. A. Garcia, and S. Li, “Adasyn: Adaptive synthetic sampling approach for imbalanced learning,” in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2008, pp. 1322–1328.

[29] H. Cao, X. L. Li, Y. K. Woon, and S. K. Ng, “Spo: Structure preserving oversampling for imbalanced time series classification,” in *Proc. IEEE Int. Conf. on Data Mining*, 2011, pp. 1008–1013.

[30] L. Abdi and S. Hashemi, “To combat multi-class imbalanced problems by means of over-sampling techniques,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 238–251, Jan. 2016.

[31] H. Cao, X. L. Li, D. K. Woon, and S. K. Ng, “Integrated oversampling for imbalanced time series classification,” *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 12, pp. 2809–2822, Dec. 2013.

[32] S. Barua, M. M. Islam, X. Yao, and K. Murase, “Mwmote—majority weighted minority oversampling technique for imbalanced data set learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, Feb. 2014.

[33] P. Lim, C. K. Goh, and K. C. Tan, “Evolutionary cluster-based synthetic oversampling ensemble (eco-ensemble) for imbalance learning,” *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2850–2861, Sep. 2017.

[34] E. Voorhees, “Implementing agglomerative hierarchic clustering algorithms for use in document retrieval,” *Information Processing and Management*, vol. 22, no. 6, pp. 465–476, 1986.

[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. USA: Cambridge University Press, 2004.

[36] H. Han, W. Wang, and B. Mao, “Borderline-smote: A new over-sampling method in imbalanced data sets learning,” in *Proc. Int. Conf. Intelligent Computing.*, 2005, pp. 878–887.

[37] X. Yang, Q. Kuang, W. Zhang, and G. Zhang, “Amdo: An over-sampling technique for multi-class imbalanced problems,” *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1672–1685, Sep. 2018.

[38] D. Dua and K. T. Efi, “Uci machine learning repository,” University of California, Irvine, School of Information and Computer Sciences, 2017. [Online]. Available: http://archive.ics.uci.edu/ml.

[39] “One-class classifier results.” [Online]. Available: http://homepage.tudelft.nl/n9d04/occ/index.html

Yi Sun is currently pursuing the Ph.D. degree in computer science and technology with Hunan University, Changsha, China.

His current research interests include imbalanced learning and data mining.

Lijun Cai received the Ph.D. degree in computer application technology from Hunan University, Changsha, China.

He is currently a Full Professor of computer science and technology with Hunan University. His research interests include parallel computing, cloud computing, and image processing.

Bo Liao received the Ph.D. degree in computational mathematics from the Dalian University of Technology, Dalian, China, in 2004.

From 2004 to 2006, he was a Post-Doctoral Fellow with the University of Chinese Academy of Sciences, Beijing, China. He is currently a Full Professor with Hainan Normal University. He has authored over 100 papers in international conferences and journals. His research interests include image processing, bioinformatics, and big data processing.

Wen Zhu received the M.S. degree in computer science and technology from Hunan University, Changsha, China, in 2010.

She is currently a Lecturer with Hainan Normal University. Her research interests include image processing and analysis and bioinformatics.

JunLin Xu is currently pursuing the Ph.D. degree in computer science and technology with Hunan University, Changsha, China.

His research interests include bioinformatics, machine learning, biochemical research method, disease-associated non-coding RNAs, and single cell.