Journal of Cases on Information Technology
Volume 26 • Issue 1

Selecting Indispensable Edge Patterns With Adaptive Sampling and Double Local Analysis for Data Description

Huina Li, Xuchang University, China

Yuan Ping, Xuchang University, China*

https://orcid.org/0000-0001-7703-4637

ABSTRACT

Support vector data description (SVDD) inspires us in data analysis, adversarial training, and machine unlearning. However, collecting support vectors requires pricey computation, while the alternative boundary selection with O(N2) is still a challenge. The authors propose an indispensable edge pattern selection method (IEPS) for data description with direct SVDD model building. IEPS suggests a double local analysis to select the global edge patterns. Edge patterns belong to a subset of the target problem of SVDD and its variants, and neighbor analysis becomes pivotal. While an excessive number of participating data result in redundant computations, an insufficient number may impede data separability or compromise the model's quality. Consequently, a data-adaptive sampling strategy has been devised to ascertain an optimal ratio of retained data for edge pattern selection. Extensive experiments indicate that IEPS keeps indispensable edge patterns for data description while reducing the interference in the norm vector generation to guarantee the effectiveness for clustering analysis.

KEYWORDS

Adaptive Sampling, Cluster Analysis, Edge Pattern Selection, K-Means++, Support Vector Data Description

INTRODUCTION

Inspired by support vector classifier, support vector data description (SVDD) (Tax & Duin, 1999) characterizes a data set by obtaining the spherically shaped boundary. Through a model built to describe the target data set, it benefits a wide range of applications, such as image description (Aslani & Seipel, 2021), novelty discovery (Hu et al., 2023), adversarial training (C. Chen et al., 2023), and machine unlearning (M. Chen et al., 2023). However, in collecting support vectors (SVs) for data description, the conventional solution conducts model training through solving a quadratic programming optimization problem. It poses a computational complexity of $O(N^3)$ where N is the number of data points. Evidently, pricey computations may significantly degrade SVDD's applicability.

Let $\mathcal{X}$ be a data set with N data points $\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ where $\mathbf{x}_i \in \mathbb{R}^d (i \in [1, N])$ in data space. The pricey model training is generally caused by solving a quadratic programming problem in terms

DOI: 10.4018/JCIT.335945

*Corresponding Author

of iterative analysis on a $N \times N$ kernel matrix. Furthermore, the number of iterative analysis is usually large and uncertain, yet a great value for the final coefficient vector β exacerbates the practical time-cost. Efficient solver for the quadratic programming problem is the major preference for improvement, such as the solver of dual coordinate descent (Y. Ping et al., 2017). However, the computational complexity falling in the range of $O(N^2)$ and $O(N^3)$ upon the specific case is still pricey (Arslan et al., 2022). Another intuitive way of improvement is to select the most representative subset of $\mathcal{X}$. However, few works in the literature focus on the subset's representativeness or purity strongly related to SVDD. They frequently select a subset on the basis of random sampling, data geometry analysis, and neighborhood relationships. For instance, Kim et al. (2015) define a sample rate $\rho(0 < \rho < 1)$ to regulate the randomly selected ρN data points during model training, while Jung et al. (2010) and Gornitz et al. (2018) leverage data geometry information by incorporating *k*-means to partition $\mathcal{X}$ into *K* subsets for local model training and subsequent global mergence. However, these data points employed for model training, whether obtained through random sampling or cluster-based geometry analysis, may not accurately capture the true distribution of $\mathcal{X}$. Random sampling introduces changes in the densities of all the retained data groups that significantly impacts data separability. The circle-like pattern hypothesis employed in subsets collection for local model training may exacerbate the adverse effects of irregular cluster shapes. Despite achieving substantial efficiency improvements, these methods often result in highly unstable accuracies. As the superset of support vectors (SVs) (Y. Ping et al. 2015), boundary generally makes an equivalent contribution to the construction of demarcation hyperplanes (Chen et al., 2023). On the basis of neighborhood relationships, Aslani and Seipel (2021) introduce locality-sensitive hashing (LSH) to gather instances near decision boundaries and eliminate nonessential ones. However, it retains many inners that may be more suitable for constructing a classifier for multi-classes problems rather than describing clusters with arbitrary shapes. Furthermore, Y. Ping et al. (2015) and Y. Ping et al. (2019) utilize the boundary to directly reformulate the dual problem. Despite achieving stable performance, the boundary selection becomes computationally expensive with a large value of *N*.

As depicted by Figure 1, boundary consists of edge and border (Li & Maguire, 2011), which extend beyond the essential requirement for cluster discovery and description. Specifically, for unsupervised learning, boundary is effectively profiled by edge patterns alone since no shared borders should be considered. The border refers to the connection between two nearest neighboring clusters. Simultaneously, both of the kernelized SVDD (Cevikalp et al., 2020) and the convex decomposition strategy (Y. Ping et al., 2020) suggest that only a subset of samples on the decomposed convex hulls is necessary for accurate data description. Thus, an optimal edge pattern selection method should preserve cluster shapes, excluding inners and border patterns, and eliminate redundant instances even though they reside on edges, as they contribute nothing in additional contributions to cluster description.

Toward this requirement, our critical observations based on the principle of SVDD encompass three aspects: 1) Inners are exclusive to each cluster, as they can also be considered as outliers of other clusters; 2) Border patterns are solely shared by any two connected clusters or prototypes, such as convex hulls in (Y. Ping et al., 2019), and this sharing is independent of the clustering method; 3) Edge patterns represent the only type of data points shared by all clusters, regardless of the chosen clustering method. Motivated by these insights, we propose an indispensable edge pattern selection method (IEPS) with data adaptive sampling and double local analysis strategies. IEPS maximizes the utility of the shrinkable boundary selection algorithm (SBS) (Y. Ping et al., 2019), p-stable distributions based LSH (pdLSH) (Datar et al., 2004), and *k*-means++ (Arthur & Vassilvitskii, 2007). The main contributions lie in:

(1) We propose an edge pattern selection strategy with double local analysis (EPSDLA) based on two rounds of data partitioning using *k*-means++. The inherent instability of *k*-means++ is leveraged

as an advantage, resulting in deliberately inconsistent clusters. Edge patterns from distinct local clusters are subsequently collected separately to derive the global edge patterns—those shared by both partitions. Despite encountering numerous challenges, the simplicity, efficiency, and instability of *k*-means++ are well integrated.

(2) To eliminate redundant instances along edges and prevent an undue inhibition to edge patterns, which potentially affect future connectivity analysis, we introduce a data-adaptive sampling strategy (DASS) based on pdLSH to assess data separability and recommends a data-related sample rate ρ for data reduction, ensuring the preservation of data patterns for EPSDLA. Given ρ, the maximin and random sampling (MMRS) can confidently be employed as a preprocessing step for large-scale data analysis.

(3) By integrating the aforementioned designs, IEPS is proposed to enhance the understanding of SVDD by efficiently collecting essential edge patterns. These patterns are crucial for subsequent hypersphere and support function construction, as well as for cluster analysis. To accommodate flexible parameter evaluation, IEPS collects global edge patterns from the retained subset of $\mathcal{X}$, ensuring that this process does not compromise the effectiveness of their convergence directions. Experimental results affirm its substantial efficiency improvement compared to the well-known border-edge pattern selection method (BEPS) (Li & Maguire, 2011). Utilizing the edge patterns collected by IEPS as input, both the fast and scalable support vector clustering (FSSVC) (Y. Ping et al. 2015) and the improved boundary support vector clustering (IBSVC) (Li et al. 2022) perform well in terms of accuracy.

The remainder is organized as follows: Section 2 (Preliminaries) briefly describes SVDD, *k*-means++, SBS, MMRS, and pdLSH. In Section 3 (The Proposed IEPS Method), we present EPSDLA, DASS, and then the architecture of IEPS as well as its full implementation. Section 4 (Performance Analysis) gives performance analysis through a series of experiments. Finally, the conclusions are drawn in the last section (Conclusion), as well as the future works to be investigated.

PRELIMINARIES

Support Vector Data Description

Given a nonlinear function $\Phi(\cdot)$ to map data points into the feature space, SVDD tries to find a minimum sphere with centers α and radius *R* which contains most of the data points. Its objective function can be formulated by:

$$\begin{aligned}&\min_{R,\alpha,\xi_i} R^2 + C\sum_i \xi_i \\ &s.t. \, || \Phi(x_i) - \alpha ||^2 \le R^2 + \xi_i\end{aligned} \tag{1}$$

where ξ_i is a slack variable, and *C* gives the trade-off between simplicity and the number of errors. Following Tax and Duin (1999), the dual problem of Eq. (1) can be formulated by

$$\begin{aligned}&\min_{\beta_j} \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{s.t.} \sum_j \beta_j = 1,\ 0 \le \beta_j \le C,\ i,j = 1,\ldots,N\end{aligned} \tag{2}$$

By optimizing Eq.(2) with Gaussian kernel $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-q\|\mathbf{x}_i - \mathbf{x}_j\|^2}$, the objective trained support function can be formulated by the squared radial distance of the image of $\mathbf{x}$ from the sphere center α given by:

$$f(\mathbf{x}) = 1 - 2\sum_j \beta_j K(\mathbf{x}_j, \mathbf{x}) + \sum_{i,j} \beta_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j), \tag{3}$$

where:

$$\alpha = \sum_j \beta_j \Phi(\mathbf{x}_j) \tag{4}$$

Apparently, the center α is a linear combination of the mapped data points with weight factors β_j. Since only a part of data points has $\beta_j > 0$, not all the data points contribute to the center construction that is similar to the construction of cluster centers in *k*-means. Meanwhile, only a small set of data points have $0 < \beta_i < C$ which are SVs locating on the boundary. So, only SVs are essential for describing the sphere, as well as shapes and connection relationship of clusters.

Various labeling strategies have been introduced to complement SVDD, resulting in a series of support vector clustering (SVC) approaches that enhance research and applications in unsupervised learning, such as the earliest SVDD plus a complete graph strategy (CG) (Ben-Hur et al., 2001), the faster and reformulated SVC (FRSVC) combining a solver reformulated SVDD and convex decomposed cluster labeling strategy (Y. Ping et al., 2017), and the Voronoi cell-based clustering (VCC) (Kim et al., 2015) integrating SVDD and a Voronoi cell-based labeling strategy.

K-Means++

Let $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_K\}$ be the expected *K* cluster centers of $\mathcal{X}$. Let $Z = [z_{iv}]_{N \times K}$, where $z_{iv} \in \{0,1\}$ indicates whether $\mathbf{x}_i$ belongs to the v-th cluster and $v = 1, \cdots, K$. Thus, the objective function of *k*-means can be formulated by:

$$\min_{Z,C} \sum_{i=1}^{N} \sum_{v=1}^{K} z_{iv} \| x_i - c_v \|^2 \tag{5}$$

To reach the objective, *k*-means++ chooses the first center $\mathbf{c}_1$ uniformly at random from $\mathcal{X}$, and repeatedly chooses the next center $\mathbf{c}_i = \mathbf{x}' \in \mathcal{X}$ with probability $D(\mathbf{x}')^2 / \sum_{\mathbf{x} \in \mathcal{X}} D(\mathbf{x})^2$ until a total of *K* centers are initialized. Here, $D(\mathbf{x})$ is the shortest distance from $\mathbf{x}$ to the closest center we have already chosen. Then, the solver of *k*-means++ iteratively updates the cluster centers and memberships formulated by the following equations, respectively.

$$c_v = \frac{\sum_{i=1}^{N} z_{iv} x_{ij}}{\sum_{i=1}^{N} z_{iv}} \qquad z_{iv} = \begin{cases} 1 & if \ \| x_i - c_v \|^2 = \min_{1 \le v \le K} \| x_i - c_v \|^2 \\ 0 & otherwise \end{cases} \tag{6}$$

Here, $\|\mathbf{x}_i - \mathbf{c}_v\|$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{c}_v$. Apparently, K initial cluster centers are critical for the iterative analysis, which will end with K final clusters. So K is the prior knowledge, and the initial centers selected are the root of its unstable performance. Furthermore, the employed Euclidean distance leads to the circle-like pattern description.

Shrinkable Boundary Selection

As shown in Figure 1, cluster boundary consists of edge and border (Li & Maguire, 2011). A cluster has independent edges, while any two clusters with overlapping region share the same border. We usually consider the latter as two components of a cluster. Even though border patterns are important for connectivity analysis, it is unnecessary to extract them for SVDD, in which edge patterns are the most informative data points for accurate data description. Derived from BEPS, for a given point $\mathbf{x}_i$ with its k_e nearest neighbors $\mathbf{x}_j (j = 1,2,\dots,k_\mathrm{e})$, SBS (Y. Ping et al., 2019) includes the follow four phases.

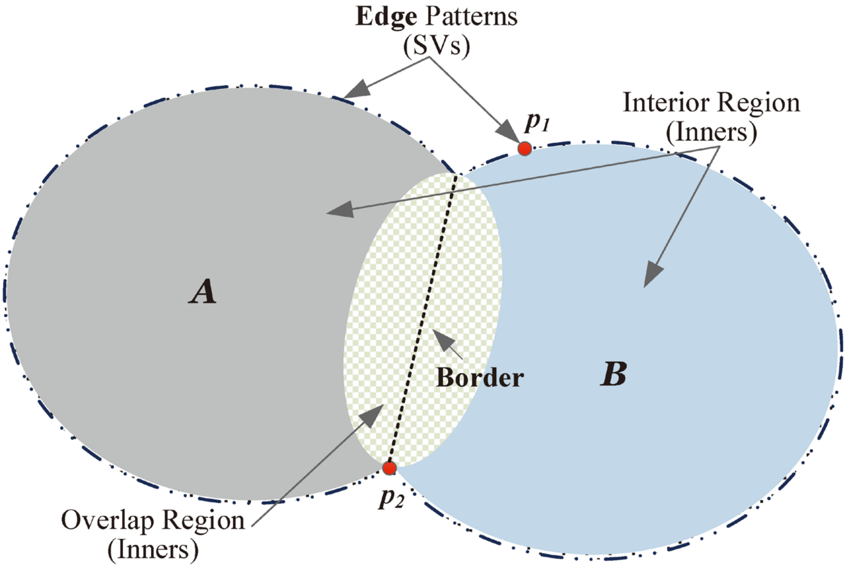
(1) Setting two thresholds γ_l and γ_u $(0 < \gamma_l < \gamma_u \le 1)$ to respectively control the curvature and the shrinkage degree of the surface above.

(2) Generating the normal vector $\mathbf{n}_i = \sum_{j=1}^{k_\mathrm{e}} \mathbf{u}_{ij}$, where $\mathbf{u}_{ij} = \mathbf{x}_j - \mathbf{x}_i$

(3) Calculating $l_i = \dfrac{1}{k_\mathrm{e}} \sum_{j=1}^{k_\mathrm{e}} g(\mathbf{n}_i^T \cdot \mathbf{u}_{ij})$, where $(\cdot)$ means inner product and the function $g(x)$ returns 1 if $x \ge 0$; otherwise it returns 0.

(4) Cluster boundary identification. If $l_i \in [\gamma_l, \gamma_u]$, then $\mathbf{x}_i$ is considered as one of the boundary points. Generally, the closer a data point is to the cluster center, the more balance the surrounding neighbors will be. Therefore, we can reduce γ_u to shrink the extracted boundary.

Figure 1. Edge patterns and border



Maximin and Random Sampling

Toward accurately portraying the distribution of $\mathcal{X}$ with a portion of data points, MMRS (Rathore et al., 2019) samples $n = \rho N(\rho \in (0,1))$ data points through the following three steps:

Step 1: *Collect* N_g *maximin data points.* MMRS identifies the *K* maximin data points in $\mathcal{X}$, which are furthest from each other. Starting with a randomly chosen data point, it selects the second maximin data point that is the furthest from the initial point with respect to a chosen distance measure. The third data point chosen should have the maximum distance from the first two, and this process continues until *K* maximin samples are collected.

Step 2: *Group each data sample with its nearest maximin data point.* By grouping each data point in $\mathcal{X}$ with its nearest maximin data point, we get N_g groups of data $\{G_i\}_{i=1}^{N_g}$ associating to N_g maximin data points, respectively.

Step 3: *Randomly select data near each maximin data point to obtain n data points.* The final data $\mathcal{X}_s$ of size *n* is built by selecting random data points from each group $G_i (i = 1, \dots, N_g)$. The number of data points n_i collected from G_i is proportional to the number of data points in G_i, i.e., $n_i = \lceil n \times | G_i | / N \rceil$.

P-Stable Distributions Based LSH

Stable distributions are defined as limits of normalized sums of independent identically distributed variables, such as Gaussian distribution. Following Datar et al. (2004), a distribution $\mathcal{D}$ over R is called *p*-stable, if the random variable $\sum_i x_i a_i$ has the same distribution as the variable $(\sum_i |x_i|^p)^{1/p} a$ (i.e., $|| \mathbf{x} ||_p \ a$), where $p \geq 0$, $\{x_1, \dots, x_d\}$ are *d* real numbers, $\{a_1, \dots, a_d\}$ are i.i.d variables with distribution $\mathcal{D}$, and a is a random variable with distribution $\mathcal{D}$.

Let $\boldsymbol{a}$ of dimension *d* be $(a_1, \dots, a_d)$, $\mathbf{x}$ of dimension *d* be $(x_1, \dots, x_d)$, a small collection of the dot product $(\boldsymbol{a} \cdot \mathbf{x})$ corresponding to different $\boldsymbol{a}$'s can be used to estimate $|| \mathbf{x} ||_p$. When $\mathbf{x}$ is a linear composition $\mathbf{x}_1 - \mathbf{x}_2$, then a specific distance measure of $|| \mathbf{x}_1 - \mathbf{x}_2 ||_p$ can be analyzed in a projected space of $\boldsymbol{a} \cdot (\mathbf{x}_1 - \mathbf{x}_2)$. Since $(\boldsymbol{a} \cdot \mathbf{x})$ projects $\mathbf{x}$ to a real line, if we ``cho''' the real line into equi-width segments of appropriate size *r*, then the more close of $\mathbf{x}_1$ and $\mathbf{x}_2$ will have greater collision probability, i.e., $\boldsymbol{a} \cdot \mathbf{x}_1$ and $\boldsymbol{a} \cdot \mathbf{x}_2$ falling into the same segment. Therefore, we can define a group of locality-preserving hash functions $g(\mathbf{x}) = \{h_1(\mathbf{x}), \cdots, h_M(\mathbf{x})\}$ in which $h_i(\mathbf{x}) : \mathcal{R}^d \rightarrow \mathcal{N}$ is formulated by:

$$h_i(\mathbf{x}) = \left\lfloor \frac{\mathbf{a}_i \cdot \mathbf{x} + b_i}{r} \right\rfloor \tag{7}$$

For $i = 1, 2, \cdots, M$, random $\boldsymbol{a}_i$ is a *d-dimensional* vector with entries chosen independently from a *p*-stable distribution (e.g., $\mathcal{N}(0,1)$), and *b* is a real number chosen uniformly from the range $[0, r]$. Thus, $g(\mathbf{x})$ partitions the input space into buckets whose id is the concatenation of projected values by *M* locality-preserving hash functions. To increase the collision probability of neighbors and improve discriminative capability of hashing, a set of hash function families rather than one hash function family is frequently constructed (Aslani & Seipel, 2021).

THE PROPOSED IEPS METHOD

Edge Pattern Selection With Double Local Analysis

From Section 2.3 (Shrinkable Boundary Selection), the k_{e} nearest neighboring data points are critical factors for determining whether $\mathbf{x}_i$ is an edge pattern. In SBS, evaluating each data point involves one round of distance measurement with a time complexity of *O*(*N*) as no data point except $\mathbf{x}_i$ can be avoided. However, pdLSH has demonstrated that the closer of $\mathbf{x}_j$ and $\mathbf{x}_i$ will result in a higher collision probability of falling into the same segment if we ``cho''' the projected space into equi-width segments. This property makes it suitable for partitioning the data space using *k*-means++ which is known for creating *K* circle-like subspaces.

Figure 2*a* illustrates the conventional edge pattern selection strategy such as SBS, where the outermost dot-and-dash line represents the collected edge patterns. Since each edge pattern is determined through global analysis, it is equivalent to chopping the data space into single subspace with one center *C*. Consequently, every data point is projected into the same subspace and is considered as a candidate for the k_{e} nearest neighbors of $\mathbf{x}_i$. In fact, a large proportion of redundant computations happens since $N \gg k_{\mathrm{e}}$. After introducing data partition, SBS can focus on the specific subspace of $\mathbf{x}_i$. As depicted in Figure 2*b*, we perform data partitioning using *k*-means++ and set $K = 2$ for $N / K \gg k_{\mathrm{e}}$, then edge patterns can be separately collected based on the nearest neighbor analysis in two subspaces C_{11} and C_{12}. Here, blue dot-and-dash lines denote the collected edge patterns. Theoretically, $\overline{\mathbf{x}_3\mathbf{x}_4}$ is the border shared by C_{11} and C_{12}, while in practice, we observe edge patterns surrounding $\overline{\mathbf{x}_3\mathbf{x}_4}$ that may belong to either C_{11} and C_{12} due to the data partition. Therefore, edges of the two subspaces can be formulated by:

$$\begin{cases} \mathrm{Edge}(C_{11}) = \{\overparen{\mathbf{x}_1\mathbf{x}_3}, \overparen{\mathbf{x}_4\mathbf{x}_1},, \overline{\mathbf{x}_3\mathbf{x}_4}\} \\ \mathrm{Edge}(C_{12}) = \{\overparen{\mathbf{x}_3\mathbf{x}_6}, \overparen{\mathbf{x}_6\mathbf{x}_4}, \mathbf{x}_4\mathbf{x}_3\} \end{cases} \tag{8}$$

Similarly, when *K* is set to 5, we get five subspaces C_{21}, C_{22}, C_{23}, C_{24} and C_{25}. The corresponding edges can be expressed by red dotted lines, i.e.:

$$\begin{cases} \mathrm{Edge}(C_{21}) = \{\mathbf{x}_1\overset{\frown}{}\mathbf{x}_3, \overline{\mathbf{x}_1\mathbf{x}_2}, \mathbf{x}_2\mathbf{x}_3\}, \\ \mathrm{Edge}(C_{22}) = \{\overparen{\mathbf{x}_1\mathbf{x}_4}, \overline{\mathbf{x}_1\mathbf{x}_2}, \mathbf{x}_2\mathbf{x}_4\}, \\ \mathrm{Edge}(C_{23}) = \{\overline{\mathbf{x}_2\mathbf{x}_3}, \mathbf{x}_3\mathbf{x}_5, \mathbf{x}_5\mathbf{x}_4, \mathbf{x}_4\mathbf{x}_2\}, \\ \mathrm{Edge}(C_{24}) = \{\overparen{\mathbf{x}_3\mathbf{x}_6}, \overline{\mathbf{x}_6\mathbf{x}_5}, \mathbf{x}_5\mathbf{x}_3\}, \\ \mathrm{Edge}(C_{25}) = \{\overparen{\mathbf{x}_6\mathbf{x}_4}, \overline{\mathbf{x}_4\mathbf{x}_5}, \mathbf{x}_5\mathbf{x}_6\}. \end{cases} \tag{9}$$

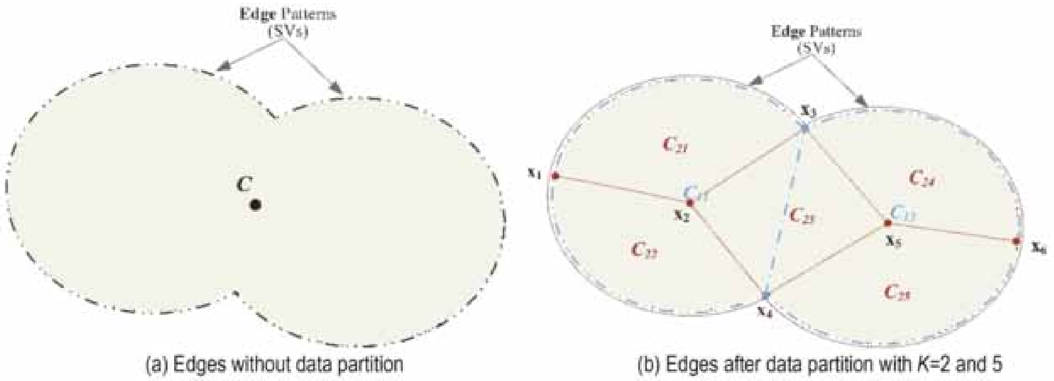
Apparently, the common elements of Eq.(8) and Eq.(9) are:

$$\mathrm{Edge}(C_{11} \cup C_{12}) \cap \mathrm{Edge}(C_{21} \cup C_{22} \cup C_{23} \cup C_{24} \cup C_{25}) = \{\overparen{\mathbf{x}_1\mathbf{x}_3}, \overparen{\mathbf{x}_4\mathbf{x}_1}, \overparen{\mathbf{x}_3\mathbf{x}_6}, \overparen{\mathbf{x}_6\mathbf{x}_4}\}, \tag{10}$$

which well coincides with edges collected without data partition in Figure 2*a*. Consequently, the proposed EPSDLA strategy is inspired by the observation and an intuitive expectation: *The collection of edge patterns in the whole data space can be alternatively done by extracting the common elements between edge patterns separately collected in two sets of subspaces partitioned by k-means++ (or any other data grouping/clustering method) with different K values.*

Based on the preceding analysis, the proposed EPSDLA strategy is succinctly described by Algorithm 1. Given two different cluster numbers K_1 and K_2, EPSDLA invokes *k*-means++ twice to partition $\mathcal{X}$ into two sets of subspaces, i.e., $X_1 = \{\mathcal{X}_{11}, \mathcal{X}_{12}, \cdots, \mathcal{X}_{K_1}\}$ and $X_2 = \{\mathcal{X}_{21}, \mathcal{X}_{22}, \cdots, \mathcal{X}_{K_2}\}$. Then, a double local analysis involving the typical SBS on all the subspaces of X_1 and X_2, denoted as SBS-SubSpace in lines 4-9, is employed to collect all the edge patterns $\bigcup_{i=1}^{K_1} \mathcal{X}_{\mathrm{e}1i}$ and $\bigcup_{j=1}^{K_2} \mathcal{X}_{\mathrm{e}2j}$. Finally, the expected edge patterns $\mathcal{X}_{\mathrm{e}}$ in global can be effortlessly extracted through an intersection operation in line 11. As all critical phases, except line 11 in Algorithm 1, involve double invocations (e.g., lines 2 and 3, 4-6, and 7-9), this strategy is termed a double edge pattern selection strategy. Notably, these phases can be executed independently to fulfill parallel processing requirements.

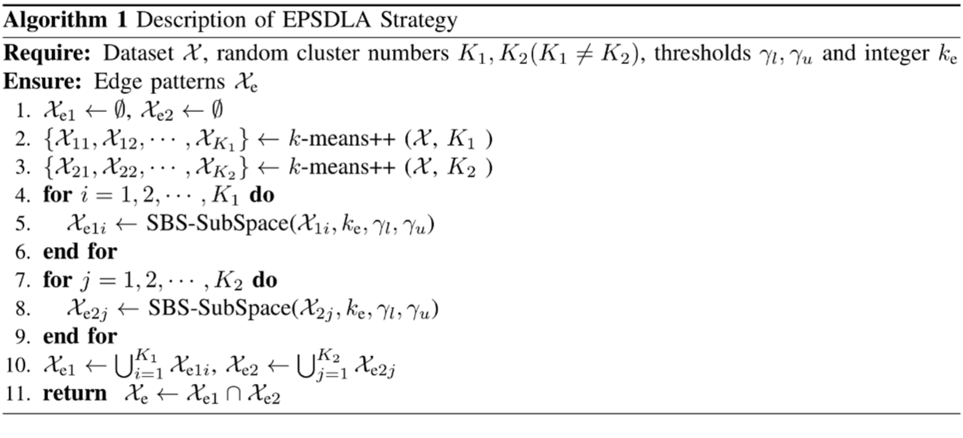
Figure 2. Principle of edge pattern selection without data partition and after data partition by k-means++ with different k values



(a) Edges without data partition

(b) Edges after data partition with K=2 and 5

Algorithm 1. Description of EPSDLA strategy



Algorithm 1 Description of EPSDLA Strategy

Require: Dataset $\mathcal{X}$, random cluster numbers $K_1, K_2 (K_1 \neq K_2)$, thresholds γ_l, γ_u and integer k_{e}

Ensure: Edge patterns $\mathcal{X}_{\mathrm{e}}$

```
1. $\mathcal{X}_{e1} \leftarrow \emptyset$, $\mathcal{X}_{e2} \leftarrow \emptyset$
2. $\{\mathcal{X}_{11}, \mathcal{X}_{12}, \cdots, \mathcal{X}_{K_1}\} \leftarrow k$-means++ ($\mathcal{X}$, $K_1$ )
3. $\{\mathcal{X}_{21}, \mathcal{X}_{22}, \cdots, \mathcal{X}_{K_2}\} \leftarrow k$-means++ ($\mathcal{X}$, $K_2$ )
4. for $i = 1, 2, \cdots, K_1$ do
5.     $\mathcal{X}_{e1i} \leftarrow$ SBS-SubSpace($\mathcal{X}_{1i}, k_e, \gamma_l, \gamma_u$)
6. end for
7. for $j = 1, 2, \cdots, K_2$ do
8.     $\mathcal{X}_{e2j} \leftarrow$ SBS-SubSpace($\mathcal{X}_{2j}, k_e, \gamma_l, \gamma_u$)
9. end for
10. $\mathcal{X}_{e1} \leftarrow \bigcup_{i=1}^{K_1} \mathcal{X}_{e1i}$, $\mathcal{X}_{e2} \leftarrow \bigcup_{j=1}^{K_2} \mathcal{X}_{e2j}$
11. return $\mathcal{X}_e \leftarrow \mathcal{X}_{e1} \cap \mathcal{X}_{e2}$
```

The complete description of SBS-SubSpace is outlined in Algorithm 2. In line with a consistent standard for data description, k_e is a global constant for edge pattern analysis across all the subspaces. To expedite the model construction of SVDD, the norm vector $\mathbf{n}_i$ for each selected edge pattern $\mathbf{x}_i$ can also be extracted with a minor adjustment to Algorithm 2.

Data Adaptive Sampling Strategy for Data Reduction

Utilizing all the N_e edge patterns as input, Y. Ping et al. (2019) confirmed that the final number of the selected SVs N_sv by solving Eq.(1) is actually smaller than or equal to N_e. This indicates that the set of edge patterns forms a superset of SVs, and not all collected edge patterns contribute equally to data description in terms of maintaining data separability while avoiding overfitting. Hence, data reduction under an appropriate sampling strategy, such as MMRS (Rathore et al., 2019), is valuable for efficiency.

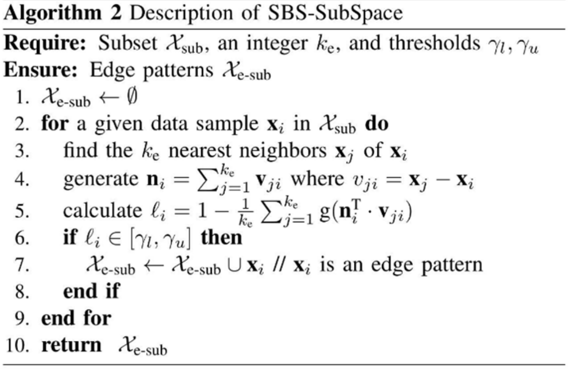
This is particularly crucial when the computational complexity of traditional edge pattern selection and dual problem solver is up to $O(N^2)$ and $O(N^3)$, respectively. However, retaining too few data points may diminish cluster density, thereby impacting data separability. Additionally, the high-dimensional approximate nearest neighbor search work by Gao and Long (2023) indicates that a high-dimensional data space requires more data points than a low-dimensional space to ensure comparable separability. Consequently, before conducting EPSDLA or data description, the decisions of whether to introduce a data sample strategy (i.e., feasibility) and how to set an appropriate sampling ratio (i.e., parameter ρ) are crucial issues. Unfortunately, these considerations have not received sufficient attention in the literature.

Before presenting DASS for data reduction, we first give some definitions following Aslani and Seipel (2021), based on pdLSH in which *L* hash function families and each family with *M* hash functions is considered.

Definition 1 (Similarity Index): For any two data points $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\mathcal{X}$, the similarity index $\mathrm{SI}(\mathbf{x}_i, \mathbf{x}_j)$ is defined as the number of common buckets between them in all L hash function families.

Inherently, SVDD employs SVs as prototypes for both data description and cluster analysis. Inspired by prototype learning (Zhang et al., 2022) and the representative instance selection method in Guo et al. (2014), we further define a representativeness index (RI) to differentiate the clustering effect around a given data point.

Algorithm 2. Description of SBS-subspace



```
Algorithm 2 Description of SBS-SubSpace
Require: Subset $\mathcal{X}_\mathrm{sub}$, an integer $k_\mathrm{e}$, and thresholds $\gamma_l, \gamma_u$
Ensure: Edge patterns $\mathcal{X}_\text{e-sub}$
$\mathcal{X}_\text{e-sub} \leftarrow \emptyset$
for a given data sample $\mathbf{x}_i$ in $\mathcal{X}_\mathrm{sub}$ do
    find the $k_\mathrm{e}$ nearest neighbors $\mathbf{x}_j$ of $\mathbf{x}_i$
    generate $\mathbf{n}_i = \sum_{j=1}^{k_\mathrm{e}} \mathbf{v}_{ji}$ where $v_{ji} = \mathbf{x}_j - \mathbf{x}_i$
    calculate $\ell_i = 1 - \frac{1}{k_\mathrm{e}} \sum_{j=1}^{k_\mathrm{e}} \mathrm{g}(\mathbf{n}_i^\mathrm{T} \cdot \mathbf{v}_{ji})$
    if $\ell_i \in [\gamma_l, \gamma_u]$ then
        $\mathcal{X}_\text{e-sub} \leftarrow \mathcal{X}_\text{e-sub} \cup \mathbf{x}_i$ // $\mathbf{x}_i$ is an edge pattern
    end if
end for
return $\mathcal{X}_\text{e-sub}$
```

Definition 2 (Representativeness Index): For a given data point $\mathbf{x}_i$, its representativeness index $\mathrm{RI}(\mathbf{x}_i)$ is defined as the number of data points $\mathbf{x}_j$ around it having $\mathrm{SI}(\mathbf{x}_i, \mathbf{x}_j) \geq 1$ for $j = 1, \cdots, N$ and $j \neq i$, i.e.:

$$\mathrm{RI}(\mathbf{x}_i) = \sum_{j=1}^{N} g(\mathrm{SI}(\mathbf{x}_i, \mathbf{x}_j) - 1) \tag{11}$$

Here, $g(x)$ is the function employed by the third phase of SBS, which returns 1 if $x \geq 0$. Clearly, when more data points within a cluster yield higher RI values, it implies that more data points are projected into the same bucket, indicating greater cluster compactness. Intuitively, the corresponding data set possesses better separability. Thus, we provide an intuitive definition of separability index (SPI) for reference analysis.

Definition 3 (Separability Index): For a data set $\mathcal{X}$, its separability index $\mathrm{SPI}(\mathcal{X})$ is defined as the two-tuples of the mean and variance (SPI_Mean, SPI_Var) over all the data points' RI values, i.e.:

$$\mathrm{SPI\_Mean} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{RI}(\mathbf{x}_\mathrm{i}) \tag{12}$$

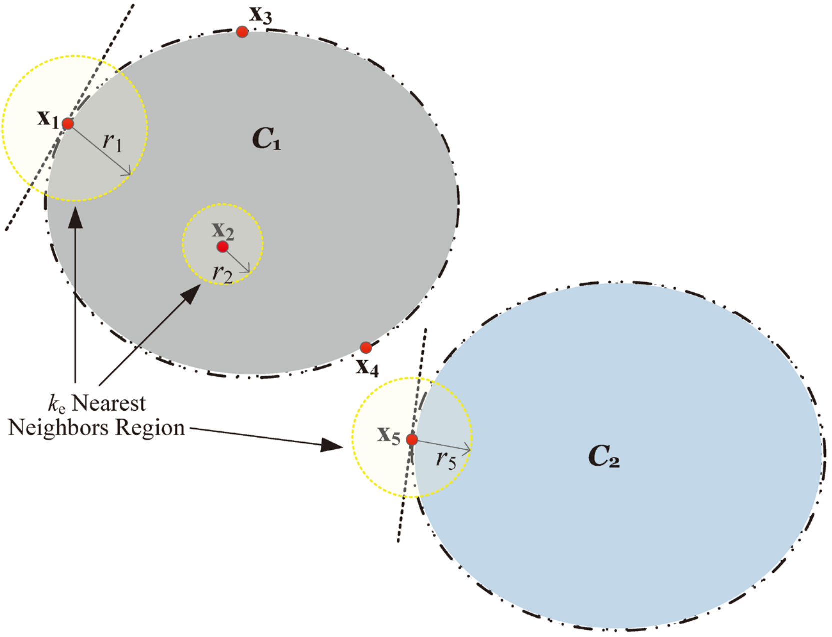
$$\mathrm{SPI\_Var} = \frac{1}{N} \sum_{i=1}^{N} [\mathrm{RI}(\mathbf{x}_i) - \mathrm{SPI\_Mean}]^2 \tag{13}$$

A higher SPI_Mean generally indicates better data separability, while a smaller SPI_Var suggests more balanced data distribution. We anticipate a notable SPI value for a dataset, whether it is a subset after data sampling or not. DASS should avoid $\mathcal{X}$'s SPI being cut down too much because data sampling frequently leads to data sparsification, reducing the possibility of neighboring data points being mapped into the same bucket. Therefore, optimal data separability is characterized not only by a relatively high SPI value but also by a substantial proportion of data points exhibiting high RI values.

To assess the adequacy of a dataset for SVDD, we illustrate the bound analysis of SPI using Fig.3 as an example. Fig.3 consists of two clusters, C_1 and C_2, without any overlap. Inherited from SBS, EPSDLA checks each data point's k_e nearest neighbor region. However, due to imbalanced distribution, data points at different locations often have varying neighborhood sizes, even though k_e is the same. For instance, the k_e nearest neighbor regions of $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_5$ have different radii r_1, r_2, and r_5, respectively. Generally, r_2 is greater than r_1 because $\mathbf{x}_2$ is located in the interior region of C_1, which typically has a higher density than the locations of edge patterns.

Therefore, in theory, we have $RI(\mathbf{x}_2) \geq RI(\mathbf{x}_1)$. From a dynamical system (3) perspective, every data point should converge to the center of the cluster or the decomposed convex hull to which it belongs (Y. Ping et al., 2012; Lee & Lee, 2006; Li & Maguire, 2011). When checking whether $\mathbf{x}_1$ is an edge pattern, it is intuitively not expected that a distant data point $\mathbf{x}_3$ be considered because it could exacerbate the instability in generating the convergence direction for $\mathbf{x}_1$. Similarly, data point

Figure 3. Bound analysis of the separability index



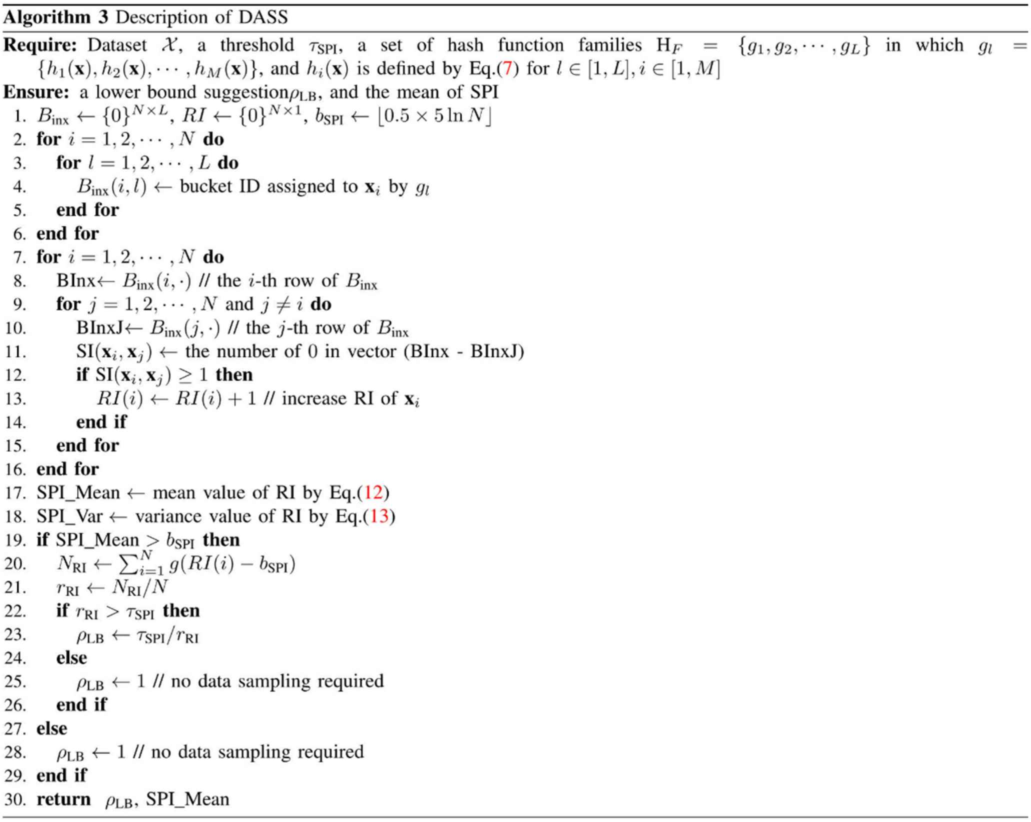
$\mathbf{x}_4$ from another cluster C_1 should be avoided when we check $\mathbf{x}_5$ since $\overrightarrow{\mathbf{x}_5\mathbf{x}_4}$ may cause an incorrect convergence direction of $\mathbf{x}_5$. By further considering the perspective of pdLSH, an ideal condition would be that all considered k_e nearest neighbors of $\mathbf{x}_i$ can be mapped into the same bucket. Unfortunately, this condition may be suitable for inner points but is not practical for edge patterns since an ideal edge pattern should have its k_e nearest neighbors located on the same side of the tangent plane, enlarging the bucket centered at $\mathbf{x}_i$ with the distance from the farthest data point as the radius. Therefore, we have $r_1 \geq r_2$ even if data points in the cluster C_i are uniformly distributed. Consequently, we suggest a lower bound of data separability (denoted by b_{SPI}) by

$$b_{\mathrm{SPI}} = \frac{1}{2} k_e \tag{14}$$

where $k_e \geq 5 \ln N$ is suggested as discussed by Li & Maguire (2011), and it can also be changed to any other value upon the prior knowledge related to the target data set. That means, in general, a data sampling strategy is suggested to be conducted only if SPI_Mean $>$ $\mathrm{b}_{\mathrm{SPI}}$.

Algorithm 3 presents the pseudocode for DASS. Utilizing pdLSH with L hash function families, each having M hash functions, lines 2-6 determine all L bucket IDs for every data point. For each data point $\mathbf{x}_i$, lines 7-16 first identify all the other data point $\mathbf{x}_j$ with $\mathrm{SI}(\mathbf{x}_i, \mathbf{x}_j) \geq 1$ by comparing their L bucket IDs. Once $\mathrm{SI}(\mathbf{x}_i, \mathbf{x}_j) \geq 1$, we increase the RI value of $\mathbf{x}_i$ to indicate that at least $\mathbf{x}_i$ can represent its neighbor $\mathbf{x}_j$ from a certain perspective. Consequently, lines 17-18 compute the SPI

Algorithm 3. Description of DASS



```
Algorithm 3 Description of DASS
Require: Dataset $\mathcal{X}$, a threshold $\tau_{\text{SPI}}$, a set of hash function families $\text{H}_F = \{g_1, g_2, \cdots, g_L\}$ in which $g_l = \{h_1(\mathbf{x}), h_2(\mathbf{x}), \cdots, h_M(\mathbf{x})\}$, and $h_i(\mathbf{x})$ is defined by Eq.(7) for $l \in [1, L], i \in [1, M]$
Ensure: a lower bound suggestion$\rho_{\text{LB}}$, and the mean of SPI
$B_{\text{inx}} \leftarrow \{0\}^{N \times L}$, $RI \leftarrow \{0\}^{N \times 1}$, $b_{\text{SPI}} \leftarrow \lfloor 0.5 \times 5 \ln N \rfloor$
for $i = 1, 2, \cdots, N$ do
    for $l = 1, 2, \cdots, L$ do
        $B_{\text{inx}}(i, l) \leftarrow$ bucket ID assigned to $\mathbf{x}_i$ by $g_l$
    end for
end for
for $i = 1, 2, \cdots, N$ do
    BInx$\leftarrow B_{\text{inx}}(i, \cdot)$ // the $i$-th row of $B_{\text{inx}}$
    for $j = 1, 2, \cdots, N$ and $j \neq i$ do
        BInxJ$\leftarrow B_{\text{inx}}(j, \cdot)$ // the $j$-th row of $B_{\text{inx}}$
        SI$(\mathbf{x}_i, \mathbf{x}_j) \leftarrow$ the number of 0 in vector (BInx - BInxJ)
        if SI$(\mathbf{x}_i, \mathbf{x}_j) \geq 1$ then
            $RI(i) \leftarrow RI(i) + 1$ // increase RI of $\mathbf{x}_i$
        end if
    end for
end for
SPI_Mean $\leftarrow$ mean value of RI by Eq.(12)
SPI_Var $\leftarrow$ variance value of RI by Eq.(13)
if SPI_Mean $> b_{\text{SPI}}$ then
    $N_{\text{RI}} \leftarrow \sum_{i=1}^{N} g(RI(i) - b_{\text{SPI}})$
    $r_{\text{RI}} \leftarrow N_{\text{RI}}/N$
    if $r_{\text{RI}} > \tau_{\text{SPI}}$ then
        $\rho_{\text{LB}} \leftarrow \tau_{\text{SPI}}/r_{\text{RI}}$
    else
        $\rho_{\text{LB}} \leftarrow 1$ // no data sampling required
    end if
else
    $\rho_{\text{LB}} \leftarrow 1$ // no data sampling required
end if
return $\rho_{\text{LB}}$, SPI_Mean
```

two-tuple following Eqs.(12-13). For the sake of simplicity, we suggest the lower bound ρ_{LB} of the sample rate formulated by:

$$\rho_{\text{LB}} = \tau_{\text{SPI}} / r_{\text{RI}} \tag{15}$$

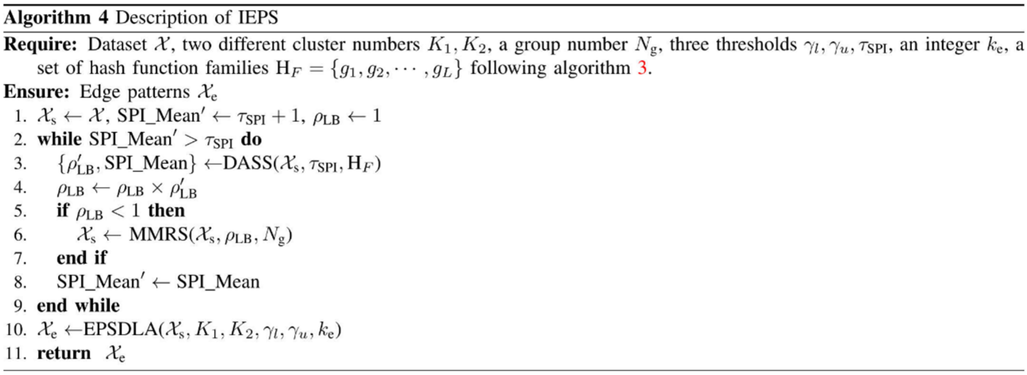
if SPI_Mean$\geq b_{\text{SPI}}$ and the proportion of data points with RI values greater than b_{SPI} (denoted by r_{RI}) surpasses a predefined threshold τ_{SPI}. If SPI_Mean is lower than b_{SPI}, it means that the number of data points in $\mathcal{X}$ is too small to form significant aggregation effect in the data space with high dimension *d*, and $\mathcal{X}$ has poor data separability. So, ρ_{LB} is set to 1 because the potential loss from data sampling likely outweighs the gain in this scenario.

Implementation of the Proposed IEPS

By integrating the aforementioned designs, Algorithm 4 presents the complete solution of the proposed IEPS.

Although *k*-means++ is recognized for its simplicity, it encounters challenges with large *K* or costly iterations. Additionally, not all edge patterns are essential for data description. Therefore, SPI_Mean' $\leftarrow \tau_{\text{SPI}} + 1$ of line 1 ensures the execution of the subsequent while loop. With a randomly

Algorithm 4. Description of IEPS



Algorithm 4 Description of IEPS

Require: Dataset $\mathcal{X}$, two different cluster numbers K_1, K_2, a group number N_g, three thresholds $\gamma_l, \gamma_u, \tau_\mathrm{SPI}$, an integer k_e, a set of hash function families $\mathrm{H}_F = \{g_1, g_2, \cdots, g_L\}$ following algorithm 3.

Ensure: Edge patterns $\mathcal{X}_\mathrm{e}$

```
1.  $\mathcal{X}_\mathrm{s} \leftarrow \mathcal{X}$, SPI_Mean$' \leftarrow \tau_\mathrm{SPI} + 1$, $\rho_\mathrm{LB} \leftarrow 1$
2.  while SPI_Mean$' > \tau_\mathrm{SPI}$ do
3.      $\{\rho'_\mathrm{LB}$, SPI_Mean$\} \leftarrow$DASS$(\mathcal{X}_\mathrm{s}, \tau_\mathrm{SPI}, \mathrm{H}_F)$
4.      $\rho_\mathrm{LB} \leftarrow \rho_\mathrm{LB} \times \rho'_\mathrm{LB}$
5.      if $\rho_\mathrm{LB} < 1$ then
6.          $\mathcal{X}_\mathrm{s} \leftarrow$ MMRS$(\mathcal{X}_\mathrm{s}, \rho_\mathrm{LB}, N_\mathrm{g})$
7.      end if
8.      SPI_Mean$' \leftarrow$ SPI_Mean
9.  end while
10. $\mathcal{X}_\mathrm{e} \leftarrow$EPSDLA$(\mathcal{X}_\mathrm{s}, K_1, K_2, \gamma_l, \gamma_u, k_\mathrm{e})$
11. return $\mathcal{X}_\mathrm{e}$
```

generated H_F following Aslani and Seipel (2021), DASS is employed to suggest the minimum proportion $\rho_{\mathrm{LB}'}$ of data points that should be retained while ensuring data separability. If the updated $\rho_\mathrm{LB} < 1$, line 6 invokes MMRS to sample $\rho_\mathrm{LB} N$ data points upon available computational resources and an acceptable run-time. Notice that the while loop in lines 2-9 minimizes the data points EPSDLA requires for efficiency. Theoretically, a higher sample rate leads to better accuracy. Therefore, a small value can be added to $\rho_{\mathrm{LB}'}$ in line 6 to make it closer to 1. Finally, EPSDLA is conducted for the global edge pattern selection in line 10. The preceding tasks contribute to the efficiency of IEPS from two aspects: (1) Using fewer data points as input $\left(N_\mathrm{s} \leq N\right)$ supports selecting indispensable rather than an excessive number; (2) Since the primary goal of *k*-means++ is data grouping, the optimization of the objective function (5) is not critical. Therefore, the iteration number can be a small integer, such as 10.

PERFORMANCE ANALYSIS

Time Complexity

As outlined in Algorithm 4, the proposed IEPS comprises three essential tasks: DASS, MMRS, and EPSDLA.

The first task, DASS, assesses an appropriate sample rate ρ_LB while ensuring a requisite data separability. From Algorithm 3, two time-consumption phases in DASS are related to lines 2-6 and lines 7-16. Since they respectively consume $O(NL)$ and $O(N^2)$ for simple computations, we record its time complexity as $O(N^2)$. As discussed by Rathore et al. (2019), the second critical task MMRS requires $O(dNN_\mathrm{g})$ to divide $\mathcal{X}$ into N_g groups and uniformly extract a subset $\mathcal{X}_\mathrm{s}$ containing N_s data points for the subsequent EPSDLA. Even though the double-round works are employed, in EPSDLA, *k*-means++ costs $O(dKN_\mathrm{s})$ to divide $\mathcal{X}_\mathrm{s}$ into K clusters, and K-round invocations of SBS-SubSpace thus take $O(N_\mathrm{s}^2/K)$ in average (i.e., $O(K(N_\mathrm{s}/K)^2)$) to collect the global edge patterns. Here, we use K to replace K_1 and K_2 for generality.

Although the while loop encompasses DASS and MMRS, their time complexities drop dramatically as the number of loops increases, and the loop count is extremely limited. The most time-consuming work is performed in the first round. Therefore, the time complexity of IEPS becomes $O(N^2+dNN_\mathrm{g}+ dKN_\mathrm{s}+N_\mathrm{s}^2/K)$ in which $O(dNN_\mathrm{g})$ exists only if $\rho_\mathrm{LB} < 1$. Although the most time-consuming part DASS reaches $O(N^2)$, which may seem comparable to conducting SBS on $\mathcal{X}$, the elementary subtraction operation between two vectors of length L has a significant advantage over the ranking work for K nearest neighbors in SBS. Experimental results in the following sections will show the evidence.

Datasets and Experimental Settings

Derived from BEPS (Li & Maguire, 2011) and SBS (Y. Ping et al., 2019), the proposed IEPS incorporates three strategies that correspond to the designs of EPSDLA and DASS and the introduction of MMRS for efficiency improvement while ensuring the concerned ability of SVDD. To comprehensively evaluate the performance of IEPS, four series of experiments on various datasets are conducted as follows:

(1) Check the validity of EPSDLA in terms of accuracy (contact ratio of edge patterns) and efficiency. In addition, the selection of K_1 and K_2 will be discussed.
(2) Verify whether SPI_Mean and SPI_Var of a data set $\mathcal{X}$ can well reflect the state of data distribution from separability and balance degree.
(3) Evaluate whether the suggested lower bound ρ_{LB} by DASS is effective when MMRS sample data points following that suggestion.
(4) Conduct feasibility analysis of IEPS by integrating it with two typical applications of SVDD, i.e., FSSVC and IBSVC, in terms of clustering accuracy on several real-world data sets with different sizes and dimensions.

The aforementioned experiments will be conducted on typical data sets from various domains. To make the analysis more intuitive and visible, the first three series of experiments use two synthetic datasets DS3 and DS4 (Karypis et al., 1999) with noise eliminated by T. Ping et al. (2012). These two-dimensional data sets have 8,543 and 7,670 data points, respectively. For intuitive comparisons, the fourth series of experiments employs five data sets from Y. Ping et al. (2022), which are listed in Table 1. Here, the breast cancer dataset wisconsin and shuttle data are provided by UCI repository (Frank & Asuncion, 2010). 20Newsgroups is a widely used text corpora from Lang (1995) and processed by Y. Ping et al. (2019) following the method of DC$_{\mathrm{GLI}}$-CCE. UNIBS Anonymized 2009 Internet Traces UNIBS-AIT (UNIBS 2010) consists of 9209 flows in four imbalance distributed categories, i.e., WEB (HTTP and HTTPS), MAIL (POP2, IMAP as well as their encrypted flows), BitTorrent, and eMule. Following the work of Guo et al. (2014), kddcup99 is a nine-dimensional data set extracted from KDD Cup 1999 Data,[1] which was used to build a network intrusion detector.

To evaluate the accuracy of clustering, we adopt the widely used similarity metrics adjusted rand index (ARI) (R. Xu, et al., 2008) formulated by Eq.(16). In Eq.(16), N_{ij} is the number of data points with true label I but they are assigned by j, $N_{i\cdot}$ and $N_{\cdot j}$ are the number of data points with label i and j, respectively:

Table 1. Description of the benchmark data sets

Data Sets	Data set Description		
	Size	Dims	# of Classes
wisconsin	683	9	2
UNIBS-AIT	9209	4	4
20Newsgroups	13998	20	20
shuttle	43500	9	7
kddcup99	494021	9	5

$$\text{ARI} = \frac{\sum_{i,j}\binom{N_{ij}}{2} - \left[\sum_{i}\binom{N_{i\cdot}}{2}\sum_{j}\binom{N_{\cdot j}}{2}\right] / \binom{N}{2}}{\frac{1}{2}\left[\sum_{i}\binom{N_{i\cdot}}{2} + \sum_{j}\binom{N_{\cdot j}}{2}\right] - \left[\sum_{i}\binom{N_{i\cdot}}{2}\sum_{j}\binom{N_{\cdot j}}{2}\right] / \binom{N}{2}} \tag{16}$$

All the algorithms are implemented using MATLAB 2021b on a mobile workstation with Intel I9-9880H processor and 128GB DRAM, running Windows 10-X64. To ensure fair comparisons, we have opted not to maximize the efficiency improvements by introducing parallel programming even though the internal operations of the three critical tasks of IEPS can be easily parallelized with *parfor* or *parfeval* methods.

Datasets and Experimental Settings

Although edge patterns are suitable for data description, following the principle of SVDD and discussions in Section 3.2 (Data Adaptive Sampling Strategy for Data Reduction), not all edge patterns are indispensable. Therefore, establishing a ground truth set of edge patterns has no practical meaning for data description. To perform a validity analysis of EPSDLA, we define a contact ratio (CRatio) formulated by Eq.(17) to represent accuracy, with edge patterns collected by the recent SBS as a reference. In Eq.(17), $\mathcal{X}_{\text{e-EPSDLA}}$ and $\mathcal{X}_{\text{e-SBS}}$ denote sets of edge patterns separately collected by EPSDLA and SBS, and $|\cdot|$ counts the number of data points:

$$\text{Accuracy}_{\text{CRatio}} = \frac{|\mathcal{X}_{\text{e-EPSDLA}} \cap \mathcal{X}_{\text{e-SBS}}|}{|\mathcal{X}_{\text{e-EPSDLA}}|} \times 100\% \tag{17}$$

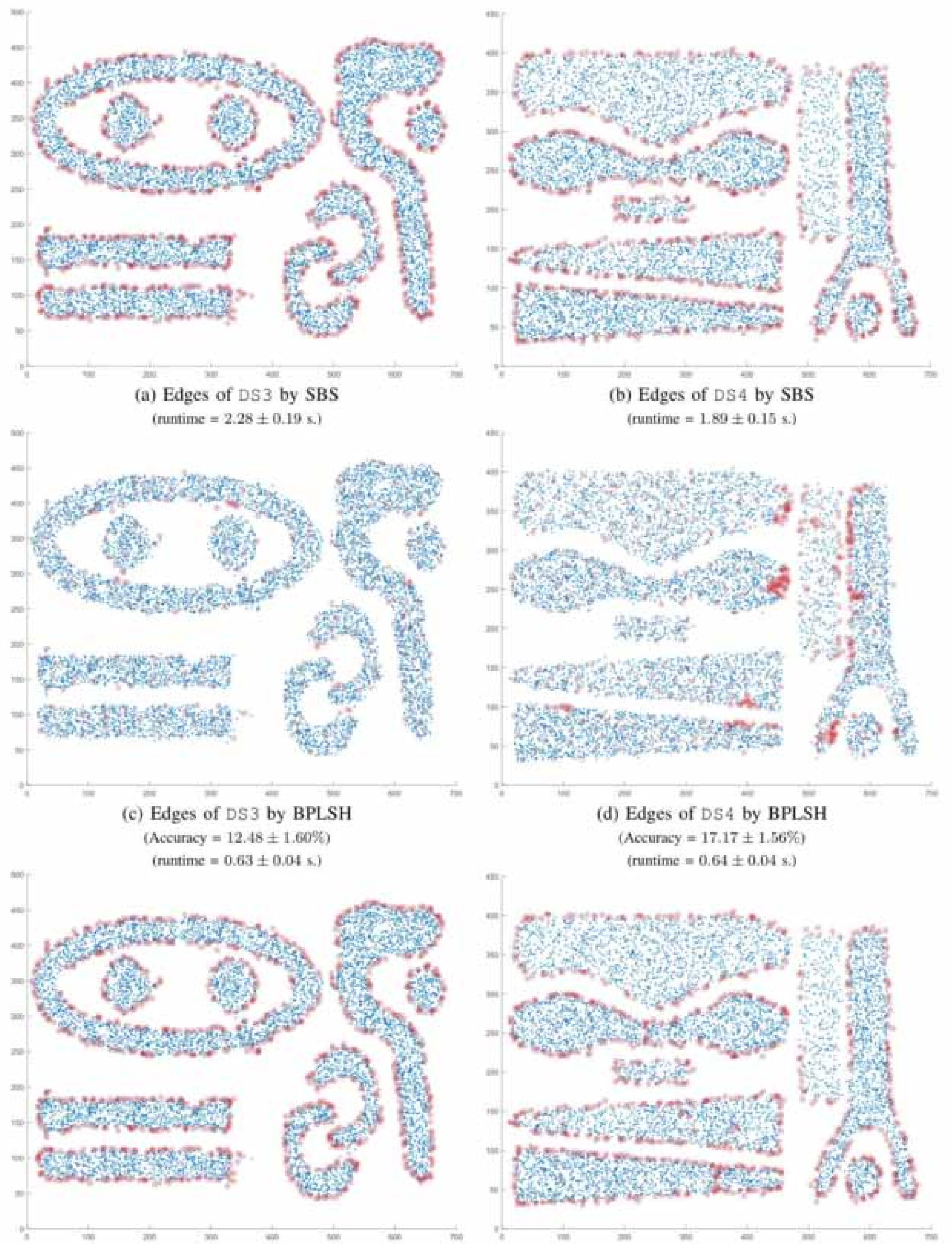
Figures 4a and 4b, respectively, depict edges of DS3 and DS4 collected by SBS, which is the equivalent to the classical BEPS due to $\gamma_u = 1$. In comparison with the ability-proven algorithm SBS, the most current boundary-aware instance selection algorithm BPLSH significantly improves efficiency but experiences a drastic drop in accuracy. In terms of CRatio after ten rounds of executions, only $12.48 \pm 1.60\%$ and $17.17 \pm 1.56\%$ data points collected are on the edges of DS3 and DS4, respectively. Despite the minimal impact on model training of SVM for supervised learning, these results cannot directly describe or discover the target data pattern in unsupervised learning due to the presence of too many inners. In contrast to BPLSH, the proposed EPSDLA has compromised runtime costs while achieving much-improved accuracy, which separately reach $98.73 \pm 0.54\%$ and $96.53 \pm 1.21\%$. Compared with SBS, the actual runtime costs have been reduced by about 32.89% and 32.80% on DS3 and DS4, respectively. Slight variances in accuracy and efficiency suggest EPSDLA's stability.

The previous experiments set K_1 and K_2 to 3 and 5, respectively. To further understand the performance related to the settings of K_1 and K_2, we change K_1 from 2 to 20 and let $K_2 = K_1 + \Delta K$ with ΔK ranging from 1 to 9. Experimental results are depicted by Figures 5 and 6 in which the bubble size is magnified 10 times to represent the variance of accuracy for visualization effect.

The center of the blue ellipse denotes the mean of all the accuracies with a fixed ΔK. Together with runtime costs shown in Figures 7a and 7b, several observations are as follows:

- As K_1 increases, the accuracy has a downward trend. However, the descent rate gradually becomes slower as ΔK increases. On DS3, the mean accuracy is reduced to 97.41% from 98.74% while a smaller interval between 97.39% and 98.05 is obtained on DS4.

Figure 4. Edge patterns with red circles selected by SBS (Ping et al. 2019), BPLSH (Aslani & Seipel, 2021), and EPSDLA with 10 round iterations of k-means++ under K_1=3, K_2=5. Both of them adopt the same parameters k_e = 30, $\gamma_l = 0.85, \gamma_u = 1$. SBS with $\gamma_u = 1$ means no edge shrinked that performs the same with BEPS (Li & Maguire, 2011).



(a) Edges of DS3 by SBS
(runtime = 2.28 ± 0.19 s.)

(b) Edges of DS4 by SBS
(runtime = 1.89 ± 0.15 s.)

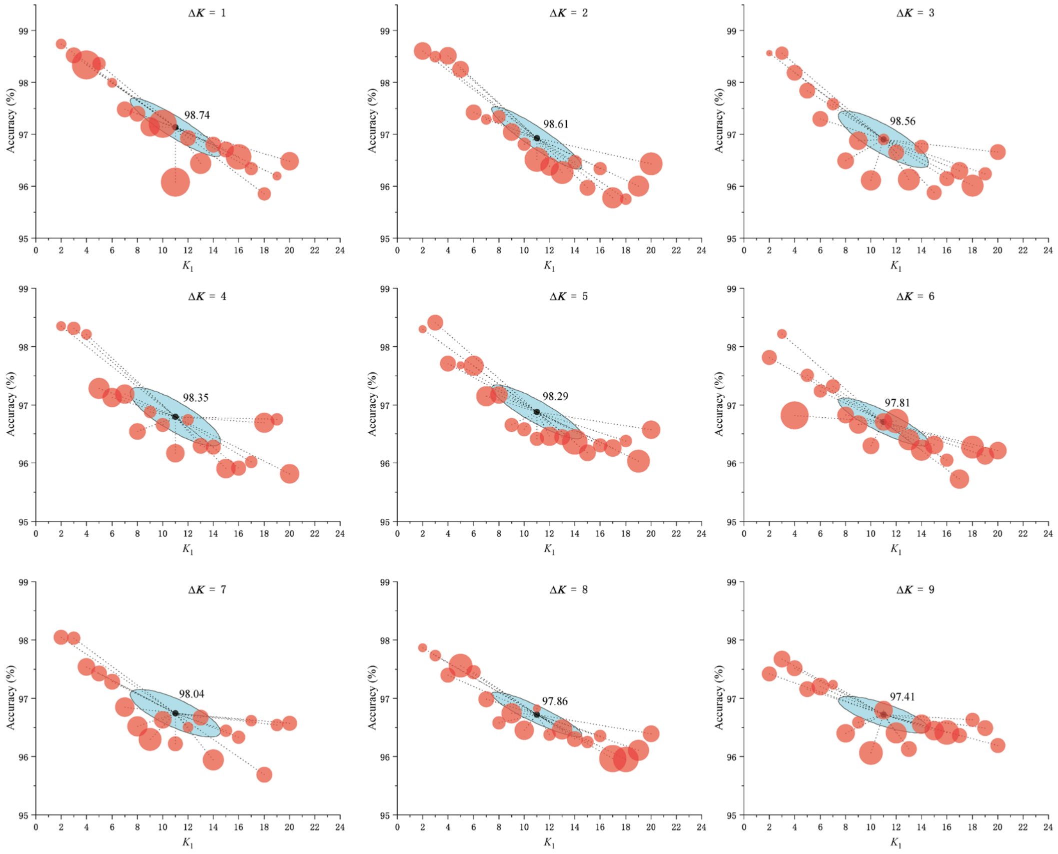
(c) Edges of DS3 by BPLSH
(Accuracy = 12.48 ± 1.60%)
(runtime = 0.63 ± 0.04 s.)

(d) Edges of DS4 by BPLSH
(Accuracy = 17.17 ± 1.56%)
(runtime = 0.64 ± 0.04 s.)

(e) Edges of DS3 by EPSDLA
(Accuracy = 98.73 ± 0.54%)
(runtime = 1.53 ± 0.04 s.)

(f) Edges of DS4 by EPSDLA
(Accuracy = 96.53 ± 1.21%)
(runtime = 1.27 ± 0.04 s.)

Figure 5. Accuracy obtained on DS3 in terms of CRatio by EPSDLA with different K_1 and $K_2 = K_1 + \Delta K$. The remaining parameters are $k_e = 30, \gamma_l = 0.85, \gamma_u = 1$.



- Variances of accuracies are relatively stable along with the increase of either K_1 or ΔK. That means the instability of *k*-means++ has a limited impact on the collection of edge patterns.
- Benefited by the low complexity of *k*-means++, more subspaces divided from the original data space significantly reduce the runtime cost as K_1 increases. However, ΔK has a limited impact on efficiency.
- As an extension of the prior observation, we have added an experiment to check the relationship between the iterations of best-effort optimization for Eq.(5) and the final accuracy. Figure 8 depicts the obtained accuracies concerning the increasing iteration number of *k*-means++. The solid line and shading region represent the mean and variance, respectively. Clearly, there is little impact on the accuracy of edge patterns collected by EPSDLA when the iteration number is greater than 15.

Based on these observations, choosing a smaller K_1 is essential to balance accuracy and efficiency. Since the cluster number is typically unknown in applications of SVDD, such as cluster discovery and description, an acceptable time-consumption should be the primary reference for selecting K_1. Furthermore, in the subsequent study, the maximal iteration number of *k*-means++ is limited to 10 for efficiency.

Figure 6. Accuracy obtained on DS4 in terms of CRatio by EPSDLA with different K_1 and $K_2 = K_1 + \Delta K$. The remaining parameters are $k_e = 30, \gamma_l = 0.85, \gamma_u = 1$.

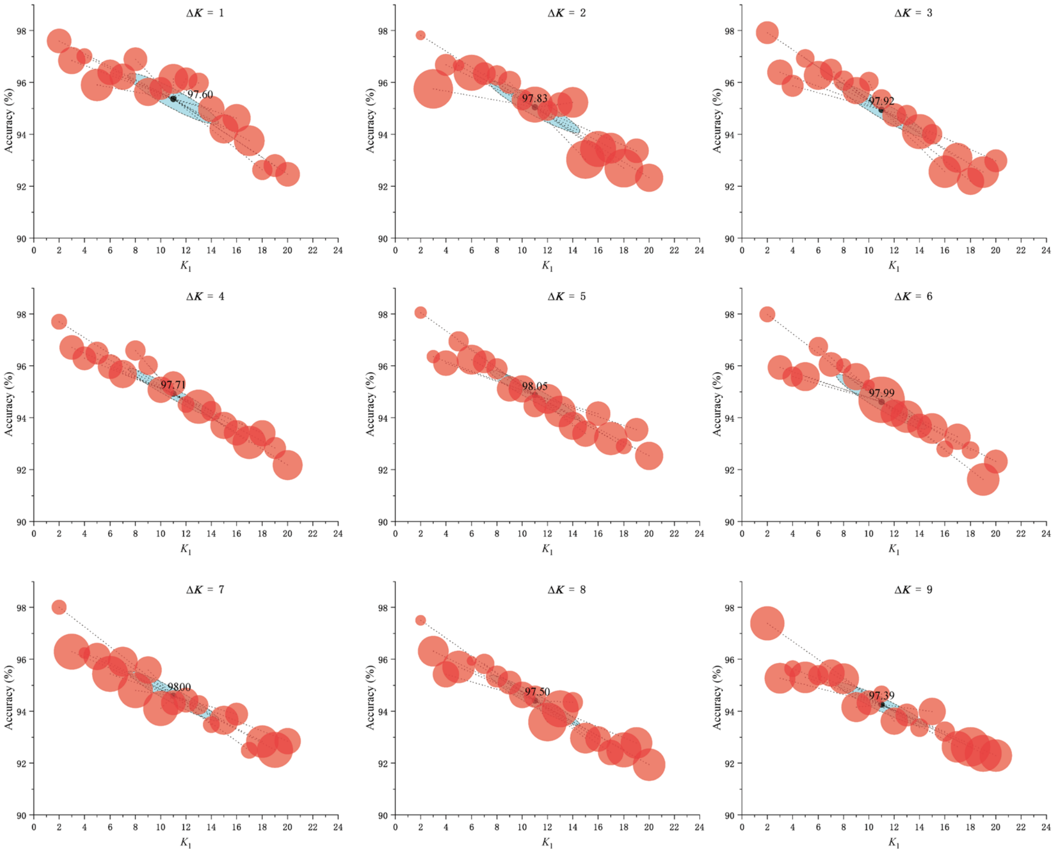
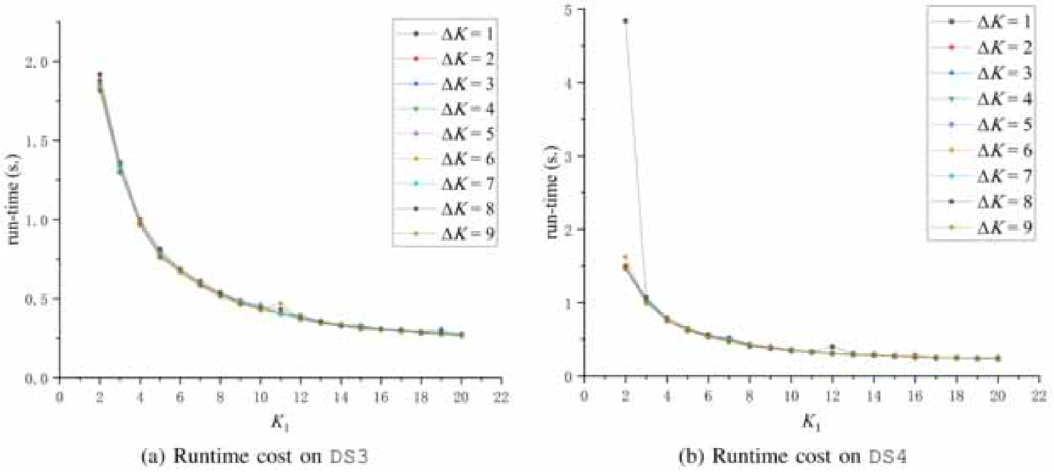


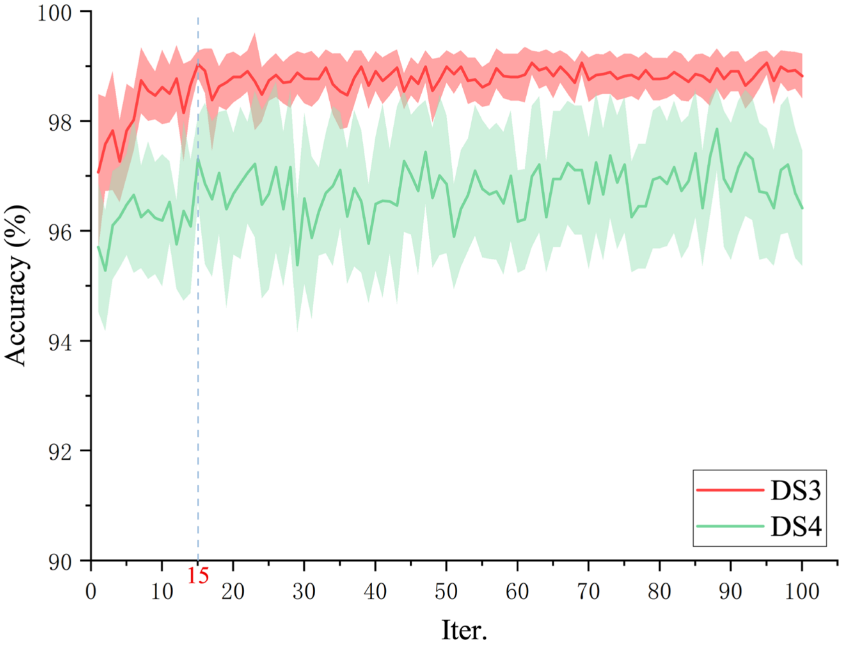
Figure 7. Runtime cost by EPSDLA with different K_1 and $K_2 = K_1 + \Delta K$ corresponding to experiments of Fig. 5 and Fig. 6.



(a) Runtime cost on DS3

(b) Runtime cost on DS4

Figure 8. Accuracies in terms of CRatio achieved by EPSDLA on DS3 and DS4 with iteration number controlled k-means++ and K_1=3, K_2 = 5, k_e = 30, $\gamma_l = 0.85, \gamma_u = 1$.



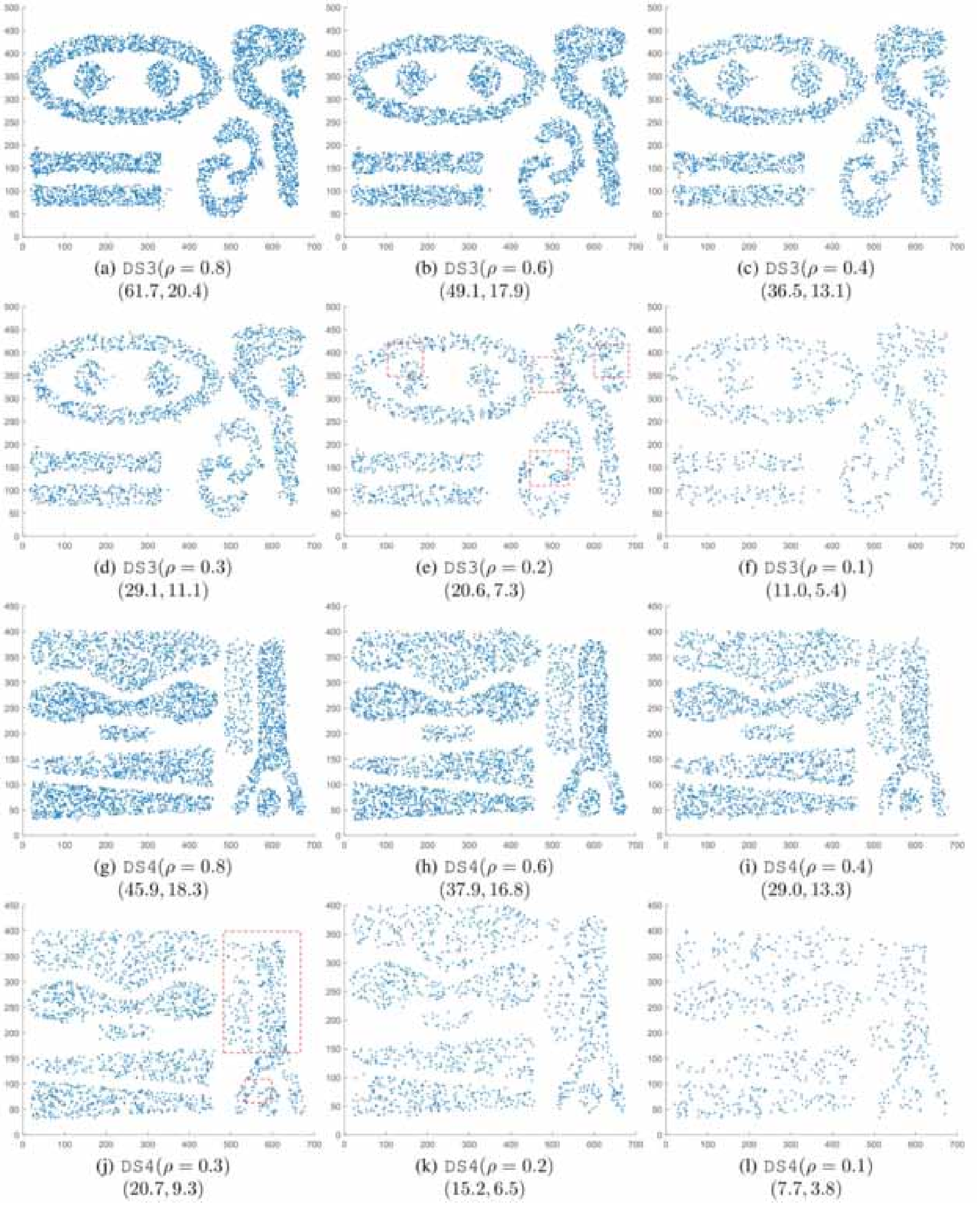
Relationship Between SPI and Data Separability

From the definition of clustering, a data set with good separability should exhibit large intra-cluster similarity and small inter-cluster similarity, making it possible to describe data patterns using fewer data points. Based on the discussions in Section 3.2 (Data Adaptive Sampling Strategy for Data Reduction), large intra-cluster similarity implies that data points in a cluster have a greater probability of being mapped into the same or close buckets by pdLSH than data points from different clusters. By fixing parameters M=100, L=30 and $r = 1$ throughout the proposed IEPS, as well as Algorithm 3, Fig. 9 presents the SPI analysis on data sets sampled from DS3 and DS4 with different sample rates, in terms of graphical sampling results and quantized SPI two-tuples.

The SPI two-tuples of DS3 and DS4 are (70.3, 23.7) and (54.8, 21.2), respectively. Since 70.3 is much greater than $b_{\text{SPI}} = 22.6$ of DS3, DASS naturally deems that data sampling is feasible. A similar conclusion can be drawn when we deal with DS4 for 54.8 > 22.4. Two variances 23.7 and 21.2 indicate the existence of gaps that separate data groups (i.e., clusters) and aggravate imbalance. Despite the large intra-cluster similarity, the variance for the largest class with 2045 data points in DS3 is 13.5, whereas the value is 7.2 for the largest class with 1558 data points in DS4. Therefore, a small variance also implies less separability.

Following the introduction of MMRS, Figures 9a-9f and Figures 9g-9l, respectively, depict the sampling results on DS3 and DS4 by setting the sample rate ρ to 0.8, 0.6, 0.4, 0.3, 0.2, and 0.1 in sequence. Apparently, the SPI two-tuples exhibit an obvious downward trend as ρ decrease. For DS3 in Figure 9e, there are four positions enclosed by dashed boxes starting to have weak connection characteristics while the first part of the SPI two-tuples 20.6 is smaller than b_{SPI}. A similar situation can be observed in Figure 9j for DS4. Together with b_{SPI}, the proposed SPI is a good choice to indicate data separability.

Figure 9. SPI analysis on data sets sampled from DS3 and DS4 with different sample rates ρ by MMRS. The lower bounds of data separability following Eq.(14) are the same, i.e., 22. The SPI two-tuples of the original DS3 and DS4 depicted by Fig.4a and Fig.4b are (70.3, 23.7) and (54.8, 21.2), respectively.



(a) DS3($\rho = 0.8$)
(61.7, 20.4)

(b) DS3($\rho = 0.6$)
(49.1, 17.9)

(c) DS3($\rho = 0.4$)
(36.5, 13.1)

(d) DS3($\rho = 0.3$)
(29.1, 11.1)

(e) DS3($\rho = 0.2$)
(20.6, 7.3)

(f) DS3($\rho = 0.1$)
(11.0, 5.4)

(g) DS4($\rho = 0.8$)
(45.9, 18.3)

(h) DS4($\rho = 0.6$)
(37.9, 16.8)

(i) DS4($\rho = 0.4$)
(29.0, 13.3)

(j) DS4($\rho = 0.3$)
(20.7, 9.3)

(k) DS4($\rho = 0.2$)
(15.2, 6.5)

(l) DS4($\rho = 0.1$)
(7.7, 3.8)

Effectiveness Evaluation of DASS

Based on the analysis in Section 4.3 (Datasets and Experimental Settings), an ideal parameter setting for the sample rate ρ should avoid breaking data separability. Good separability requires a large number of representative data points with great RI values. In this section, we conduct effectiveness evaluations by setting the data independent parameter $\tau_{\text{SPI}} = 0.4$ for an expectation of at least 40% data points having $RI(\mathbf{x}) \geq b_{\text{SPI}}$ and observing whether DASS can give a usable suggestion of ρ_{LB}.

Table 2 illustrates the suggested ρ_{LB} and clustering results by *k*-means++ on each data set. Notice that *k*-means++ with DASS performs cluster analysis on the original data set using the extracted *K* centers from the sampled data following the suggested ρ_{LB}. Based on the suggested ρ_{LB}, the SPI two-tuples of the subset sampled from either DS3 or DS4 shows the preservation of data separability when we take Figure 9 as a reference. Compared with performances of *k*-means++ on DS3 and DS4, *k*-means++ with DASS achieves comparable results on their subsets. Additionally, as depicted by Figure 10, the runtime costs by IEPS on DS3 and DS4 can be drastically reduced to $0.26 \pm 0.10\,\text{s}$ and $0.23 \pm 0.10\,\text{s}$, respectively. However, the selected edge patterns guarantee the description ability for the original data patterns, confirming the effectiveness of DASS.

Clustering Performance With FSSVC and IBSVC

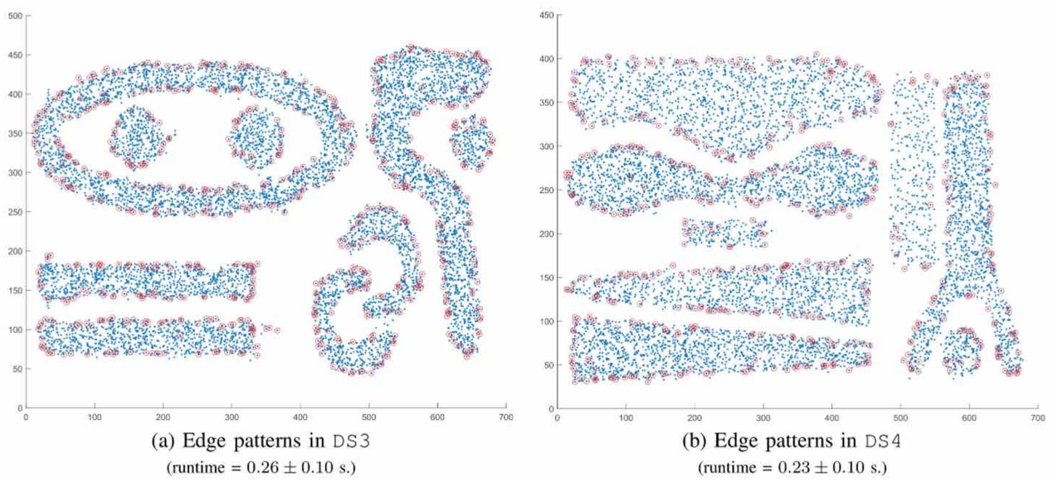
IEPS aims to effectively select a set of edge patterns, generally constituting a superset of SVs collected through model training of SVDD, such as FRSVC (Y. Ping et al., 2017) and VCC (Kim et al., 2015). However, as two typical variants, FRSVC and VCC cannot perform direct model training with edge patterns due to the absence of inner points which benefit convergence direction analysis. Among all the variants of SVDD, the fast and scalable SVC (FSSVC) (Y. Ping et al., 2015) and improved boundary SVC (IBSVC) (Y. Ping et al., 2022) are designed for direct model construction with edge patterns. Serving as the foundation for boundary SVC (BSVC) (Y. Ping et al., 2019), FSSVC exclusively conducts model training with edge patterns while IBSVC also uses the corresponding norm vectors (convergence directions) as an additional supplement for parameter adjustment. Since IEPS can readily provide the norm vector for each edge pattern by introducing an additional output in Algorithm 2, this section combines IEPS with FSSVC (denoted by IEPS-FSSVC) and IBSVC (denoted by IEPS-IBSVC) to evaluate the clustering performance on data sets in Table 1. Baselines models encompass VCC, FSSVC, FRSVC, BSVC, IBSVC, and the reformative SVC with elementary operations (RSVC-EO) (Y. Ping et al., 2020), which is a variant of FRSVC. Benchmark results are illustrated in Table 3, with the last row providing suggested ρ_{LB} values for each dataset.

Among the eight methods, VCC, IEPS-FSSVC, and IEPS-IBSVC have incorporated sampling strategies to enhance efficiency. Notably, two distinctions exist: (1) VCC adopts a random sampling strategy, while IEPS-FSSVC and IEPS-IBSVC prefer MMRS; (2) Whether to use the sampling strategy is mandatory and gratuitous for VCC, whereas IEPS-FSSVC and IEPS-IBSVC defer the judgment of DASS. Therefore, following the precedent set by Kim et al. (2015), in this study the sample rate for VCC is set to 0.01 for kddcup99 and 0.1 for the remaining datasets. However, DASS recommends no data sampling for wisconsion and 20Newsgroups due to their high dimensions coupled with small data sizes, with suggested ρ_{LB} of 0.45, 0.21, and 0.01 for UNIBS-AIT, shuttle, and kddcup99, respectively.

Table 2. Performance of k-means++ with/without DASS

Data Sets	*K*-Means++	*K*-Means++ With DASS	
	ARI	ρ_{LB}	ARI
DS3	0.4303 ± 0.0095	0.4002	0.4301 ± 0.0090
DS4	0.4357 ± 0.0207	0.4024	0.4308 ± 0.0209

Figure 10. Edge patterns selected by IEPS with 10 round iterations of k-means++ under K_1=3, K_2=5, and the other parameters are $k_e = 30, \gamma_l = 0.85, \gamma_u = 1$



(a) Edge patterns in DS3
(runtime = 0.26 ± 0.10 s.)

(b) Edge patterns in DS4
(runtime = 0.23 ± 0.10 s.)

Table 3. Benchmark results on five typical data sets

Method	Wisconsin			UNIBS-AIT			20Newsgroups			Shuttle			kddcup99		
	ARI	Time	N_c	ARI	Time	N_c	ARI	Time	N_c	ARI	Time	N_c	ARI	Time	N_c
VCC	0.8543	2.60	2	0.7455	7.94	5	0.4858	14.62	37	0.6096	**11.41**	14	0.7955	**175.96**	9
FRSVC	0.8798	0.66	2	0.8678	37.60	4	0.4927	145.81	26	0.8050	380.91	13	—	—	—
BSVC	*0.8963*	0.88	2	0.8565	8.61	4	0.4752	21.05	23	***0.8843***	108.55	7	***0.8677***	6191.20	8
RSVC-EO	0.8632	0.35	2	*0.8807*	9.24	4	**0.6084**	32.13	26	0.7337	343.46	9	0.7621	9489.38	5
FSSVC	**0.9248**	0.71	6	***0.8815***	***3.23***	4	0.3628	17.92	105	0.6857	86.81	33	—	—	—
IBSVC	0.8739	*0.31*	2	0.7482	**2.82**	5	0.5796	**4.23**	24	0.6929	***19.89***	8	**0.9120**	5500.93	12
IEPS-FSSVC	0.8909	***0.19***	2	**0.8818**	3.86	4	*0.5805*	***6.09***	29	0.7029	*53.87*	13	0.8009	*3533.13*	6
IEPS-IBSVC	***0.8965***	**0.15**	2	0.8370	*3.77*	4	***0.6009***	*7.48*	24	**0.8863**	56.77	6	*0.8049*	***3526.14***	4
	ρ_{LB} =1.00			ρ_{LB} = 0.45			ρ_{LB} = 1.00			ρ_{LB} = 0.21			ρ_{LB} = 0.01		

Note: 1. **Boldface** rank 1, ***Bold italic*** rank 2, *italic* rank 3; — means not available or more than 10,000 seconds.
2. VCC sets sample rate to 0.001 for kddcup99 and 0.1 for the others.
3. The maximum iter. Number for the solver of FRSVC, BSVC, RSVC-EO, IBSVC and IEPS-IBSVC are set to 3.

Regarding accuracy measured by ARI, each of FSSVC, IEPS-FSSVC, RSVC-EO, IEPS-IBSVC, and IBSVC reaches the first rank in one instance. BSVC, RSVC-EO, and IEPS-FSSVC achieve the top three ranks twice, while IEPS-IBSVC attains this position three times. When compared with FSSVC, IEPS-FSSVC excels on four data sets, with the exception of wisconsion. Meanwhile, IEPS-IBSVC outperforms IBSVC on four data sets except for kddcup99. To further validate the effectiveness of introducing IEPS, we conducted a typical nonparametric statistical test of Friedman test (Sheskin, 2003) by setting IEPS-IBSVC as the control method. Following Garcia and Herrera (2008), the average ranks and unadjusted p values are illustrated in Table 4. By introducing the Bergmann-Hommel procedure (Bergmann & Hommel, 1988), the adjusted p-value denoted by p_{Homm} corresponding to each pair comparison is also obtained. Obviously, IEPS-IBSVC, IEPS-FSSVC, and BSVC reaches

Table 4. Comparison results under non-parametric statistical test

Methods	Average Ranks	Unadjusted p	p_{Homm}
Control Method: IEPS-IBSVC, Average Rank = 2.80			
VCC	7.00	0.0067	0.0469
FSSVC	5.10	0.1376	0.5505
FRSVC	4.90	0.1752	0.6915
IBSVC	4.80	0.1967	0.6915
RSVC-EO	4.20	0.3662	0.6985
BSVC	3.80	0.5186	0.6985
IEPS-FSSVC	3.40	0.6985	0.6985

the top three performance. Furthermore, IEPS-IBSVC and IEPS-FSSVC respectively outperform IBSVC and FSSVC although they collect fewer edge patterns following the suggested ρ_{LB} by DASS. These results show the evidence of IEPS in collecting informative edge patterns. Additionally, strong evidence is found in the discovered cluster numbers N_{c} by IEPS-FSSVC and IEPS-IBSVC on each dataset, aligning closely with the actual numbers, as indicated in Table 1.

Besides VCC, in terms of efficiency, IBSVC, IEPS-FSSVC, and IEPS-IBSVC demonstrate significant advantages. For IBSVC, the main reason is the introduction of *k*-means, whereas both MMRS and EPSDLA contribute to IEPS-FSSVC and IEPS-IBSVC. As N increases, experimental results indicate that DASS becomes the most time-consuming task. For example, DASS consumes 40.15s and 3513.58s on shuttle and kddcup99, respectively. If we exclude the runtime costs incurred by DASS on shuttle and kddcup99, the remaining runtimes for IEPS-FSSVC and IEPS-IBSVC can be notably reduced to {13.72s, 19.55s} and {16.62s, 12.56s}, respectively.

Based on the aforementioned observations and analysis, it becomes evident that the proposed IEPS aligns well with the state-of-the-art variants of SVDD, considering both accuracy and efficiency comprehensively.

Application Discussion

From an unsupervised learning standpoint, edge patterns represent data points that are common across all clusters. As a superset of SVs, indispensable edge patterns can be one of the objectives for SVDD and an accelerator for SVC, thereby optimizing the efficacy of clustering analysis. As illustrated in Table 3, with IEPS, FSSVC and IBSVC perform better on works such as flow analysis (UNIBS-AIT), intrusion detection (kddcup99), and text clustering (20Newsgroups).

Practically, the utility of IEPS extends beyond unsupervised learning, positively impacting supervised learning as well. The concept of an edge as the boundary for a labeled data group implies that samples outside this boundary may represent novelties or abnormalities. Leveraging the disparity between edge patterns selected by IEPS on datasets with and without labels allows for the identification of risk regions, akin to the overlap depicted in Figure 1. Data points within these risk regions are frequently susceptible to attacks and are crucial for adversarial training (C. Chen et al. 2023), enhancing model robustness against various attacks, e.g., backdoor injection. In addition, fast selecting indispensable edge patterns also proves advantageous in uncovering nearest neighbors with different labels. As explored by H. Xu et al. (2023), these neighboring data points and labels play a role in adjusting the decision boundary and generating noise to help the target model forget specific data points for privacy protection. Even though numerous applications related to data description can be found, this paper focuses on a typical clustering analysis of SVC for clarity.

CONCLUSION

In unsupervised learning, SVDD shows us a good demonstration of data pattern discovery and description with insufficient information. However, the huge computational consumption poses a challenge for collecting SVs through model training in feature space or obtaining a superset of SVs by boundary analysis in input space. We find that the whole set of boundary patterns is unnecessary for data pattern description, and the global neighbor analysis for informative boundary creates too much redundant computation. For the former, not only border can be removed from boundary analysis, but also MMRS is employed for data reduction. Toward a reasonable sample rate, we define SI, RI, and SPI and design a data-adaptive sampling strategy DASS that assesses the SPI of the target data set and gives the lower bound of sample rate while keeping data separability. For the latter, EPSDLA is presented to select edge patterns with double local analysis and output of the global edge patterns after an intersection operation. By integrating DASS, MMRS, and EPSDLA, IEPS is proposed to select as few edge patterns as possible without affecting data description ability. Extensive experiments confirm the effectiveness of IEPS, as well as its adaptability to clustering methods.

Edge patterns and SVs are informative data points or representative samples for data description. In practice, despite that the cluster number of a data set is small, excessively high dimensions pose a challenge as they considerably decrease the probability of mapping neighboring data points into the same bucket. This not only impacts the effectiveness of data representation but also amplifies the time consumption of DASS. Although random projections provide an alternative way of dealing with the downspace of $\mathcal{X}$ through approximate preservation of distances in probability, they also exacerbate the instability. To mitigate this issue, a fast and stable dimensionality reduction strategy matching pdLSH to improve IEPS is a further appealing work.

AUTHOR NOTE

Data Availability: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare that there is no conflict of interest regarding the publication of this paper.

Funding Statement: This work is supported by the National Natural Science Foundation of China under Grant No. 62162009, the Scientific Program of Xuchang University under Grant No. 2023ZX045, the Scientific Research Innovation Team of Xuchang University under Grant No. 2022CXTD003, and Innovation Scientists and Technicians Troop Construction Projects of Henan Province under Grant No. CXTD2017099.

REFERENCES

Arslan, G., Madran, U., & Soyoğlu, D. (2022). An algebraic approach to clustering and classification with support vector machines. *Mathematics*, *10*(1), 128. doi:10.3390/math10010128

Arthur, D., & Vassilvitskii, S. (2007). *K*-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms (SODA '07).* ACM-SIAM. https://doi.org/doi:10.5555/1283383.1283494

Aslani, M., & Seipel, S. (2021). Efficient and decision boundary aware instance selection for support vector machines. *Information Sciences*, *577*(10), 579–598. doi:10.1016/j.ins.2021.07.015

Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. N. (2001). Support vector clustering. *Journal of Machine Learning Research*, *2*(Dec), 125–137. https://www.jmlr.org/papers/volume2/horn01a/horn01a.pdf

Bergmann, G., & Hommel, G. (1988). Improvements of general multiple test procedures for redundant systems of hypotheses. Multiple Hypotheses Testing, 70, 100-115. doi:10.1007/978-3-642-52307-6_8

Cevikalp, H., Yavuz, H. S., & Triggs, B. (2020). Face recognition based on videos by using convex hulls. *IEEE Transactions on Circuits and Systems for Video Technology*, *30*(12), 4481–4495. doi:10.1109/TCSVT.2019.2926165

Chen, C., Zhang, J., Xu, X., Lyu, L., Chen, C., Hu, T., & Chen, G. (2023). Decision boundary-aware data augmentation for adversarial training. *IEEE Transactions on Dependable and Secure Computing*, *20*(3), 1882–1894. doi:10.1109/TDSC.2022.3165889

Chen, M., Gao, W., Liu, G., Peng, K., & Wang, C. (2023). Boundary unlearning. In The IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE. doi:10.1109/CVPR52729.2023.00750

Datar, M., Immorlica, N., Indyk, P., & Mirrokni, V. S. (2004). Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on computational geometry (SCG '04).* ACM. doi:10.1145/997817.997857

Frank, A., & Asuncion, A. (2010). *UCI machine learning repository*. http://archive.ics.uci.edu/ml

Gao, J., & Long, C. (2023). High-dimensional approximate nearest neighbor search with reliable and efficient distance comparison operations. In *ACM SIGMOD/PODS international conference on management of data*. ACM. doi:10.1145/3589282

Garcia, S., & Herrera, F. (2008). An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, *9*(Dec), 2677–2694. https://www.jmlr.org/papers/volume9/garcia08a/garcia08a.pdf

Gornitz, N., Lima, L. A., Muller, K.-R., Kloft, M., & Nakajima, S. (2018). Support vector data descriptions and *k*-means clustering: One class? *IEEE Transactions on Neural Networks and Learning Systems*, *29*(9), 3994–4006. doi:10.1109/TNNLS.2017.2737941 PMID:28961127

Guo, C., Zhou, Y., Ping, Y., Zhang, Z., Liu, G., & Yang, Y. (2014). A distance sum-based hybrid method for intrusion detection. *Applied Intelligence*, *40*(1), 178–188. doi:10.1007/s10489-013-0452-6

Hu, W., Hu, T., Wei, Y., Lou, J., & Wang, S. (2023). Global plus local jointly regularized support vector data description for novelty detection. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(9), 6602–6614. doi:10.1109/TNNLS.2021.3129321 PMID:34851836

Jung, K.-H., Lee, D., & Lee, J. (2010). Fast support-based clustering method for large-scale problems. *Pattern Recognition*, *43*(5), 1975–1983. doi:10.1016/j.patcog.2009.12.010

Karypis, G., Han, E.-H., & Kumar, V. (1999). Chameleon: Hierarchical clustering algorithm using dynamic modeling. *Computer*, *32*(8), 68–75. doi:10.1109/2.781637

Kim, K., Son, Y., & Lee, J. (2015). Voronoi cell-based clustering using a kernel support. *IEEE Transactions on Knowledge and Data Engineering*, *27*(4), 1146–1156. doi:10.1109/TKDE.2014.2359662

Lang, K. (1995). Newsweeder: Learning to filter netnews. In *Proceedings of the 12th international conference of machine learning (ICML'95).* Morgan Kaufmann Publishers Inc. doi:10.1016/B978-1-55860-377-6.50048-7

Lee, J., & Lee, D. (2006). Dynamic characterization of cluster structures for robust and inductive support vector clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(11), 1869–1874. doi:10.1109/TPAMI.2006.225 PMID:17063691

Li, H., Ping, Y., Hao, B., Guo, C., & Liu, Y. (2022). Improved boundary support vector clustering with self-adaption support. *Electronics (Basel)*, *11*(2), 1854. doi:10.3390/electronics11121854

Li, Y., & Maguire, L. (2011). Selecting critical patterns based on local geometrical and statistical information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *33*(6), 1189–1201. doi:10.1109/TPAMI.2010.188 PMID:21493967

Ping, Y., Chang, Y., Zhou, Y., Tian, Y., Yang, Y., & Zhang, Z. (2015). Fast and scalable support vector clustering for large-scale data analysis. *Knowledge and Information Systems*, *43*(2), 281–310. doi:10.1007/s10115-013-0724-9

Ping, Y., Hao, B., Hei, X., Wu, J., & Wang, B. (2020). Maximized privacy-preserving outsourcing on support vector clustering. *Electronics, 9*(11), 1894:1-19. 10.3390/electronics9010178

Ping, Y., Hao, B., Li, H., Lai, Y., Guo, C., Ma, H., Wang, B., & Hei, X. (2019). Efficient training support vector clustering with appropriate boundary information. *IEEE Access : Practical Innovations, Open Solutions*, *7*(10), 146964–146978. doi:10.1109/ACCESS.2019.2945926

Ping, Y., Tian, Y., Guo, C., Wang, B., & Yang, Y. (2017). FRSVC: Towards making support vector clustering consume less. *Pattern Recognition*, *69*(9), 286–298. doi:10.1016/j.patcog.2017.04.025

Ping, Y., Tian, Y., Zhou, Y., & Yang, Y. (2012). Convex decomposition based cluster labeling method for support vector clustering. *Journal of Computer Science and Technology*, *27*(2), 428–442. doi:10.1007/s11390-012-1232-1

Rathore, P., Kumar, D., Bezdek, J. C., Rajasegarar, S., & Palaniswami, M. (2019). A rapid hybrid clustering algorithm for large volumes of high dimensional data. *IEEE Transactions on Knowledge and Data Engineering*, *31*(4), 641–654. doi:10.1109/TKDE.2018.2842191

Sheskin, D. (2007). *Handbook of parametric and nonparametric statistical procedures*. Chapman & Hall/CRC. doi:10.5555/1529939

Tax, D. M., & Duin, R. P. (1999). Support vector domain description. *Pattern Recognition Letters*, *20*(11-13), 1191–1199. doi:10.1016/S0167-8655(99)00087-2

UNIBS. (2009). *The unibs anonymized 2009 internet traces*. http://www.ing.unibs.it/ntw/tools/traces

Xu, H., Zhu, T., Zhang, L., Zhou, W., & Yu, P.S. (2023). Machine unlearning: A survey. *ACM Computing Surveys, 56*(1), 9:1-36. 10.1145/3603620

Xu, R., & Wunsch, D. C. (2008). *Clustering*. John Wiley & Sons. 10.1002/9780470382776

Zhang, X., Zhu, Z., Zhao, Y., & Zhao, Y. (2022). Prototype learning in machine learning: A literature review. *Ruan Jian Xue Bao. Journal of Software*, *33*(10), 3732–3753. https://doi.org/son10.13328/j.cnki.jos.006365

ENDNOTE

[1] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Huina Li received the BS degree in electronics and information engineering from Huazhong Normal University in 2003, and the MS degree in mathematics from He'nan University in 2008. She is an associate professor with the School of Information Engineering, Xuchang University. Her research interests include machine learning, data mining, and signal processing.

Yuan Ping received the BS degree in electronics and information engineering from Southwest Normal University in 2003, the MS degree in mathematics from He'nan University in 2008, and the PhD degree in information security from Beijing University of Posts and Telecommunications in 2012. He is a professor with the School of Information Engineering, Xuchang University. He was a visiting scholar with the School of Computing and Informatics, University of Louisiana at Lafayette and with the Department of Computing Science, University of Alberta. His research interests include machine learning, public key cryptography, data privacy and security, and cloud and edge computing.