Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Relative Probabilistic Density Weighted Granular-Ball Twin Support Vector Machine for Robust Classification

Yuxin Qiu
Jinan University
Peiai Zhang
tzhangpa@jnu.edu.cn
Jinan University
Duxin Deng
Jinan University
Xiaowei Cheng
Jinan University
Zhisheng Li
Jinan University

Research Article

Keywords: support vector machine, twin support vector machine, granular-ball computing, local density, relative probabilistic density, density-ratio estimation

Posted Date: October 1st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10545705/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A Relative Probabilistic Density Weighted Granular-Ball Twin Support Vector Machine for Robust Classification

Yuxin Qiu[1], Peiai Zhang[1*], Duxin Deng[1], Xiaowei Cheng[1], Zhisheng Li[2]

[1*] Department of Mathematics, College of Information Science and Technology, Jinan University, Guangzhou, 510632, Guangdong, China .
[2*] Guangdong Tobacco Qingyuan Co., Ltd., Qingyuan, Guangdong, China .

*Corresponding author(s). E-mail(s): tzhangpa@jnu.edu.cn;
Contributing authors: qiuyuxin4552@foxmail.com;

Abstract

Classical Twin Support Vector Machine (TWSVM) uses individual samples as input and constructs two nonparallel hyperplanes for binary classification. To improve computational efficiency, Granular-Ball Twin Support Vector Machine (GBTWSVM) replaces point-level input with granular-ball-level input. However, most existing granular-ball generation methods based on virtual centers may fail to accurately represent the original data distribution on large-scale and complex datasets. In addition, existing relative-density weighting methods mainly operate at the point level, whereas distribution-aware weighting at the granular-ball-center level has not been sufficiently investigated. To address these problems, this paper proposes a Twin Support Vector Machine based on Local Density Granular-Ball Generation and Relative Probabilistic Density weighting, termed RPDLDGBTWSVM. The proposed method selects real samples with relatively high local densities as granular-ball centers and determines granular-ball radii according to within-class compactness and between-class separability. A KLIEP-based relative probabilistic density estimation method is introduced at the granular-ball-center level. The resulting weights are incorporated into the slack-variable penalty terms, while the granular-ball radii are introduced into the classification constraints. Theoretical analysis shows that the approximation error between the granular-ball-center-level RPD-weighted loss and the corresponding point-level weighted loss is controlled by the granular-ball radii. The

resulting subproblems remain convex and feasible. The proposed model can also be extended to nonlinear classification through kernel methods. Comparative experiments are conducted on multiple UCI datasets and medical binary classification datasets. The experimental results show that the proposed method achieves competitive overall performance in terms of classification accuracy and computational efficiency.

Keywords: support vector machine, twin support vector machine, granular-ball computing, local density, relative probabilistic density, density-ratio estimation

1 Introduction

Support Vector Machine (SVM) is a classical classification method derived from statistical learning theory [1]. To reduce the computational complexity of SVM, Twin Support Vector Machine (TWSVM) constructs two nonparallel hyperplanes, each of which is close to one class and as far as possible from the other. In this way, one large-scale quadratic programming problem is transformed into two smaller optimization problems [2]. Subsequently, to further improve the efficiency and stability of TWSVM, models such as Least Squares TWSVM, Twin Bounded SVM, and Twin Parametric-Margin SVM modified the solution form of the optimization problems, regularization structures, and margin construction mechanisms [3–5]. Existing studies have shown that TWSVM has been applied to various machine learning tasks, including pattern classification, regression, and imbalanced learning [6].

To handle noisy samples, outliers, and uncertain boundary samples in real-world data, existing studies have improved TWSVM from the perspectives of loss function design, sample weighting, and structural information extraction. CTSVM introduces the bounded C-loss function to weaken the influence of outliers [7]. SLTSVM employs the symmetric LINEX loss function to reduce the effects of mislabeled samples and biased observations [8]. Average-depth-based TSVM characterizes sample centrality according to the geometric distribution of data, thereby improving the generalization ability of the model for peripheral samples and outliers [9]. Yuan et al. proposed a least-squares twin bounded support vector machine with a generalized correntropy-induced metric, termed CLSTBSVM, which replaces the squared L_2 distance with a generalized correntropy-induced metric to improve robustness to outliers [10]. Si et al. proposed a kernel-free quadratic surface twin support vector machine with a capped L_1-norm distance metric, termed CL_1QTSVM. This method constructs kernel-free quadratic surfaces and employs a bounded distance metric in the constraints to reduce the influence of outliers [11]. In addition, intuitionistic fuzzy information, dissimilarity measures, kernel-target alignment, and mass-based dissimilarity have also been employed to construct more refined sample weights [12–15]. These studies demonstrate that sample weighting is an effective strategy for improving the robustness of TWSVM. However, most of these methods still depend on local neighborhood information or specifically designed loss functions and may therefore be insufficient for reliably

handling high-dimensional sparse data, locally overlapping classes, and complex class boundaries.

Relative density provides another effective approach to sample weighting. Methods based on local geometric information and the K-nearest-neighbor rule can select representative training samples and reduce computational complexity [16]. RDSVM characterizes relative-density information according to the distances between samples and class centers, distinguishing support vectors from suspicious samples and thereby reducing the influence of noise on the classification boundary [17]. FSVM-RD combines relative density with fuzzy membership assignment to handle differences in sample contributions under imbalanced classification settings [18]. Relative Probabilistic Density (RPD) characterizes the relative consistency of a sample with different class distributions through the probability-density ratio between classes and has been applied to label-noise detection [19]. However, relative-density methods based on KNN or local distance measures tend to become unreliable in high-dimensional spaces [20]. In contrast, direct density-ratio estimation does not estimate the probability densities of the two classes separately. Instead, it directly estimates their ratio, thereby avoiding the accumulation of errors caused by separate density estimation [21]. KLIEP is a classical density-ratio estimation method that directly estimates importance weights by minimizing the Kullback–Leibler divergence [22]. Moreover, uLSIF, relative density-ratio estimation, and likelihood-ratio estimation based on convex risk demonstrate that density-ratio estimation can provide more stable information about distributional differences than local distance measures [23–26]. Nevertheless, most existing RPD- or density-ratio-based weighting methods still operate on point-level samples and have not sufficiently investigated weighting mechanisms under granular-ball representations.

To reduce computational complexity, Granular-Ball Computing (GBC) replaces point-level input with ball-level input. A granular ball is generally described by attributes such as its center, radius, and label. GBC employs granular balls rather than individual sample points as the basic computational units, thereby representing the original data with a smaller number of coarse-grained structures [27]. Granular-Ball Support Vector Machine (GBSVM) incorporates granular balls into the SVM framework and performs classification under constraints involving granular-ball centers and radii. It demonstrates that granular-ball input can improve model stability while reducing computational costs [28]. Granular-Ball Twin Support Vector Machine (GBTWSVM) further combines granular-ball representations with the two-hyperplane structure of TWSVM, thereby improving classification efficiency [29]. Granular-Ball Computing Based Fuzzy Twin Support Vector Machine (GBFTWSVM) introduces fuzzy membership and non-membership information at the granular-ball level to distinguish the contributions of different granular balls to the classification hyperplanes, thus improving model robustness [30]. Ganaie and Ahire proposed a Granular Ball Twin Support Vector Machine with Universum Data, termed GBU-TSVM, which incorporates Universum samples outside the target classes into the granular-ball model to refine the classification boundaries [31]. Grewal et al. proposed a Granular Ball Twin Bounded Support Vector Machine with Generalized Pinball Loss, termed GPin-GBTSVM. This method introduces generalized pinball loss and structural-risk

regularization into the granular-ball classifier, and its large-scale formulation avoids the explicit inversion of large matrices [32]. These studies demonstrate the considerable potential of combining granular-ball representations with TWSVM. However, existing granular-ball-based TWSVM methods still face two limitations. First, they generally depend on purity thresholds and k-means-based mean centers to generate granular balls, which may fail to accurately represent data with complex boundaries or sparse distributions. Second, granular-ball weights constructed from geometric distances or fuzzy functions may become unreliable for high-dimensional sparse data.

Local Density Granular-Ball Generation (LDGBG) can more effectively characterize the original data distribution and improve the quality of granular-ball generation. Unlike conventional granular balls constructed using mean centers, LDGBG selects real samples with relatively high local density as granular-ball centers and determines granular-ball radii according to within-class compactness and between-class separability [33]. For granular-ball-level weighting, compared with fuzzy weights constructed only from geometric distances between samples, RPD can characterize the relative importance of granular-ball centers from the perspective of the distributions of the two classes [19, 22]. Therefore, this paper combines LDGBG with RPD to simultaneously improve the quality of granular-ball representations and the effectiveness of granular-ball weighting. Accordingly, a Twin Support Vector Machine based on Local Density Granular-Ball Generation and Relative Probabilistic Density weighting, termed RPDLDGBTWSVM, is proposed.

The main contributions of this paper are summarized as follows.

1. A distribution-aware and efficient granular-ball classification framework is proposed for complex noisy real-world data. The framework integrates LDGBG, RPD, and TWSVM. Granular balls are used as the basic training units to reduce the classifier input size. Real samples with relatively high local densities are selected as granular-ball centers, reducing the deviation of conventional mean centers from the original data distribution.
2. An RPD-weighted granular-ball TWSVM model is developed. KLIEP directly estimates the relative probability density between positive- and negative-class granular-ball centers, and the estimated values are transformed into granular-ball weights. These weights are incorporated into the slack-variable penalties and the upper bounds of the dual variables. Granular-ball radii are introduced into the classification constraints. This design reduces the influence of boundary, anomalous, and class-overlapping granular balls on the classification hyperplanes. Linear and nonlinear RPDLDGBTWSVM models are established, together with their optimization problems, dual formulations, and decision functions. Theoretical analysis establishes the approximation relationship between the granular-ball-center-level weighted loss and the point-level weighted loss, and proves the convexity and feasibility of the resulting subproblems.
3. Experiments are conducted on 19 real-world datasets, including several large-scale UCI datasets covering medical diagnosis, credit classification, molecular classification, and decision-making tasks under noisy environments. The proposed model is evaluated in terms of classification accuracy, running time, noise robustness, granular-ball compression, parameter sensitivity, and statistical significance. The

results show that RPDLDGBTWSVM achieves superior and stable performance across different datasets and experimental settings.

2 Preliminaries

In this section, we briefly review Local Density Granular-Ball Generation (LDGBG), Relative Probabilistic Density (RPD), Twin Support Vector Machine (TWSVM), and Fuzzy Support Vector Machine (FSVM). These methods provide the granular-ball representation, distribution-aware weighting, and optimization basis for the proposed model.

2.1 Local Density Granular-Ball Generation

Let

$$D = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^d,\ y_i \in \{-1, +1\},\ i = 1, \ldots, n\} \tag{1}$$

be a binary classification dataset. A granular ball is represented as

$$GB_i = (\mathbf{c}_i, r_i, l_i), \tag{2}$$

where $\mathbf{c}_i$, r_i, and l_i denote its center, radius, and label, respectively.

Conventional granular-ball generation methods usually define the center as the mean of the samples contained in the granular ball. However, the mean center is a virtual point and may deviate from the original data distribution when the samples have a sparse or non-spherical structure. LDGBG selects real samples with relatively high local density as granular-ball centers [33].

Let d_c denote the cut-off distance of the dataset. The neighborhood radius is defined as

$$\delta = g d_c, \tag{3}$$

where $g > 0$ is the granularity coefficient. For a sample $\mathbf{x}_i$, its local density is calculated as

$$\rho_i = \sum_{\mathbf{x}_j \in \mathcal{N}_\delta(\mathbf{x}_i)} \exp\left[-\left(\frac{d_{ij}}{\delta}\right)^2\right] + N_\delta(\mathbf{x}_i), \tag{4}$$

where $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|_2$, $\mathcal{N}_\delta(\mathbf{x}_i)$ is the neighborhood of $\mathbf{x}_i$, and $N_\delta(\mathbf{x}_i)$ is the number of samples in this neighborhood.

Equation (4) considers both the number of neighboring samples and their compactness. Samples with larger local density are therefore selected as granular-ball centers.

To describe the sparsity of a dataset, LDGBG defines

$$\mathrm{Sparse}(D) = \frac{n}{d}. \tag{5}$$

For sparse datasets, the outlier detection condition is relaxed so that isolated but useful samples can be retained.

After obtaining the granular-ball centers, let $\mathbf{c}_j$ be the nearest center of $\mathbf{c}_i$. The radius corresponding to $\mathbf{c}_i$ is calculated as

$$r_i = \begin{cases} \dfrac{\|\mathbf{c}_i - \mathbf{c}_j\|_2}{2}, & l_i = l_j \text{ or } \rho_i > \rho_j, \\ \dfrac{\|\mathbf{c}_i - \mathbf{c}_j\|_2}{2} \dfrac{\rho_i}{\rho_i + \rho_j}, & l_i \neq l_j \text{ and } \rho_i \leq \rho_j. \end{cases} \tag{6}$$

This radius construction makes granular balls of the same class compact and increases the separation between granular balls of different classes. The resulting granular-ball set is denoted by

$$\mathcal{G} = \{(\mathbf{c}_i, r_i, l_i) \mid i = 1, \ldots, m\}, \qquad m < n. \tag{7}$$

2.2 Relative Probabilistic Density

Let $p_+(\mathbf{x})$ and $p_-(\mathbf{x})$ denote the class-conditional probability density functions of the positive and negative classes, respectively. For a positive-class sample, its relative probabilistic density is defined as

$$r_+(\mathbf{x}) = \frac{p_-(\mathbf{x})}{p_+(\mathbf{x})}, \tag{8}$$

while for a negative-class sample,

$$r_-(\mathbf{x}) = \frac{p_+(\mathbf{x})}{p_-(\mathbf{x})}. \tag{9}$$

A larger RPD value indicates that a sample is relatively more consistent with the opposite-class distribution. Such a sample may be located near the class boundary or may be an outlier or mislabeled sample [19].

Instead of estimating the two probability densities separately, KLIEP directly estimates the density ratio [21, 22]. Let

$$r(\mathbf{x}) = \frac{p_{\mathrm{nu}}(\mathbf{x})}{p_{\mathrm{de}}(\mathbf{x})} \tag{10}$$

be the target density ratio. KLIEP approximates it by

$$\widehat{r}(\mathbf{x}) = \sum_{q=1}^{b} \alpha_q \phi_q(\mathbf{x}), \qquad \alpha_q \geq 0, \tag{11}$$

where $\phi_q(\mathbf{x})$ is a nonnegative basis function.

The coefficients are obtained by solving

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \frac{1}{n_{\mathrm{nu}}} \sum_{j=1}^{n_{\mathrm{nu}}} \log \left[\sum_{q=1}^{b} \alpha_q \phi_q \left(\mathbf{x}_j^{\mathrm{nu}} \right) \right] \\ \text{s.t.} \quad & \frac{1}{n_{\mathrm{de}}} \sum_{i=1}^{n_{\mathrm{de}}} \sum_{q=1}^{b} \alpha_q \phi_q \left(\mathbf{x}_i^{\mathrm{de}} \right) = 1, \\ & \alpha_q \geq 0, \qquad q = 1, \ldots, b. \end{aligned} \tag{12}$$

For a granular-ball center $\mathbf{c}_i$, its RPD score is calculated according to its class label:

$$\mathrm{RPD}(\mathbf{c}_i) = \begin{cases} \widehat{r}_+(\mathbf{c}_i), & l_i = +1, \\ \widehat{r}_-(\mathbf{c}_i), & l_i = -1. \end{cases} \tag{13}$$

The corresponding weight is defined as

$$s_i = \frac{1}{\mathrm{RPD}(\mathbf{c}_i) + \varepsilon}, \tag{14}$$

where $\varepsilon > 0$ is a small constant. Thus, granular balls that are more consistent with the opposite-class distribution receive smaller weights.

2.3 Twin Support Vector Machine and Fuzzy Support Vector Machine

2.3.1 Twin Support Vector Machine

Let

$$A \in \mathbb{R}^{m_1 \times d}, \qquad B \in \mathbb{R}^{m_2 \times d} \tag{15}$$

denote the positive- and negative-class sample matrices. TWSVM constructs two nonparallel hyperplanes [2]:

$$\mathbf{w}_1^T \mathbf{x} + b_1 = 0, \qquad \mathbf{w}_2^T \mathbf{x} + b_2 = 0. \tag{16}$$

The primal problems of linear TWSVM are

$$\begin{aligned} \min_{\mathbf{w}_1, b_1, \boldsymbol{\xi}_2} \quad & \frac{1}{2} \left\| A\mathbf{w}_1 + \mathbf{e}_1 b_1 \right\|_2^2 + c_1 \mathbf{e}_2^T \boldsymbol{\xi}_2 \\ \text{s.t.} \quad & -\left(B\mathbf{w}_1 + \mathbf{e}_2 b_1 \right) + \boldsymbol{\xi}_2 \geq \mathbf{e}_2, \qquad \boldsymbol{\xi}_2 \geq \mathbf{0}. \end{aligned} \tag{17}$$

and

$$\begin{aligned} \min_{\mathbf{w}_2, b_2, \boldsymbol{\xi}_1} \quad & \frac{1}{2} \left\| B\mathbf{w}_2 + \mathbf{e}_2 b_2 \right\|_2^2 + c_2 \mathbf{e}_1^T \boldsymbol{\xi}_1 \\ \text{s.t.} \quad & \left(A\mathbf{w}_2 + \mathbf{e}_1 b_2 \right) + \boldsymbol{\xi}_1 \geq \mathbf{e}_1, \qquad \boldsymbol{\xi}_1 \geq \mathbf{0}. \end{aligned} \tag{18}$$

Let

$$H = [A\ \mathbf{e}_1], \qquad G = [B\ \mathbf{e}_2]. \tag{19}$$

The dual problems are

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \mathbf{e}_2^T \boldsymbol{\alpha} - \frac{1}{2} \boldsymbol{\alpha}^T G \left(H^T H + \varepsilon I \right)^{-1} G^T \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq c_1 \mathbf{e}_2, \end{aligned} \tag{20}$$

and

$$\begin{aligned} \max_{\boldsymbol{\gamma}} \quad & \mathbf{e}_1^T \boldsymbol{\gamma} - \frac{1}{2} \boldsymbol{\gamma}^T H \left(G^T G + \varepsilon I \right)^{-1} H^T \boldsymbol{\gamma} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\gamma} \leq c_2 \mathbf{e}_1. \end{aligned} \tag{21}$$

The two hyperplane parameters are obtained by

$$\begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix} = - \left(H^T H + \varepsilon I \right)^{-1} G^T \boldsymbol{\alpha}, \tag{22}$$

and

$$\begin{bmatrix} \mathbf{w}_2 \\ b_2 \end{bmatrix} = \left(G^T G + \varepsilon I \right)^{-1} H^T \boldsymbol{\gamma}. \tag{23}$$

For a test sample $\mathbf{x}$, the predicted label is determined by

$$\widehat{y}(\mathbf{x}) = \arg \min_{q \in \{1,2\}} \frac{\left| \mathbf{w}_q^T \mathbf{x} + b_q \right|}{\|\mathbf{w}_q\|_2}. \tag{24}$$

2.3.2 Fuzzy Support Vector Machine

FSVM assigns different membership degrees to training samples and reduces the contribution of unreliable samples [34]. Given

$$D_f = \{(\mathbf{x}_i, y_i, s_i) \mid i = 1, \ldots, n\}, \tag{25}$$

where $s_i \in (0, 1]$ is the membership degree of $\mathbf{x}_i$, the primal problem of FSVM is

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}} \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^{n} s_i \xi_i \\ \text{s.t.} \quad & y_i \left[\mathbf{w}^T \phi(\mathbf{x}_i) + b \right] \geq 1 - \xi_i, \\ & \xi_i \geq 0, \qquad i = 1, \ldots, n. \end{aligned} \tag{26}$$

Its dual form is

$$\max_{\boldsymbol{\lambda}} \quad \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_i \lambda_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{27}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} \lambda_i y_i = 0,$$
$$0 \le \lambda_i \le C s_i, \qquad i = 1, \dots, n.$$

Equation (27) shows that the membership degree controls the upper bound of the corresponding Lagrange multiplier. Samples with smaller membership degrees therefore have weaker effects on the classification hyperplane. This mechanism provides the basic idea for introducing RPD-based granular-ball weights into the proposed model.

3 RPDLDGBTWSVM

Conventional TWSVM uses individual samples as the basic learning units and assigns the same penalty to different training samples. Therefore, its optimization scale increases with the number of samples, while noisy samples, boundary samples, and outliers may have excessive effects on the classification hyperplanes. To address these problems, this paper combines LDGBG, RPD, and TWSVM and proposes RPDLDGBTWSVM.

Suppose that LDGBG generates m_1 positive granular balls and m_2 negative granular balls. The corresponding granular-ball sets are denoted by

$$\mathcal{G}_A = \left\{ (\mathbf{c}_i^+, r_i^+, s_i^+) \mid i = 1, \dots, m_1 \right\}, \tag{28}$$

and

$$\mathcal{G}_B = \left\{ (\mathbf{c}_j^-, r_j^-, s_j^-) \mid j = 1, \dots, m_2 \right\}, \tag{29}$$

where $\mathbf{c}_i^+$ and $\mathbf{c}_j^-$ are granular-ball centers, r_i^+ and r_j^- are granular-ball radii, and s_i^+ and s_j^- are the corresponding RPD-based weights.

Let

$$C_A = \begin{bmatrix} (\mathbf{c}_1^+)^T \\ \vdots \\ (\mathbf{c}_{m_1}^+)^T \end{bmatrix}, \qquad C_B = \begin{bmatrix} (\mathbf{c}_1^-)^T \\ \vdots \\ (\mathbf{c}_{m_2}^-)^T \end{bmatrix} \tag{30}$$

denote the center matrices of the positive and negative granular balls. Their radius vectors are defined as

$$\mathbf{r}_A = \begin{bmatrix} r_1^+ & \cdots & r_{m_1}^+ \end{bmatrix}^T, \qquad \mathbf{r}_B = \begin{bmatrix} r_1^- & \cdots & r_{m_2}^- \end{bmatrix}^T. \tag{31}$$

According to the RPD estimation described in Section 2.2, the granular-ball weights are calculated by

$$s_i^+ = \frac{1}{\widehat{\mathrm{RPD}}_+(\mathbf{c}_i^+) + \varepsilon}, \qquad s_j^- = \frac{1}{\widehat{\mathrm{RPD}}_-(\mathbf{c}_j^-) + \varepsilon}, \tag{32}$$

where $\varepsilon > 0$ is a small constant. A granular ball that is relatively more consistent with the opposite-class distribution receives a smaller weight.

The diagonal weight matrices are

$$D_A = \operatorname{diag}\left(s_1^+, \ldots, s_{m_1}^+\right), \qquad D_B = \operatorname{diag}\left(s_1^-, \ldots, s_{m_2}^-\right). \tag{33}$$

3.1 Linear RPDLDGBTWSVM

For a linear classification problem, RPDLDGBTWSVM constructs two nonparallel hyperplanes:

$$\mathbf{w}_1^T\mathbf{x} + b_1 = 0, \qquad \mathbf{w}_2^T\mathbf{x} + b_2 = 0. \tag{34}$$

The first hyperplane is close to the positive granular-ball centers and far from the surfaces of the negative granular balls. Its primal problem is formulated as

$$\begin{aligned} \min_{\mathbf{w}_1, b_1, \boldsymbol{\xi}_B} \quad & \frac{1}{2}\left\|C_A\mathbf{w}_1 + \mathbf{e}_A b_1\right\|_2^2 + \frac{c_3}{2}\left\|\mathbf{w}_1\right\|_2^2 + c_1\mathbf{e}_B^T D_B \boldsymbol{\xi}_B \\ \text{s.t.} \quad & -\left(C_B\mathbf{w}_1 + \mathbf{e}_B b_1\right) - \mathbf{r}_B + \boldsymbol{\xi}_B \geq \mathbf{e}_B, \quad \boldsymbol{\xi}_B \geq \mathbf{0}. \end{aligned} \tag{35}$$

Similarly, the second hyperplane is close to the negative granular-ball centers and far from the surfaces of the positive granular balls. Its primal problem is

$$\begin{aligned} \min_{\mathbf{w}_2, b_2, \boldsymbol{\xi}_A} \quad & \frac{1}{2}\left\|C_B\mathbf{w}_2 + \mathbf{e}_B b_2\right\|_2^2 + \frac{c_4}{2}\left\|\mathbf{w}_2\right\|_2^2 + c_2\mathbf{e}_A^T D_A \boldsymbol{\xi}_A \\ \text{s.t.} \quad & \left(C_A\mathbf{w}_2 + \mathbf{e}_A b_2\right) - \mathbf{r}_A + \boldsymbol{\xi}_A \geq \mathbf{e}_A, \quad \boldsymbol{\xi}_A \geq \mathbf{0}. \end{aligned} \tag{36}$$

Compared with conventional TWSVM, the proposed model contains three modifications. First, point-level samples are replaced by granular-ball centers, reducing the number of training objects. Second, granular-ball radii are introduced into the constraints, so that the classification hyperplanes separate granular-ball surfaces rather than only their centers. Third, RPD-based weights are introduced into the slack-variable penalties, thereby reducing the influence of unreliable granular balls.

To derive the solution of (35), define

$$E = \begin{bmatrix} C_A & \mathbf{e}_A \end{bmatrix}, \qquad F = \begin{bmatrix} C_B & \mathbf{e}_B \end{bmatrix}, \tag{37}$$

and

$$\mathbf{z}_1 = \begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix}, \qquad \mathbf{z}_2 = \begin{bmatrix} \mathbf{w}_2 \\ b_2 \end{bmatrix}. \tag{38}$$

Because the regularization terms are imposed on the normal vectors rather than the bias terms, define

$$J_d = \begin{bmatrix} I_d & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}. \tag{39}$$

Then, the first primal problem can be written as

$$\begin{aligned} \min_{\mathbf{z}_1, \boldsymbol{\xi}_B} \quad & \frac{1}{2}\mathbf{z}_1^T\left(E^T E + c_3 J_d\right)\mathbf{z}_1 + c_1\mathbf{e}_B^T D_B \boldsymbol{\xi}_B \\ \text{s.t.} \quad & -F\mathbf{z}_1 + \boldsymbol{\xi}_B \geq \mathbf{e}_B + \mathbf{r}_B, \quad \boldsymbol{\xi}_B \geq \mathbf{0}. \end{aligned} \tag{40}$$

Let $\boldsymbol{\alpha} \geq \mathbf{0}$ and $\boldsymbol{\beta} \geq \mathbf{0}$ be the Lagrange multiplier vectors. The Lagrangian function is

$$\begin{aligned} L_1 =& \frac{1}{2}\mathbf{z}_1^T \left(E^T E + c_3 J_d\right) \mathbf{z}_1 + c_1 \mathbf{e}_B^T D_B \boldsymbol{\xi}_B \\ & - \boldsymbol{\alpha}^T \left(-F\mathbf{z}_1 + \boldsymbol{\xi}_B - \mathbf{e}_B - \mathbf{r}_B\right) - \boldsymbol{\beta}^T \boldsymbol{\xi}_B. \end{aligned} \tag{41}$$

According to the KKT conditions,

$$\left(E^T E + c_3 J_d\right) \mathbf{z}_1 + F^T \boldsymbol{\alpha} = \mathbf{0}, \tag{42}$$

and

$$c_1 D_B \mathbf{e}_B - \boldsymbol{\alpha} - \boldsymbol{\beta} = \mathbf{0}. \tag{43}$$

Let

$$M_1 = E^T E + c_3 J_d. \tag{44}$$

Then

$$\mathbf{z}_1 = -M_1^{-1} F^T \boldsymbol{\alpha}. \tag{45}$$

Since $\boldsymbol{\beta} \geq \mathbf{0}$, the dual variables satisfy

$$\mathbf{0} \leq \boldsymbol{\alpha} \leq c_1 D_B \mathbf{e}_B. \tag{46}$$

Substituting (45) into the Lagrangian gives the first dual problem:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^T \left(\mathbf{e}_B + \mathbf{r}_B\right) - \frac{1}{2}\boldsymbol{\alpha}^T F M_1^{-1} F^T \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq c_1 D_B \mathbf{e}_B. \end{aligned} \tag{47}$$

For the second hyperplane, let

$$M_2 = F^T F + c_4 J_d. \tag{48}$$

Following the same derivation, the second dual problem is

$$\begin{aligned} \max_{\boldsymbol{\gamma}} \quad & \boldsymbol{\gamma}^T \left(\mathbf{e}_A + \mathbf{r}_A\right) - \frac{1}{2}\boldsymbol{\gamma}^T E M_2^{-1} E^T \boldsymbol{\gamma} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\gamma} \leq c_2 D_A \mathbf{e}_A. \end{aligned} \tag{49}$$

After solving (47) and (49), the parameters of the two hyperplanes are obtained as

$$\begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix} = -M_1^{-1} F^T \boldsymbol{\alpha}, \qquad \begin{bmatrix} \mathbf{w}_2 \\ b_2 \end{bmatrix} = M_2^{-1} E^T \boldsymbol{\gamma}. \tag{50}$$

For a test sample $\mathbf{x}$, its class label is determined by

$$\widehat{y}(\mathbf{x}) = \arg \min_{q \in \{1,2\}} \frac{\left|\mathbf{w}_q^T \mathbf{x} + b_q\right|}{\|\mathbf{w}_q\|_2}. \tag{51}$$

Proposition 1. If $c_1, c_2, c_3, c_4 > 0$, D_A and D_B are positive diagonal matrices, and both classes contain at least one granular ball, then the two primal problems in (35) and (36) are feasible convex optimization problems and admit finite optimal solutions. Moreover, the obtained hyperplane parameters are unique.

Proof. For the first problem, choose

$$\mathbf{w}_1 = \mathbf{0}, \qquad b_1 = 0, \qquad \boldsymbol{\xi}_B = \mathbf{e}_B + \mathbf{r}_B + \tau \mathbf{e}_B,$$

where $\tau > 0$. This point strictly satisfies all constraints. Therefore, the feasible set is nonempty and Slater's condition holds.

For any nonzero

$$\mathbf{z}_1 = \begin{bmatrix} \mathbf{w}_1 \\ b_1 \end{bmatrix},$$

we have

$$\mathbf{z}_1^T M_1 \mathbf{z}_1 = \|C_A \mathbf{w}_1 + \mathbf{e}_A b_1\|_2^2 + c_3 \|\mathbf{w}_1\|_2^2 > 0. \tag{52}$$

Indeed, if the right-hand side is zero, then $c_3 > 0$ implies $\mathbf{w}_1 = \mathbf{0}$, and consequently $\mathbf{e}_A b_1 = \mathbf{0}$. Since $m_1 > 0$, it follows that $b_1 = 0$, which contradicts $\mathbf{z}_1 \neq \mathbf{0}$. Therefore, M_1 is positive definite.

Hence, the objective function is strictly convex with respect to $(\mathbf{w}_1, b_1)$. Since all RPD weights are positive, the slack penalty is nonnegative and increases with $\boldsymbol{\xi}_B$. Together with the affine constraints, the first primal problem has a finite optimum and a unique hyperplane parameter. The same argument applies to the second primal problem. Slater's condition further guarantees strong duality between the primal and dual problems. This completes the proof.

3.2 Nonlinear RPDLDGBTWSVM

For datasets with nonlinear class boundaries, a kernel function is introduced. Let

$$C = \begin{bmatrix} C_A \\ C_B \end{bmatrix} \tag{53}$$

denote the matrix containing all granular-ball centers, and let

$$K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x})^T \phi(\mathbf{z}) \tag{54}$$

be a kernel function.

Define

$$K_A = K(C_A, C^T), \qquad K_B = K(C_B, C^T). \tag{55}$$

The two nonlinear hyperplanes are expressed as

$$K(\mathbf{x}, C^T)\mathbf{u}_1 + b_1 = 0, \qquad K(\mathbf{x}, C^T)\mathbf{u}_2 + b_2 = 0. \tag{56}$$

The first nonlinear primal problem is

$$\begin{aligned} \min_{\mathbf{u}_1, b_1, \boldsymbol{\xi}_B} \quad & \frac{1}{2} \left\| K_A \mathbf{u}_1 + \mathbf{e}_A b_1 \right\|_2^2 + \frac{c_3}{2} \left\| \mathbf{u}_1 \right\|_2^2 + c_1 \mathbf{e}_B^T D_B \boldsymbol{\xi}_B \\ \text{s.t.} \quad & -\left(K_B \mathbf{u}_1 + \mathbf{e}_B b_1 \right) - \mathbf{r}_B + \boldsymbol{\xi}_B \geq \mathbf{e}_B, \quad \boldsymbol{\xi}_B \geq \mathbf{0}. \end{aligned} \tag{57}$$

The second nonlinear primal problem is

$$\begin{aligned} \min_{\mathbf{u}_2, b_2, \boldsymbol{\xi}_A} \quad & \frac{1}{2} \left\| K_B \mathbf{u}_2 + \mathbf{e}_B b_2 \right\|_2^2 + \frac{c_4}{2} \left\| \mathbf{u}_2 \right\|_2^2 + c_2 \mathbf{e}_A^T D_A \boldsymbol{\xi}_A \\ \text{s.t.} \quad & -\left(K_A \mathbf{u}_2 + \mathbf{e}_A b_2 \right) - \mathbf{r}_A + \boldsymbol{\xi}_A \geq \mathbf{e}_A, \quad \boldsymbol{\xi}_A \geq \mathbf{0}. \end{aligned} \tag{58}$$

Define

$$P = \left[K_A \ \mathbf{e}_A \right], \qquad Q = \left[K_B \ \mathbf{e}_B \right], \tag{59}$$

and

$$J_m = \begin{bmatrix} I_m & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix}, \qquad m = m_1 + m_2. \tag{60}$$

Let

$$M_{K1} = P^T P + c_3 J_m, \qquad M_{K2} = Q^T Q + c_4 J_m. \tag{61}$$

Following the same Lagrangian and KKT derivation as the linear model, the two nonlinear dual problems are

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \boldsymbol{\alpha}^T \left(\mathbf{e}_B + \mathbf{r}_B \right) - \frac{1}{2} \boldsymbol{\alpha}^T Q M_{K1}^{-1} Q^T \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha} \leq c_1 D_B \mathbf{e}_B, \end{aligned} \tag{62}$$

and

$$\begin{aligned} \max_{\boldsymbol{\gamma}} \quad & \boldsymbol{\gamma}^T \left(\mathbf{e}_A + \mathbf{r}_A \right) - \frac{1}{2} \boldsymbol{\gamma}^T P M_{K2}^{-1} P^T \boldsymbol{\gamma} \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\gamma} \leq c_2 D_A \mathbf{e}_A. \end{aligned} \tag{63}$$

The nonlinear hyperplane parameters are obtained by

$$\begin{bmatrix} \mathbf{u}_1 \\ b_1 \end{bmatrix} = -M_{K1}^{-1} Q^T \boldsymbol{\alpha}, \qquad \begin{bmatrix} \mathbf{u}_2 \\ b_2 \end{bmatrix} = M_{K2}^{-1} P^T \boldsymbol{\gamma}. \tag{64}$$

Let

$$K_C = K(C, C^T). \tag{65}$$

The norm of the qth nonlinear hyperplane in the reproducing kernel Hilbert space is

$$\left\| \mathbf{w}_q \right\|_{\mathcal{H}}^2 = \mathbf{u}_q^T K_C \mathbf{u}_q. \tag{66}$$

Therefore, the distance between a test sample $\mathbf{x}$ and the qth nonlinear hyperplane is

$$d_q^K(\mathbf{x}) = \frac{\left| K(\mathbf{x}, C^T) \mathbf{u}_q + b_q \right|}{\sqrt{\mathbf{u}_q^T K_C \mathbf{u}_q}}, \qquad q = 1, 2. \tag{67}$$

The nonlinear decision function is

$$\widehat{y}(\mathbf{x}) = \arg \min_{q \in \{1,2\}} d_q^K(\mathbf{x}). \tag{68}$$

The positive-definiteness and feasibility argument in Proposition 1 also applies to the nonlinear model after replacing E and F with P and Q. Therefore, the nonlinear primal problems are convex and satisfy strong duality.

Algorithm 1. Projected gradient ascent for estimating $\widehat{RPD}(x)$

Input: Basis functions $\{\psi_i(x)\}_{i=1}^{l}$, positive sample set T^+, and negative sample set T^-
Output: $\widehat{RPD}(x)$

Initialize $\lambda > 0$, $0 < \varepsilon \ll 1$
Compute

$$A_{j,i} \leftarrow \psi_i(x_j^+)$$

$$b_i \leftarrow \frac{1}{N_-}\sum_{j=1}^{N_-}\psi_i(x_j^-)$$

repeat
    $\lambda \leftarrow \lambda + \varepsilon A^T(1./A\lambda)$
    $\lambda \leftarrow \lambda + \dfrac{1 - b^T\lambda}{b^Tb}b$
    $\lambda \leftarrow \max(0, \lambda)$
    $\lambda \leftarrow \lambda/(b^T\lambda)$
until convergence
Obtain

$$\widehat{RPD}(x) \leftarrow \sum_{i=1}^{l}\lambda_i\psi_i(x)$$

Algorithm 2. Pseudo code of linear RPDLDGBTWSVM

Input: Training dataset $\{(x_i, y_i) \mid i = 1, 2, \ldots, N\}$, granularity coefficient g, and test sample x
Output: Predicted label for test sample x

Generate the granular-ball set by LDGBG

$$\mathcal{G} = \{(c_i, r_i, l_i)\}_{i=1}^{m}$$

Construct the center matrices and radius vectors

$$C_A, \quad C_B, \quad r_A, \quad r_B$$

Estimate the RPD values of all granular-ball centers by Algorithm 1
Compute the granular-ball weights

$$s_i^+ = \frac{1}{\widehat{RPD}_+(c_i^+) + \varepsilon},$$

$$s_j^- = \frac{1}{\widehat{RPD}_-(c_j^-) + \varepsilon}.$$

Construct the diagonal weighting matrices

$$D_A = \mathrm{diag}(s_1^+, s_2^+, \ldots, s_{m_1}^+)$$

$$D_B = \mathrm{diag}(s_1^-, s_2^-, \ldots, s_{m_2}^-)$$

Set the training parameters c_1, c_2, c_3, c_4
Obtain w_1 and b_1 by solving (47)
Obtain w_2 and b_2 by solving (49)
Classify the test sample x by

$$y(x) = \arg\min_{q\in\{1,2\}} \frac{\left|w_q^T x + b_q\right|}{\|w_q\|_2}$$

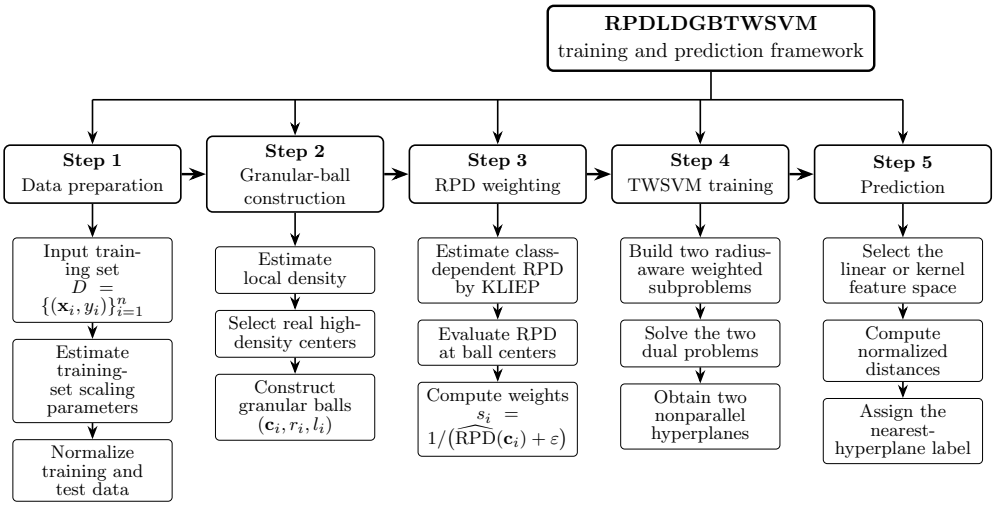


Fig. 1 Overall workflow of RPDLDGBTWSVM.

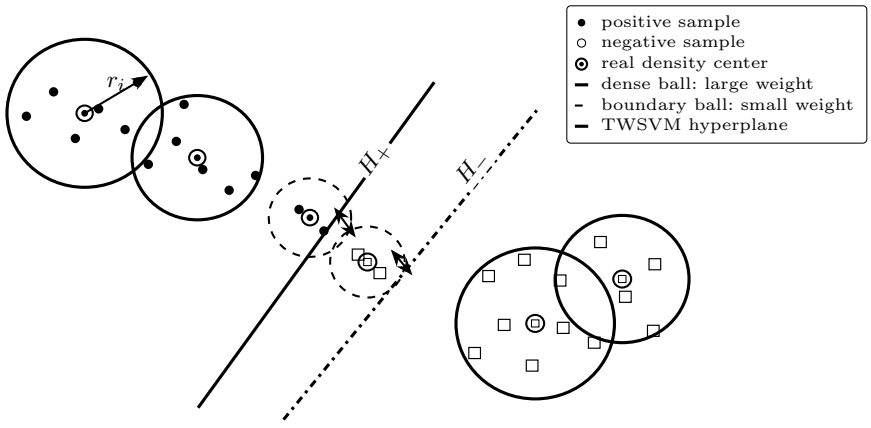


Fig. 2 Two-dimensional mechanism of RPDLDGBTWSVM.

Figure 1 presents the complete computation process of RPDLDGBTWSVM. The framework contains five major stages, and each stage consists of several elementary operations. The scaling parameters are estimated only from the training set and are then applied to the training and test data. LDGBG selects real high-density samples as centers and constructs the corresponding granular balls. KLIEP estimates the class-dependent relative density ratios at the granular-ball centers. The estimates are converted into RPD weights. The ball radii and RPD weights are incorporated into the two TWSVM subproblems. Solving the corresponding dual problems produces two nonparallel hyperplanes. A test sample is assigned according to its normalized distances to the two hyperplanes in the linear or kernel feature space.

Figure 2 provides a simplified geometric interpretation of the proposed method. Solid granular balls represent dense intra-class regions and receive relatively large RPD weights. Dashed granular balls are located near the class boundary and receive smaller weights. Each granular-ball center is an observed high-density sample rather than a virtual mean. The radius r_i is incorporated into the constraints. The two hyperplanes are therefore learned relative to the granular-ball surfaces instead of only the ball centers.

3.3 Mathematical Justification

This subsection establishes an approximation bound between the point-level RPD-weighted loss and the corresponding granular-ball-level loss. It also shows that the RPD weighting does not change the convexity of the resulting optimization problems.

Let

$$G_i = \{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \ldots, \mathbf{x}_{in_i}\} \tag{69}$$

be the sample set represented by the ith granular ball. Let $\mathbf{c}_i$, r_i, and y_i denote its center, radius, and class label, respectively. Then,

$$\|\mathbf{x}_{ij} - \mathbf{c}_i\|_2 \leq r_i, \qquad j = 1, \ldots, n_i. \tag{70}$$

For a sample belonging to the class of G_i, let $q_i(\mathbf{x})$ denote its class-dependent RPD value. The corresponding weight is defined as

$$s_i(\mathbf{x}) = \frac{1}{q_i(\mathbf{x}) + \varepsilon}, \qquad \varepsilon > 0. \tag{71}$$

Theorem 1 (Approximation bound of the RPD-weighted loss) *Assume that $q_i(\mathbf{x}) \geq 0$ and that q_i is Lipschitz continuous on each granular ball:*

$$|q_i(\mathbf{x}) - q_i(\mathbf{z})| \leq L_q \|\mathbf{x} - \mathbf{z}\|_2 . \tag{72}$$

Assume that, for a fixed classifier f, the loss satisfies

$$0 \leq \ell(f(\mathbf{x}), y_i) \leq M_\ell \tag{73}$$

and

$$|\ell(f(\mathbf{x}), y_i) - \ell(f(\mathbf{z}), y_i)| \leq L_\ell \|\mathbf{x} - \mathbf{z}\|_2 . \tag{74}$$

Then the point-level and granular-ball-level RPD-weighted empirical losses satisfy

$$|\mathcal{R}_P(f) - \mathcal{R}_G(f)| \leq \left(\frac{M_\ell L_q}{\varepsilon^2} + \frac{L_\ell}{\varepsilon} \right) \sum_{i=1}^{m} \frac{n_i}{n} r_i, \tag{75}$$

where

$$\mathcal{R}_P(f) = \frac{1}{n} \sum_{i=1}^{m} \sum_{j=1}^{n_i} s_i(\mathbf{x}_{ij}) \ell(f(\mathbf{x}_{ij}), y_i), \tag{76}$$

$$\mathcal{R}_G(f) = \sum_{i=1}^{m} \frac{n_i}{n} s_i(\mathbf{c}_i) \ell(f(\mathbf{c}_i), y_i), \tag{77}$$

and

$$n = \sum_{i=1}^{m} n_i. \tag{78}$$

Proof For any $\mathbf{x}_{ij} \in G_i$, the difference between its RPD weight and the center weight is

$$\begin{aligned}\left|s_i(\mathbf{x}_{ij}) - s_i(\mathbf{c}_i)\right| &= \frac{\left|q_i(\mathbf{x}_{ij}) - q_i(\mathbf{c}_i)\right|}{\left(q_i(\mathbf{x}_{ij}) + \varepsilon\right)\left(q_i(\mathbf{c}_i) + \varepsilon\right)} \\ &\leq \frac{L_q}{\varepsilon^2}\left\|\mathbf{x}_{ij} - \mathbf{c}_i\right\|_2 \\ &\leq \frac{L_q}{\varepsilon^2} r_i. \end{aligned} \tag{79}$$

Since $q_i(\mathbf{c}_i) \geq 0$,

$$0 < s_i(\mathbf{c}_i) \leq \frac{1}{\varepsilon}. \tag{80}$$

Using the triangle inequality gives

$$\begin{aligned}&\left|s_i(\mathbf{x}_{ij})\ell(f(\mathbf{x}_{ij}), y_i) - s_i(\mathbf{c}_i)\ell(f(\mathbf{c}_i), y_i)\right| \\ &\leq \left|s_i(\mathbf{x}_{ij}) - s_i(\mathbf{c}_i)\right| \ell(f(\mathbf{x}_{ij}), y_i) \\ &\quad + s_i(\mathbf{c}_i)\left|\ell(f(\mathbf{x}_{ij}), y_i) - \ell(f(\mathbf{c}_i), y_i)\right|. \end{aligned} \tag{81}$$

Applying (70), (73), (74), (79), and (80) yields

$$\left|s_i(\mathbf{x}_{ij})\ell(f(\mathbf{x}_{ij}), y_i) - s_i(\mathbf{c}_i)\ell(f(\mathbf{c}_i), y_i)\right| \leq \left(\frac{M_\ell L_q}{\varepsilon^2} + \frac{L_\ell}{\varepsilon}\right) r_i. \tag{82}$$

Summing (82) over all samples and dividing by n gives

$$\begin{aligned}\left|\mathcal{R}_P(f) - \mathcal{R}_G(f)\right| &\leq \frac{1}{n}\sum_{i=1}^{m}\sum_{j=1}^{n_i}\left(\frac{M_\ell L_q}{\varepsilon^2} + \frac{L_\ell}{\varepsilon}\right) r_i \\ &= \left(\frac{M_\ell L_q}{\varepsilon^2} + \frac{L_\ell}{\varepsilon}\right)\sum_{i=1}^{m}\frac{n_i}{n} r_i. \end{aligned} \tag{83}$$

This proves (75). □

For the hinge-type loss associated with a fixed hyperplane, the loss is the maximum of two affine functions. It is therefore Lipschitz continuous. On a finite dataset, it is also bounded. The assumptions of Theorem 1 are thus satisfied for the empirical loss considered here.

Since

$$\sum_{i=1}^{m}\frac{n_i}{n} r_i \leq \max_{1\leq i\leq m} r_i, \tag{84}$$

Theorem 1 directly gives the following result:

$$\max_{1\leq i\leq m} r_i \longrightarrow 0 \quad \Longrightarrow \quad \left|\mathcal{R}_P(f) - \mathcal{R}_G(f)\right| \longrightarrow 0. \tag{85}$$

The approximation error is therefore controlled by the granular-ball radii. Compact granular balls provide center-level weighted losses close to the corresponding point-level weighted losses. LDGBG uses real samples with high local density as centers. These centers retain local distribution information without introducing virtual representative points.

Theorem 2 (Convexity and feasibility of the weighted subproblems) *Let the center weights be defined by (71), with $\varepsilon > 0$ and $q_i(\mathbf{c}_i) \geq 0$. Assume that the quadratic matrices in the two RPDLDGBTWSVM subproblems are positive semidefinite, the constraints are affine, and the slack variables are nonnegative. Then both subproblems are convex and feasible.*

Proof For every granular-ball center,

$$s_i(\mathbf{c}_i) = \frac{1}{q_i(\mathbf{c}_i) + \varepsilon} > 0. \tag{86}$$

Hence, the weight matrices D_A and D_B are positive diagonal matrices. The weighted slack terms are linear in the slack variables and remain nonnegative.

The fitting terms are convex quadratic functions because their Hessian matrices are positive semidefinite. The constraints are affine. Therefore, each subproblem is a convex quadratic programming problem.

For any fixed hyperplane parameters, sufficiently large nonnegative slack variables satisfy the inequality constraints. The feasible sets are therefore nonempty. □

The two results show that center-level RPD weighting has a controlled approximation error and preserves the convex structure of RPDLDGBTWSVM.

3.4 Time Complexity

Let n be the number of training samples, d be the feature dimension, and $m = m_1 + m_2$ be the number of granular balls generated by LDGBG. Define the granular-ball retention ratio as $r = m/n$, where generally $r \ll 1$.

Granular-ball generation is usually the most computationally expensive stage of the proposed method. The computational complexity of RPDLDGBTWSVM mainly consists of three parts: granular-ball generation, RPD estimation, and TWSVM optimization. LDGBG calculates class-wise sample distances, local-density centers, and granular-ball radii, and assigns each sample to its nearest granular ball. For a dataset with k classes, its main time complexity is approximately $O(n^2 d/k + nmd)$. Since $m = rn$, the complexity of granular-ball generation can also be written as $O!\left(n^2 d/k + rn^2 d\right)$.

The two weighted TWSVM subproblems are constructed at the granular-ball level. Similarly, RPD estimation is performed on the m granular-ball centers rather than on the n original samples. The implemented KLIEP method constructs dense kernel matrices and uses a limited number of iterations, resulting in an approximate time complexity of $O(m^2 d)$. For the nonlinear model, kernel matrix construction requires $O(m^2 d)$ time, and the main dense matrix operations require approximately $O(m^3)$ time. The SOR solver introduces an additional cost of $O(S_1 m_2^2 + S_2 m_1^2)$, where S_1 and S_2 denote the numbers of SOR sweeps required by the two subproblems.

The overall training time complexity of nonlinear RPDLDGBTWSVM can therefore be summarized as $O!\left(n^2 d/k + nmd + m^2 d + m^3 + S_1 m_2^2 + S_2 m_1^2\right)$. For the linear model, the kernel-related $O(m^3)$ term is replaced by lower-dimensional matrix operations associated with the feature dimension d. When $m \ll n$, the overall computational cost of the model is usually dominated by LDGBG.

Traditional nonlinear TWSVM constructs kernel matrices and optimization problems directly from the original samples. Therefore, its kernel matrix construction cost grows quadratically with n, while its dense matrix computation cost grows cubically with n. In contrast, RPDLDGBTWSVM reduces the input size of RPD estimation and TWSVM optimization from n samples to $m = rn$ granular balls. As a result, the quadratic matrix computation cost is reduced from $O(n^2)$ to $O(m^2) = O(r^2n^2)$, while the cubic matrix computation cost is reduced from $O(n^3)$ to $O(m^3) = O(r^3n^3)$. Therefore, a smaller value of r provides a more evident computational advantage in the classifier optimization stage.

4 Experiments

This section evaluates the classification performance, running time, noise robustness, granular-ball compression, parameter sensitivity, and statistical significance of RPDLDGBTWSVM. The experiments contain five parts. The first part compares classification performance on datasets without artificial noise. The second part compares running time. The third part tests robustness under Gaussian feature noise. The fourth part studies granular-ball compression and parameter sensitivity. The final part reports the Friedman test and the Nemenyi post-hoc test.

4.1 Experimental Setup

All experiments are implemented in MATLAB R2021b on a 64-bit Windows system equipped with an AMD Ryzen 5 5500U processor at 2.10 GHz and 16 GB RAM. The device name, graphics-card memory, storage usage, device ID, product ID, and touch-screen information are not reported because they do not affect algorithmic reproducibility.

Each dataset is randomly divided into 80% training samples and 20% testing samples. For the rth repetition, the data partition is generated using `rng(2026+r)`. The same partition is used for all compared methods in that repetition. The normalization parameters are estimated from the training set using `mapminmax` and then applied unchanged to the testing set. Each experiment is independently repeated ten times. The mean value and standard deviation are reported.

The compared methods are TWSVM, GBTWSVM, GBFTWSVM, RPDGBTWSVM, RPDTWSVM, and RPDLDGBTWSVM. The same paired data partitions and parameter-search ranges are used for all methods where applicable. For RPDLDGBTWSVM, the regularization parameters are fixed as $c_3 = c_4 = 10^{-5}$.

The Gaussian kernel is defined as

$$K(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|_2^2}{\sigma^2}\right), \tag{87}$$

where σ is the kernel parameter. The search ranges are

$$c_1, c_2, \sigma \in \left\{2^{-5}, 2^{-4}, \ldots, 2^5\right\}, \tag{88}$$

and

$$g \in \{0.5, 0.6, \ldots, 1.5\}. \tag{89}$$

The penalty parameters are set as $c_1 = c_2$ during parameter selection. The optimal parameters of all models are not listed in a separate table. Such a table is long and does not explain parameter behavior. The parameter influence is studied directly in Section 4.7.

In the Gaussian-noise experiments, the noisy training set in the rth repetition is generated using `rng(9000+r)`. All compared methods receive the same noisy training set and the same clean testing set in each repetition. The noise comparison is therefore paired.

Accuracy is used as the main comparison measure. Precision, Recall, Specificity, F1-score, and G-mean are used as supplementary measures. Let TP, TN, FP, and FN denote the numbers of true positives, true negatives, false positives, and false negatives, respectively. The metrics are defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{90}$$

$$\text{Precision} = \frac{TP}{TP + FP}, \qquad \text{Recall} = \frac{TP}{TP + FN}, \tag{91}$$

$$\text{Specificity} = \frac{TN}{TN + FP}, \tag{92}$$

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{93}$$

and

$$\text{G-mean} = \sqrt{\text{Recall} \cdot \text{Specificity}}. \tag{94}$$

4.2 Dataset Information

The experiments use 17 UCI benchmark datasets and two MedMNIST medical image datasets. Their basic information is listed in Table 1. The UCI datasets are cited using the repository reference. The medical image datasets are cited using the MedMNIST reference.

All 19 datasets are used in the main accuracy comparison and the statistical test.

4.3 Classification Results without Artificial Noise

Table 2 reports the classification accuracy on the 19 datasets. The best result on each dataset is shown in bold.

RPDLDGBTWSVM obtains the highest or tied-highest accuracy on 18 of the 19 datasets. Its average accuracy is 90.39%. This value is higher than the average accuracies of the other five methods.

The improvement is clear on Parkinson, Darwin, Mammographic, Spect, and Med-Pneumonia. On Parkinson, RPDLDGBTWSVM reaches 93.13%. The second-highest result is 89.99%. The improvement is 3.14 percentage points. On Darwin, the improvement over the second-highest result is 3.35 percentage points. On Mammographic, the improvement is 3.31 percentage points.

Table 1 Information on the 19 datasets used in the experiments.

Dataset	Samples used	Features	Task	Source
Australian	690	14	Credit classification	[35]
Pima	768	8	Medical classification	[35]
Haberman	306	3	Medical classification	[35]
Sonar	208	60	Binary classification	[35]
Votes	435	16	Binary classification	[35]
German	1000	20	Credit classification	[35]
Diabetes	520	16	Medical classification	[35]
Wdbc	569	30	Medical classification	[35]
Ionosphere	351	34	Binary classification	[35]
Med-Breast[a]	546	784	Medical image classification	[36]
Med-Pneumonia[a]	4708	784	Medical image classification	[36]
Parkinson	756	754	Medical classification	[35]
Darwin	174	451	Medical classification	[35]
Spambase	4601	57	Binary classification	[35]
Spect	267	22	Medical classification	[35]
QSAR	1055	41	Biodegradation classification	[35]
Mammographic	961	5	Medical classification	[35]
Banknote Authentication	1372	4	Binary classification	[35]
Musk2	6598	166	Molecular classification	[35]

[a]The current MATLAB files use the training subsets of BreastMNIST and PneumoniaMNIST. Each 28×28 image is flattened into 784 features.

Table 2 Accuracy comparison on datasets without artificial noise (%).

Dataset	TWSVM	GBTWSVM	GBFTWSVM	RPDGBTWSVM	RPDTWSVM	RPDLDGBTWSVM
Australian	86.21 ± 2.85	86.50 ± 3.03	86.58 ± 2.68	86.68 ± 2.78	87.68 ± 2.16	$\mathbf{88.18 \pm 2.98}$
Pima	76.84 ± 2.55	76.94 ± 2.18	76.96 ± 2.15	77.10 ± 2.65	$\mathbf{78.61 \pm 2.55}$	77.50 ± 2.76
Haberman	74.43 ± 4.69	76.79 ± 4.55	76.93 ± 4.13	76.56 ± 4.56	76.62 ± 3.34	$\mathbf{77.30 \pm 5.48}$
Sonar	87.98 ± 5.20	87.68 ± 5.02	88.52 ± 4.96	88.40 ± 5.13	88.95 ± 4.64	$\mathbf{90.95 \pm 3.69}$
Votes	94.46 ± 2.19	94.36 ± 1.45	94.50 ± 1.38	94.04 ± 1.43	96.09 ± 1.64	$\mathbf{96.89 \pm 1.98}$
German	73.45 ± 1.78	76.02 ± 4.46	76.30 ± 4.34	76.00 ± 4.87	77.65 ± 1.88	$\mathbf{77.80 \pm 4.13}$
Diabetes	96.21 ± 1.81	95.23 ± 1.54	96.15 ± 1.34	96.26 ± 1.78	97.98 ± 1.31	$\mathbf{98.10 \pm 1.66}$
Wdbc	97.35 ± 1.70	96.13 ± 1.45	96.09 ± 1.54	96.66 ± 1.39	98.24 ± 1.37	$\mathbf{98.65 \pm 1.65}$
Ionosphere	93.01 ± 2.49	93.17 ± 3.85	93.44 ± 4.01	93.57 ± 1.43	95.28 ± 1.18	$\mathbf{96.10 \pm 3.26}$
Med-Breast	79.81 ± 3.49	78.93 ± 4.12	79.62 ± 4.34	79.58 ± 4.53	82.99 ± 3.31	$\mathbf{83.34 \pm 4.43}$
Med-Pneumonia	94.87 ± 0.90	94.38 ± 1.89	94.39 ± 2.01	94.04 ± 2.23	95.60 ± 0.50	$\mathbf{97.69 \pm 1.78}$
Parkinson	89.99 ± 2.71	85.56 ± 3.56	86.14 ± 4.67	88.24 ± 2.87	88.90 ± 2.21	$\mathbf{93.13 \pm 2.17}$
Darwin	86.16 ± 5.15	85.43 ± 5.76	86.13 ± 5.45	85.89 ± 4.99	86.36 ± 4.87	$\mathbf{89.71 \pm 4.35}$
Spambase	93.01 ± 0.87	92.45 ± 1.02	92.96 ± 0.89	92.03 ± 1.03	92.78 ± 0.87	$\mathbf{93.02 \pm 0.98}$
Spect	80.43 ± 4.98	82.25 ± 4.71	82.30 ± 4.43	82.67 ± 4.12	83.25 ± 4.71	$\mathbf{85.36 \pm 4.31}$
QSAR	85.55 ± 2.77	84.89 ± 2.89	86.83 ± 2.45	85.84 ± 2.12	87.97 ± 2.13	$\mathbf{88.64 \pm 1.79}$
Mammographic	79.11 ± 3.43	81.23 ± 2.76	82.52 ± 2.16	81.34 ± 3.24	81.34 ± 1.98	$\mathbf{85.83 \pm 2.10}$
Banknote Authentication	99.56 ± 0.57	99.34 ± 0.36	99.78 ± 0.03	97.32 ± 1.83	$\mathbf{99.99 \pm 0.01}$	$\mathbf{99.99 \pm 0.02}$
Musk2	99.24 ± 0.17	94.78 ± 2.34	97.17 ± 1.02	97.01 ± 0.45	97.65 ± 2.34	$\mathbf{99.32 \pm 0.02}$
Average	87.77	87.48	88.07	87.85	89.15	**90.39**

The proposed method also performs well on the high-dimensional datasets. It obtains 93.13% on Parkinson with 754 features, 89.71% on Darwin with 451 features, and 99.32% on Musk2 with 166 features. The local-density granular-ball representation therefore retains useful classification information after sample compression.

The proposed method does not rank first on Pima. On Pima, RPDLDGBTWSVM obtains 77.50%, while RPDTWSVM obtains 78.61%. The difference is 1.11 percentage points. The result shows that granular-ball representation is not always beneficial on a low-dimensional dataset.

On Banknote Authentication, RPDTWSVM and RPDLDGBTWSVM both obtain 99.99%. On Spambase, the difference between RPDLDGBTWSVM and TWSVM is only 0.01 percentage points. These datasets are already easy to classify. The available improvement is limited.

The detailed evaluation results of RPDLDGBTWSVM are reported in Table 3. Accuracy alone does not describe class-specific performance. F1-score and G-mean are more informative when class distributions are imbalanced.

Table 3 Detailed classification results of RPDLDGBTWSVM on datasets with complete class-specific records.

Dataset	Accuracy (%)	Precision	Recall	Specificity	F1-score	G-mean
Australian	88.18 ± 2.98	0.8158	0.7569	0.8697	0.7834	0.8103
Pima	77.50 ± 2.76	0.7839	0.8977	0.5293	0.8364	0.6872
Haberman	77.30 ± 5.48	0.4300	0.2011	0.9266	0.2523	0.3359
Sonar	90.95 ± 3.69	0.9284	0.9151	0.9130	0.9167	0.9112
Votes	96.89 ± 1.98	0.9652	0.9587	0.9399	0.9617	0.9489
German	77.80 ± 4.13	0.5649	0.4347	0.8551	0.4846	0.6067
Diabetes	98.10 ± 1.66	0.9695	0.9587	0.9813	0.9634	0.9697
Wdbc	98.65 ± 1.65	0.9710	0.9971	0.9441	0.9838	0.9701
Ionosphere	96.10 ± 3.26	0.9411	0.8592	0.9722	0.8971	0.9136
Med-Breast	83.34 ± 4.43	0.8387	0.9182	0.5442	0.8754	0.7033
Med-Pneumonia	97.69 ± 1.78	0.9717	0.9782	0.9742	0.9654	0.9733
Parkinson	93.13 ± 2.17	0.9248	0.9822	0.7692	0.9524	0.8680
Darwin	89.71 ± 4.35	0.8310	0.8728	0.8471	0.8492	0.8588
Spambase	93.02 ± 0.98	0.8534	0.9173	0.8960	0.8840	0.9065
Spect	85.36 ± 4.31	0.8972	0.8378	0.6426	0.8657	0.7300
QSAR	88.64 ± 1.79	0.9304	0.8386	0.8684	0.8818	0.8530
Mammographic	85.83 ± 2.10	0.7898	0.7748	0.8270	0.7804	0.7992
Banknote Authentication	99.99 ± 0.02	0.9984	1.0000	0.9987	0.9992	0.9993
Musk2	99.32 ± 0.02	0.9622	0.9880	0.9929	0.9748	0.9905

Sonar, Votes, Diabetes, Wdbc, Ionosphere, Banknote Authentication, and Musk2 obtain high F1-score and G-mean values. The model identifies both classes well on these datasets. Haberman has low Recall and low G-mean. Its overall accuracy should therefore be interpreted together with the class-specific metrics.

4.4 Running-Time Analysis

Table 4 reports the average time of one complete run. The time contains the operations required by each method. For RPDLDGBTWSVM, it includes granular-ball generation, relative density estimation, model training, and prediction.

RPDLDGBTWSVM obtains the best average running-time rank. Its running time is below 0.04 s on Australian, Pima, Haberman, Sonar, Votes, German, Diabetes, Wdbc, Ionosphere, and Darwin.

The arithmetic mean of RPDLDGBTWSVM is 4.172 s. This value is mainly affected by Med-Pneumonia and Musk2. Their running times are 62.120 s and 13.284 s, respectively. Both datasets require more distance calculations and relative density estimation.

On Spambase, RPDLDGBTWSVM uses 0.713 s. TWSVM uses 7.934 s, and RPDTWSVM uses 14.332 s. The granular-ball representation reduces the number of optimization inputs and produces a clear speed advantage on this dataset.

The results on Med-Pneumonia and Parkinson show a limitation. High feature dimension increases the cost of distance calculation, granular-ball generation, and relative density estimation. RPDLDGBTWSVM is therefore not the fastest method on every dataset.

Table 4 Average running time on datasets without artificial noise (s).

Dataset	TWSVM	GBTWSVM	GBFTWSVM	RPDGBTWSVM	RPDTWSVM	RPDLDGBTWSVM
Australian	0.054	0.360	0.428	0.489	0.185	0.017
Pima	0.054	0.800	0.741	0.786	0.373	0.018
Haberman	0.032	0.247	0.165	0.155	0.035	0.006
Sonar	0.014	0.090	0.111	0.109	0.031	0.018
Votes	0.028	0.145	0.128	0.115	0.073	0.025
German	0.238	0.567	0.648	0.578	0.162	0.035
Diabetes	0.031	0.097	0.118	0.115	0.108	0.019
Wdbc	0.038	0.096	0.092	0.122	0.138	0.027
Ionosphere	0.189	0.153	0.126	0.135	0.550	0.024
Med-Breast	0.511	0.501	0.473	0.522	0.420	0.702
Med-Pneumonia	4.654	3.108	2.785	3.132	60.760	62.120
Parkinson	0.089	0.623	0.551	0.535	0.587	1.231
Darwin	0.014	0.054	0.012	0.013	0.013	0.038
Spambase	7.934	2.783	3.231	2.988	14.332	0.713
Spect	0.012	0.215	0.157	0.231	0.051	0.189
QSAR	0.203	0.493	0.573	0.542	0.756	0.207
Mammographic	0.387	0.654	0.703	0.640	0.542	0.205
Banknote Authentication	0.570	0.076	0.082	0.087	0.084	0.397
Musk2	21.446	2.650	2.780	2.010	85.422	13.284
Average time	1.921	0.722	0.732	0.700	8.664	4.172
Average rank	2.842	3.947	3.789	4.026	3.763	**2.632**

4.5 Robustness under Gaussian Feature Noise

Gaussian noise is added only to the training features. The testing data and all class labels remain unchanged. Let $\mathbf{X}$ be the training feature matrix. A random matrix $\mathbf{M}$ is generated by

$$M_{ij} \sim \mathcal{N}(0,1). \tag{95}$$

The noisy training matrix is

$$\widetilde{\mathbf{X}} = \mathbf{X} + t\frac{\|\mathbf{X}\|_F}{\|\mathbf{M}\|_F}\mathbf{M}, \tag{96}$$

where t controls the relative noise energy. The noise levels are

$$t \in \{0.05, 0.10, 0.20, 0.30\}. \tag{97}$$

All methods use the same training–testing partition and the same noisy training set in each repeated run. The comparison is therefore paired.

Table 5 Accuracy under different Gaussian feature-noise levels (%).

Dataset	t	TWSVM	GBTWSVM	GBFTWSVM	RPDGBTWSVM	RPDTWSVM	RPDLDGBTWSVM
Sonar	0.05	85.24 ± 6.43	84.76 ± 5.29	84.86 ± 5.23	84.76 ± 5.29	88.24 ± 5.43	$\mathbf{89.29 \pm 3.93}$
	0.10	84.52 ± 7.03	85.71 ± 4.76	86.53 ± 5.27	86.43 ± 5.27	87.24 ± 5.53	$\mathbf{89.76 \pm 4.21}$
	0.20	81.90 ± 6.66	83.10 ± 5.20	83.34 ± 4.52	82.14 ± 4.52	86.24 ± 5.87	$\mathbf{88.10 \pm 5.83}$
	0.30	81.67 ± 6.25	82.86 ± 9.44	82.57 ± 8.15	83.57 ± 9.15	85.74 ± 5.33	$\mathbf{85.95 \pm 5.77}$
Wdbc	0.05	96.58 ± 1.05	93.16 ± 3.55	93.77 ± 2.79	91.49 ± 2.71	$\mathbf{97.82 \pm 1.46}$	97.11 ± 1.10
	0.10	96.58 ± 1.34	95.09 ± 1.90	92.98 ± 4.07	92.98 ± 2.92	96.52 ± 1.46	$\mathbf{97.19 \pm 1.15}$
	0.20	96.23 ± 1.55	94.12 ± 2.07	93.16 ± 2.58	90.88 ± 3.59	95.12 ± 1.46	$\mathbf{97.46 \pm 1.46}$
	0.30	96.14 ± 1.56	92.81 ± 4.58	90.00 ± 3.31	90.35 ± 5.64	94.02 ± 1.46	$\mathbf{96.93 \pm 1.61}$
Parkinson	0.05	88.53 ± 1.87	83.71 ± 3.09	84.77 ± 2.56	83.64 ± 3.06	85.50 ± 1.91	$\mathbf{89.80 \pm 1.80}$
	0.10	87.08 ± 1.32	81.66 ± 2.25	81.52 ± 1.45	80.53 ± 4.35	84.44 ± 2.17	$\mathbf{87.75 \pm 1.75}$
	0.20	$\mathbf{86.82 \pm 2.37}$	80.53 ± 4.35	81.72 ± 2.17	81.79 ± 1.96	80.33 ± 3.72	85.70 ± 2.63
	0.30	$\mathbf{84.83 \pm 3.47}$	78.81 ± 5.68	80.20 ± 4.09	81.66 ± 2.25	74.04 ± 3.07	45.56 ± 4.41
Musk2	0.05	99.11 ± 0.13	92.74 ± 1.80	93.12 ± 2.46	93.30 ± 1.22	97.91 ± 0.27	$\mathbf{99.14 \pm 0.21}$
	0.10	98.78 ± 0.24	93.30 ± 1.31	91.68 ± 1.69	92.42 ± 1.73	97.86 ± 0.21	$\mathbf{98.88 \pm 0.25}$
	0.20	96.87 ± 0.80	90.98 ± 1.60	90.53 ± 2.11	90.78 ± 1.66	95.41 ± 0.85	$\mathbf{97.02 \pm 0.65}$
	0.30	92.83 ± 1.74	90.81 ± 1.76	90.95 ± 1.74	90.79 ± 1.79	$\mathbf{93.61 \pm 0.76}$	92.68 ± 1.62

RPDLDGBTWSVM obtains the highest accuracy in 12 of the 16 noise settings.

On Sonar, RPDLDGBTWSVM ranks first at all four noise levels. Its accuracy decreases from 89.29% at $t = 0.05$ to 85.95% at $t = 0.30$. The decrease is moderate. The model remains effective on this small and high-dimensional dataset.

On Wdbc, RPDLDGBTWSVM ranks second at $t = 0.05$. It ranks first at the other three noise levels. Its accuracy remains between 96.93% and 97.46%. The variation is small.

On Parkinson, RPDLDGBTWSVM ranks first at $t = 0.05$ and $t = 0.10$, and remains competitive at $t = 0.20$. Its accuracy decreases from 89.80% to 87.75% and 85.70% over these three noise levels. Recall remains high, changing from 0.9725 to 0.9690 and 0.9628, whereas Specificity decreases from 0.6841 to 0.6157 and 0.5569. The degradation under mild and moderate noise is therefore mainly associated with weaker recognition of the negative class.

At $t = 0.30$, the Parkinson accuracy decreases sharply to 45.56%. The average number of granular balls reaches 605.00, which equals the number of training samples. The granular-ball ratio is therefore 100%, and the actual compression ratio is 0%. Since the data partitions, paired noise samples, and program execution were verified, this result is treated as a limitation of the model rather than an implementation error. Severe feature perturbation destroys the local-density neighborhood structure and eliminates the compression effect. It may also destabilize the class-dependent density-ratio weights. Recall decreases to 0.5651, Specificity decreases to 0.1435, and G-mean decreases to 0.2790, indicating poor separation of both classes.

On Musk2, RPDLDGBTWSVM ranks first at $t = 0.05$ and $t = 0.20$. It ties with TWSVM at $t = 0.10$. RPDTWSVM ranks first at $t = 0.30$. RPDLDGBTWSVM still obtains 92.68% under the strongest noise level.

4.6 Granular-Ball Compression Analysis

Let n_{tr} be the number of training samples and m be the number of generated granular balls. The granular-ball ratio is defined as

$$\mathrm{GBR} = \frac{m}{n_{\mathrm{tr}}} \times 100\%. \tag{98}$$

The compression ratio is

$$\mathrm{CR} = \left(1 - \frac{m}{n_{\mathrm{tr}}}\right) \times 100\%. \tag{99}$$

A smaller granular-ball ratio and a larger compression ratio indicate stronger compression. Table 6 reports the average granular-ball number and the running-time components of RPDLDGBTWSVM.

The compression level changes with the data distribution. Spambase obtains the largest compression ratio of 91.62%. Mammographic, Haberman, Pima, German, and Banknote Authentication also obtain compression ratios close to or above 80%.

Sonar, Parkinson, and Darwin have compression ratios of 4.28%, 5.40%, and 2.01%, respectively. Most samples are retained as granular-ball representatives. These datasets have high-dimensional or complex local structures. Strong compression may remove useful boundary information. LDGBG therefore keeps a finer representation.

Table 6 Granular-ball compression and running-time results of RPDLDGBTWSVM on datasets with complete component records.

Dataset	Balls	GBR (%)	CR (%)	GB time (s)	Model time (s)	Total time (s)
Australian	165.30 ± 5.38	29.95 ± 0.97	70.05 ± 0.97	0.0133	0.0039	0.0172
Pima	122.70 ± 3.43	19.98 ± 0.56	80.02 ± 0.56	0.0147	0.0039	0.0187
Haberman	43.40 ± 2.22	17.71 ± 0.91	82.29 ± 0.91	0.0048	0.0018	0.0066
Sonar	158.90 ± 1.60	95.72 ± 0.96	4.28 ± 0.96	0.0126	0.0053	0.0180
Votes	277.70 ± 1.64	79.80 ± 0.47	20.20 ± 0.47	0.0167	0.0088	0.0255
German	158.80 ± 5.07	19.85 ± 0.63	80.15 ± 0.63	0.0278	0.0073	0.0351
Diabetes	174.60 ± 6.22	41.97 ± 1.50	58.03 ± 1.50	0.0145	0.0052	0.0197
Wdbc	261.70 ± 5.72	57.52 ± 1.26	42.48 ± 1.26	0.0198	0.0077	0.0275
Ionosphere	209.10 ± 3.21	74.41 ± 1.14	25.59 ± 1.14	0.0181	0.0066	0.0247
Med-Breast	368.60 ± 3.50	84.35 ± 0.80	15.65 ± 0.80	0.6608	0.0412	0.7021
Parkinson	572.30 ± 2.98	94.60 ± 0.49	5.40 ± 0.49	1.4913	0.0858	1.5771
Darwin	136.20 ± 1.23	97.99 ± 0.88	2.01 ± 0.88	0.0315	0.0068	0.0383
Spambase	308.60 ± 8.50	8.38 ± 0.23	91.62 ± 0.23	0.4939	0.0193	0.5132
Spect	184.60 ± 2.46	86.26 ± 1.15	13.74 ± 1.15	0.0156	0.0055	0.0211
QSAR	671.70 ± 4.45	79.59 ± 0.53	20.41 ± 0.53	0.1142	0.0935	0.2077
Mammographic	121.60 ± 4.43	15.81 ± 0.58	84.19 ± 0.58	0.0161	0.0045	0.0205
Banknote Authentication	218.00 ± 5.64	19.85 ± 0.51	80.15 ± 0.51	0.0327	0.0070	0.0397
Musk2	2718.20 ± 13.36	51.50 ± 0.25	48.50 ± 0.25	11.0960	2.1886	13.2847

The number of granular balls is not fixed in advance. It changes with sample distribution, feature dimension, and local density. This adaptive process avoids forced over-compression.

4.7 Parameter Sensitivity

RPDLDGBTWSVM contains the penalty parameters c_1 and c_2, the Gaussian kernel parameter σ, and the granularity coefficient g. The penalty parameters control the cost of classification errors. The kernel parameter determines the nonlinear neighborhood scale. The coefficient g controls the construction of granular balls.

The penalty parameters are set as $c_1 = c_2$. Their values and σ are searched over

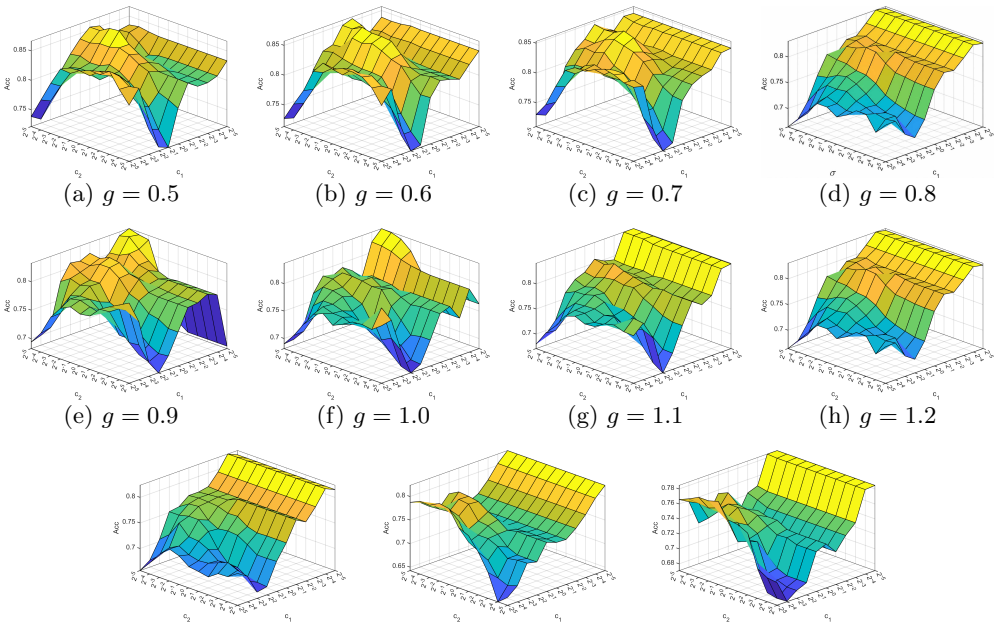
$$c_1 = c_2,\ \sigma \in \left\{2^{-5}, 2^{-4}, \ldots, 2^{4}, 2^{5}\right\}.$$

The coefficient g is varied from 0.5 to 1.5 with an interval of 0.1. For each fixed value of g, the mean classification accuracy obtained under different combinations of $c_1 = c_2$ and σ is displayed using a three-dimensional surface.

Figure 3 shows that the classification accuracy changes with g, $c_1 = c_2$, and σ. The highest mean accuracy is obtained when $g = 0.5$. Under this value, the surface contains a connected high-accuracy region rather than a single isolated peak. Several adjacent combinations of $c_1 = c_2$ and σ produce similar results. The model is therefore not highly sensitive to small parameter changes near the optimum.

The position and width of the high-accuracy region change as g increases. Larger values of g do not consistently improve the classification accuracy. Some surfaces become smoother, but their maximum accuracy is lower than that obtained at $g = 0.5$. This result indicates that excessive granularity adjustment may weaken the preservation of local class structure.

For the QSAR dataset, $g = 0.5$ is selected. The final values of $c_1 = c_2$ and σ are determined from the maximum point of the corresponding surface. The parameter selection considers both the maximum accuracy and the stability of its neighboring region.



(a) $g = 0.5$ (b) $g = 0.6$ (c) $g = 0.7$ (d) $g = 0.8$

(e) $g = 0.9$ (f) $g = 1.0$ (g) $g = 1.1$ (h) $g = 1.2$

(i) $g = 1.3$ (j) $g = 1.4$ (k) $g = 1.5$

Fig. 3 Parameter sensitivity of RPDLDGBTWSVM on the QSAR dataset.

4.8 Statistical Comparison

The Friedman test is used to compare the six methods on the 19 datasets with complete accuracy results [37]. The mean accuracy over ten runs is used for ranking. Rank 1 is assigned to the method with the highest accuracy on each dataset. Tied methods receive the average rank. The Nemenyi test is applied after the Friedman null hypothesis is rejected.

The average rank of the jth method is defined as

$$R_j = \frac{1}{N} \sum_{i=1}^{N} r_i^{(j)}, \tag{100}$$

where $N = 19$, and $r_i^{(j)}$ is the rank of the jth method on the ith dataset.

Table 7 reports the average ranks and the Nemenyi comparisons with RPDLDGBTWSVM.

RPDLDGBTWSVM obtains the lowest average rank of 1.079. RPDTWSVM ranks second with an average rank of 2.316. GBTWSVM has the largest average rank of 5.211.

Table 7 Average ranks and Nemenyi comparisons with RPDLDGBTWSVM.

Method	Average rank	Rank difference	Significant
RPDLDGBTWSVM	**1.079**	0.000	–
RPDTWSVM	2.316	1.237	No
GBFTWSVM	3.684	2.605	Yes
TWSVM	4.316	3.237	Yes
RPDGBTWSVM	4.395	3.316	Yes
GBTWSVM	5.211	4.132	Yes

The accuracy table contains tied results. A tie correction is therefore applied to the Friedman statistic. The correction factor is

$$C = 1 - \frac{\sum_{i=1}^{N} \sum_{h=1}^{G_i} \left(t_{ih}^3 - t_{ih}\right)}{N\left(k^3 - k\right)}, \tag{101}$$

where G_i is the number of tied groups on the ith dataset, and t_{ih} is the number of methods in the hth tied group.

The tie-corrected Friedman statistic is

$$\chi_F^2 = \frac{12N}{Ck(k+1)} \left[\sum_{j=1}^{k} R_j^2 - \frac{k(k+1)^2}{4} \right], \tag{102}$$

where $k = 6$. The calculated values are

$$C = 0.996992, \qquad \chi_F^2 = 63.650, \qquad p = 2.13 \times 10^{-12}. \tag{103}$$

The null hypothesis is rejected at the 0.05 significance level. The six methods have different overall classification performance.

The critical difference of the Nemenyi test is

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}, \tag{104}$$

where $q_{0.05} = 2.850$. For $k = 6$ and $N = 19$, the critical difference is

$$CD = 1.730. \tag{105}$$

The rank difference between RPDLDGBTWSVM and RPDTWSVM is 1.237. This value is smaller than the critical difference. The difference between these two methods is not statistically significant.

The rank differences between RPDLDGBTWSVM and GBFTWSVM, TWSVM, RPDGBTWSVM, and GBTWSVM are 2.605, 3.237, 3.316, and 4.132, respectively. All

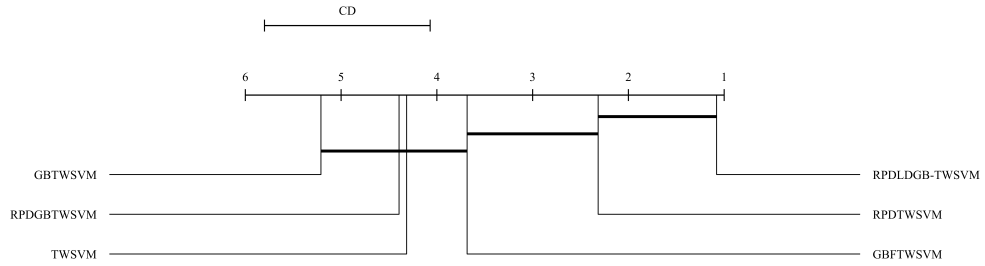


Fig. 4 Critical-difference diagram of the six compared methods.

four values are larger than the critical difference. RPDLDGBTWSVM is significantly better than these four methods at the 0.05 level.

Figure 4 presents the updated critical-difference diagram. Methods connected by the same horizontal line do not have a significant rank difference.

5 Conclusion

This paper proposes a new granular-ball-level data-weighting model, termed RPDLDGBTWSVM. The model selects real samples with high local density as granular-ball centers. KLIEP is then used to estimate class-dependent density ratios at these centers and assign larger weights to granular balls with more stable class distributions. The granular-ball radii are also incorporated into the constraints of the two TWSVM subproblems.

Theoretical analysis shows that, under radius control, the center-level RPD-weighted loss approximates the corresponding point-level weighted loss. Under the stated conditions, both weighted subproblems remain convex and feasible. Experiments on 19 datasets show that RPDLDGBTWSVM achieves the highest or tied-highest accuracy on 18 datasets, with an average accuracy of 90.39%. The proposed method also obtains the best average running-time rank. The Friedman–Nemenyi test shows significant differences between RPDLDGBTWSVM and TWSVM, GBTWSVM, GBFTWSVM, and RPDGBTWSVM. The rank difference between RPDLDGBTWSVM and RPDTWSVM is not statistically significant.

The experimental results further show that RPDLDGBTWSVM is effective for large datasets with complex class boundaries and local cluster structures. However, two limitations remain. For high-dimensional datasets with weak granular-ball compression, the costs of local-density calculation and density-ratio estimation increase. Severe feature noise may destroy the local neighborhood structure and cause the granular-ball representation to degenerate toward point-level input. This phenomenon is observed on the Parkinson dataset at $t = 0.30$. Future work will investigate adaptive granularity control, more stable density-ratio weighting, and efficient extensions for large-scale and severely corrupted data.

Declarations

The authors declare that they have no conflict of interest.

References

[1] Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning **20**, 273–297 (1995)

[2] Jayadeva, Khemchandani, R., Chandra, S.: Twin support vector machines for pattern classification. IEEE Transactions on Pattern Analysis and Machine Intelligence **29**(5), 905–910 (2007)

[3] Kumar, M.A., Gopal, M.: Least squares twin support vector machines for pattern classification. Expert Systems with Applications **36**(4), 7535–7543 (2009)

[4] Shao, Y.H., Zhang, C.H., Wang, X.B., Deng, N.Y.: Improvements on twin support vector machines. IEEE Transactions on Neural Networks **22**(6), 962–968 (2011)

[5] Peng, X.: TPMSVM: A novel twin parametric-margin support vector machine for pattern recognition. Pattern Recognition **44**(10–11), 2678–2692 (2011)

[6] Tanveer, M., Rajani, T., Rastogi, R., Shao, Y.H., Ganaie, M.A.: Comprehensive review on twin support vector machines. Annals of Operations Research **339**, 1223–1268 (2024)

[7] Zheng, X., Zhang, L., Yan, L.: CTSVM: A robust twin support vector machine with correntropy-induced loss function for binary classification problems. Information Sciences **559**, 22–45 (2021)

[8] Si, Q., Yang, Z., Ye, J., Pan, R.: Symmetric LINEX loss twin support vector machine for robust classification and its fast iterative algorithm. Neural Networks **168**, 143–160 (2023)

[9] Xu, J., Wang, H., Zhang, L., Wen, S.: Robust twin depth support vector machine based on average depth. Knowledge-Based Systems **274**, 110627 (2023)

[10] Yuan, C., Zhou, C., Pan, H., Arvin, F., Peng, J., Li, H.: Robust least squares twin bounded support vector machine with a generalized correntropy-induced metric. Information Sciences **699**, 121798 (2025)

[11] Si, Q., Yang, Z., Zhao, J., Chen, R.: A kernel-free quadratic surface twin support vector machine with capped L_1-norm distance metric for robust classification. Applied Soft Computing **180**, 113317 (2025)

[12] Li, Z., Zhang, P.: Affinity-based elastic net intuitionistic fuzzy twin support vector machines. International Journal of Machine Learning and Cybernetics **15**, 2439–2455 (2024)

[13] Qiu, J., Xie, J., Zhang, D., Zhang, R., Lin, M.: A fuzzy twin support vector machine based on dissimilarity measure and its biomedical applications. International Journal of Fuzzy Systems **26**, 2750–2766 (2024)

[14] Gupta, D., Hazarika, B.B., Gupta, U., Pedrycz, W.: A robust fuzzy twin support vector machine with kernel-target alignment for binary classification. Engineering Applications of Artificial Intelligence **161**, 112189 (2025)

[15] Wang, X., Wu, G., Hao, G., Zhang, Z.: A novel fuzzy twin support vector machine using mass-based dissimilarity measure. Knowledge and Information Systems **67**, 4233–4300 (2025)

[16] Li, Y.: Selecting training points for one-class support vector machines. Pattern Recognition Letters **32**(11), 1517–1522 (2011)

[17] Xia, S., Xiong, Z., Luo, Y., Dong, L., Zhang, G.: Relative density based support vector machine. Neurocomputing **149**, 1424–1432 (2015)

[18] Yu, H., Sun, C., Yang, X., Zheng, S., Zou, H.: Fuzzy support vector machine with relative density information for classifying imbalanced data. IEEE Transactions on Fuzzy Systems **27**(12), 2353–2367 (2019)

[19] Xia, S., Huang, L., Wang, G., Gao, X., Shao, Y., Chen, Z.: An adaptive and general model for label noise detection using relative probabilistic density. Knowledge-Based Systems **239**, 107907 (2022)

[20] Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Transactions on Information Theory **13**(1), 21–27 (1967)

[21] Sugiyama, M., Suzuki, T., Kanamori, T.: Density Ratio Estimation in Machine Learning. Cambridge University Press, Cambridge (2012)

[22] Sugiyama, M., Nakajima, S., Kashima, H., Bünau, P., Kawanabe, M.: Direct importance estimation with model selection and its application to covariate shift adaptation. In: Advances in Neural Information Processing Systems 20 (2007)

[23] Kanamori, T., Hido, S., Sugiyama, M.: A least-squares approach to direct importance estimation. Journal of Machine Learning Research **10**, 1391–1445 (2009)

[24] Yamada, M., Suzuki, T., Kanamori, T., Hachiya, H., Sugiyama, M.: Relative density-ratio estimation for robust distribution comparison. Neural Computation **25**(5), 1324–1370 (2013)

[25] Nguyen, X.L., Wainwright, M.J., Jordan, M.I.: Estimating divergence functionals and the likelihood ratio by convex risk minimization. IEEE Transactions on Information Theory **56**(11), 5847–5861 (2010)

[26] Menon, A.K., Ong, C.S.: Linking losses for density ratio and class-probability estimation. In: Proceedings of the 33rd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 48, pp. 304–313.

PMLR, New York, New York, USA (2016)

[27] Xia, S., Wang, G., Gao, X., Lian, X.: Granular ball computing classifiers for efficient, scalable and robust learning. Information Sciences **483**, 136–152 (2019)

[28] Xia, S., Lian, X., Wang, G., Gao, X., Chen, J., Peng, X.: GBSVM: An efficient and robust support vector machine framework via granular-ball computing. IEEE Transactions on Neural Networks and Learning Systems **36**(5), 9253–9267 (2025)

[29] Quadir, A., Sajid, M., Tanveer, M.: Granular ball twin support vector machine. IEEE Transactions on Neural Networks and Learning Systems **36**(7), 12444–12453 (2025)

[30] Lang, G., Zhao, L., Miao, D., Ding, W.: Granular-ball computing based fuzzy twin support vector machine for pattern classification. IEEE Transactions on Fuzzy Systems **33**(7), 2148–2160 (2025)

[31] Ganaie, M.A., Ahire, V.: Granular ball twin support vector machine with universum data. Neural Networks **193**, 107974 (2026)

[32] Grewal, N., Gupta, S.K., Kumar, S.: Granular ball twin bounded support vector machine with generalized pinball loss. Information Sciences, 123557 (2026). Available online 27 April 2026

[33] Liu, F., Zhang, Q., Xia, S., Xie, Q., Liao, W., Zhang, S.: A granular-ball generation method based on local density for classification. Information Sciences **717**, 122295 (2025)

[34] Lin, C.F., Wang, S.D.: Fuzzy support vector machines. IEEE Transactions on Neural Networks **13**(2), 464–471 (2002)

[35] Kelly, M., Longjohn, R., Nottingham, K.: The UCI Machine Learning Repository. https://archive.ics.uci.edu. University of California, Irvine (2023)

[36] Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., Ni, B.: Medmnist v2: A large-scale lightweight benchmark for 2d and 3d biomedical image classification. Scientific Data **10**(1), 41 (2023)

[37] Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research **7**, 1–30 (2006)