Chemometrics and Intelligent Laboratory Systems 188 (2019) 70–78

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics

A multi-class classifier based on support vector hyper-spheres for steel plate surface defects

Rongfen Gong, Maoxiang Chu *, Yonghui Yang **, Yao Feng

School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, China

ARTICLE INFO

Keywords:
Steel plate surface defects
Support vector machine
Pinball loss
Local within-class sample density

ABSTRACT

In order to improve classification accuracy and efficiency for steel plate surface defects, a novel multi-class classifier termed as support vector hyper-spheres with insensitivity to noise (INSVHs) is proposed in this paper. On one hand, the INSVHs classifier introduces pinball loss to reduce its sensitivity to noise around decision boundary. On the other hand, the INSVHs classifier reduces the adverse effect of label noise and enhances the beneficial effect of important samples by adding local within-class sample density weight. Moreover, the INSVHs classifier builds an independent hyper-sphere for each type of defect to improve classification efficiency. The testing experiments for steel plate surface defects show that our INSVHs classifier is insensitive to noise and improves classification accuracy and efficiency.

1. Introduction

Steel plate is the main product of steel enterprises, and its surface quality affects directly the performance of steel products. So, the surface quality monitor system is always adopted in the production line of steel plate [1–8]. There may be many problems in the process of steel plate processing, casting and rolling. These problems will lead to many types of defects on steel plate surface. Then, in the surface quality monitor system, the defects classification becomes an important step. In recent years, more and more scholars are devoting themselves to the study of defects classification. Dupont et al. [1] used *k*-nearest neighbor classifier to realize the defects classification for cold-rolled steel products. Yan et al. [2] classified six kinds of steel surface defects by using a weak classifier with stacked structure and adaptive enhancement algorithm. Pernkopf [3] proposed a new Bayesian network classifier based on the theory of augmented naive Bayes classifier, which was used to recognize crude steel block surface defects. Peng et al. [4] proposed a back propagation neural network algorithm, which improved the classification performance for steel plate surface defects by modifying error function. Amid et al. [5] built a multi-class classifier with decision tree and support vector machine (SVM) to realize the multi-class classification for steel surface defects. Chen et al. [6] proved that the SVM classifier with multi-kernel learning was fit for real-time steel surface monitor system. Chu et al. [7] proposed a multi-density twin support vector machine classifier, which also improved the classification accuracy and efficiency for steel plate surface defects. Based on least square twin support vector machine, Chu et al. [8] solved the multi-class classification for steel plate surface defects with large-scale, unbalanced and corrupted samples by introducing error variable contribution and sample weight parameters.

High performance classifiers were used to realize the multi-class classification for steel plate surface defects in Refs. [1–8]. Standard and improved SVMs were adopted and obtained satisfactory results in Refs. [5–8]. However, these methods seldom consider noise in dataset. In reality, there are inevitably noise in defects dataset. These noise will affect classification results of standard and improved SVM seriously. The sample weight was used to restrain the adverse effect of noise in Refs. [7, 8]. This method is useful for label noise far from the decision boundary but is helpless for noise around the decision boundary.

In order to solve the problem that SVM [9] is sensitive to noise around the decision boundary, Huang et al. [10] proposed a novel pinball loss support vector machine (Pin-SVM) classifier. Pin-SVM holds the idea of solving a large quadratic programming problem (QPP). However, it changes the idea of maximizing the shortest distance between two classes into maximizing the quantile distance with pinball loss, which reduces its sensitivity to noise. On the other hand, many excellent extended SVM classifiers have been proposed and used widely in recent years [11–15]. Peng et al. [14] proposed twin support vector hyper-sphere (TSVH) based on twin support vector machine (TSVM) [12] and support vector data

* Corresponding author.
** Corresponding author.
E-mail addresses: fx_gong@hotmail.com (R. Gong), chu52_2004@163.com (M. Chu), yangyh2636688@163.com (Y. Yang), fyasdf1222@hotmail.com (Y. Feng).

https://doi.org/10.1016/j.chemolab.2019.03.010
Received 20 November 2018; Received in revised form 16 March 2019; Accepted 18 March 2019
Available online 20 March 2019

description (SVDD) [13]. TSVH holds the idea of maximizing the shortest distance between two classes. However, it constructs two hyper-spheres by solving two small QPPs. So, TSVH has the merits of SVM, TSVM and SVDD, which obtains better classification result for pattern recognition.

Based on the advantage of TSVH and Pin-SVM, a novel multi-class classification method for steel plate surface defects is proposed in this paper. It builds a multi-class classifier termed as support vector hyper-spheres with insensitivity to noise (INSVHs). The INSVHs classifier has the structure of hyper-sphere in TSVH and pinball loss in Pin-SVM. So, the INSVHs classifier constructs multiple hyper-spheres by solving corresponding QPPs and is insensitive to noise just like Pin-SVM. Moreover, the local within-class sample density weight is introduced into INSVHs. The sample density weight not only is helpful to restrain the adverse effect brought by isolated samples and label noise, but also can strengthen the beneficial effect brought by important samples to decision boundary. Beyond that, the INSVHs classifier can realize multi-class classification easily. In the end, the novel INSVHs classifier is used to classify six types of steel plate surface defects. The experimental results show that the INSVHs classifier can restrain the adverse effect of noise and improve the classification accuracy and efficiency.

2. Related work

2.1. Defect dataset

Before studying steel plate surface defects classification method, it is necessary to build a complete defects dataset. A large number of defects images are firstly collected from our national large steel mills with surface defects detection system equipped on steel production lines. The system collects steel plate surface images with CCD vision machine and selects defects images with detection method for region of defects [16, 17]. In this paper, six types of surface defects images are collected, such as hole, scratch, bruise, wave, scarring and scale. Typical images of them are shown in Fig. 1.

In order to obtain defects dataset for training and testing, those defects images should be processed in advance, such as preprocessing, segmentation and feature extraction. Preprocessing mainly realizes the enhancement and restoration for defects images. Segmentation can determine the region of defect. Switching bilateral filter algorithm [18] and watersheds algorithm [19] are used to realize preprocessing and segmentation. Feature extraction is used to describe the region of defect. For better description, many kinds of features are adopted, such as gray, shape, texture and invariant moment. The feature extraction method can refer to reference [20]. Finally, the region of defect is described with 38 features. So, each sample vector is composed of 38 elements.

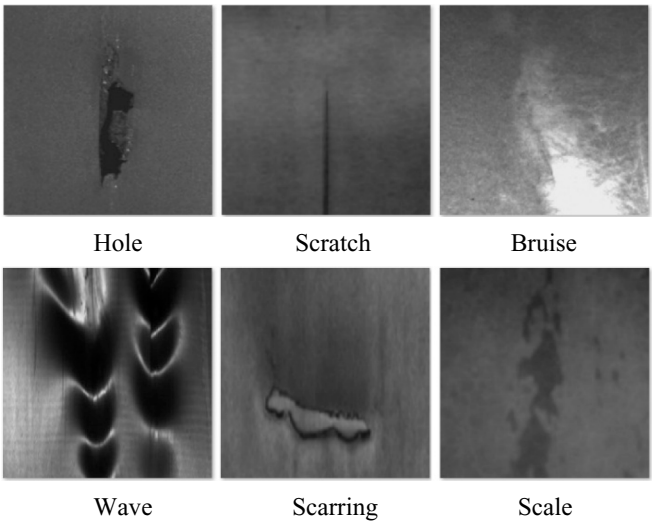

Fig. 1. Typical images for six types of defects.

2.2. Multi-class classification model

The multi-class classifier for steel plate surface defects is based on the novel INSVHs model. Firstly, the sample set is assumed as $X = \{X^n | X^n \in \Re^{m^n \times d},\ n = 1, 2, \cdots, N\}$. There are N types of defects and the total number of samples is m in X, where $m = m^1 + \cdots + m^N$. For binary classification, the two hyper-spheres of INSVHs model are built just like TSVH.

$$\Omega^+ : \quad ||\psi(\boldsymbol{x}) - \boldsymbol{C}^+||^2 \le \left(R^+\right)^2, \Omega^- : \quad ||\psi(\boldsymbol{x}) - \boldsymbol{C}^-||^2 \le \left(R^-\right)^2, \tag{1}$$

where $\psi(\cdot)$ is the feature space mapping function. $\boldsymbol{C}^+$ and $\boldsymbol{C}^-$ are the centers of the hyper-spheres Ω^+ and Ω^- respectively. R^+ and R^- are the radii of Ω^+ and Ω^- respectively. Then, for N types of defects, the INSVHs model only needs to build N hyper-spheres. These hyper-spheres are defined as follows:

$$\Omega^n : \quad \left|\left|\psi(\boldsymbol{x}) - \boldsymbol{C}^n\right|\right|^2 \le (R^n)^2, \ \ n = 1, 2, \cdots, N, \tag{2}$$

where $\boldsymbol{C}^n$ is the center of Ω^n and R^n is the radius of Ω^n. In fact, the INSVHs classifier is made up of N sub-models called as INSVH-n, where each INSVH-n meets a QPP. And each QPP can be optimized to obtain a hyper-sphere. Finally, the multi-class classification model based on INSVHs is shown in Fig. 2. It can be seen that the INSVHs classifier has the structure of hyper-sphere in TSVH. In addition, the INSVHs classifier also has the structure of pinball loss in Pin-SVM. So, the INSVHs classifier is insensitive to noise just like Pin-SVM. Moreover, the local within-class sample density weight is introduced into INSVHs.

The model of INSVHs includes two parts: training and testing. For training part, the INSVHs model adopts one-against-rest strategy to realize multi-class classification. INSVH-n reckons X^n as training dataset for class $+1$ and reckons the others $X - X^n$ as training dataset for class -1. Then the classification hyper-sphere Ω^n with center C^n and radius R^n can be obtained by solving QPP. This process adopts one-against-rest strategy. In this process, INSVH-n only solves a small QPP. Using the same strategy, TSVH needs to solve two small QPPs and Pin-SVM needs to solve a large QPP. Finally, INSVHs classifier obtains N classification hyper-spheres by solving N small QPPs. For testing part, the INSVHs classifier builds directly the multi-class classification decision function according to these hyper-spheres. The multi-class classification decision rule of INSVHs is that a new sample $\boldsymbol{x}$ belongs to the farthest hyper-sphere. So, the decision function for INSVHs is defined as follows:

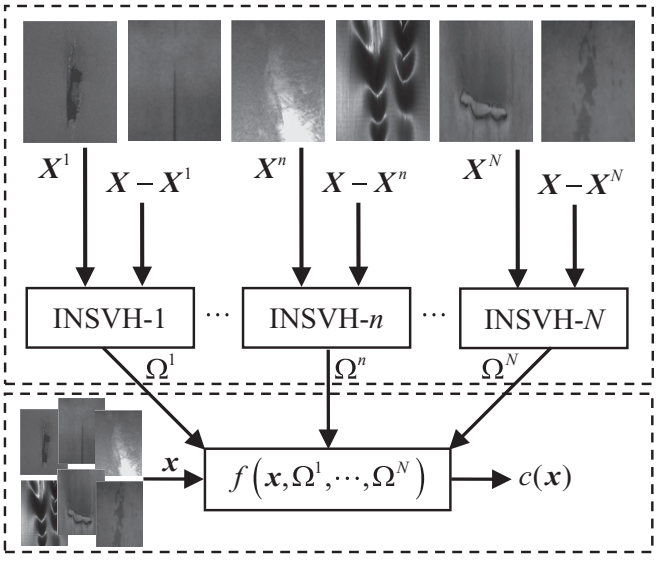

Fig. 2. Multi-class classification model based on INSVHs.

$$f(\boldsymbol{x}, \Omega_1, \cdots, \Omega_N) = \arg\max_n \left\{ \frac{||\psi(\boldsymbol{x}) - \boldsymbol{C}^n||^2}{(R^n)^2} \middle| n = 1, 2, \cdots, N \right\}. \quad (3)$$

3. INSVHs algorithm

The samples matrices for class +1 and class −1 are assumed as $\boldsymbol{A} = [\boldsymbol{A}_1 \cdots \boldsymbol{A}_i \cdots \boldsymbol{A}_{m^+}]^T$ and $\boldsymbol{B} = [\boldsymbol{B}_1 \cdots \boldsymbol{B}_j \cdots \boldsymbol{B}_{m^-}]^T$ respectively, and $m = m^+ + m^-$.

3.1. Constraint condition of pinball loss

It can be seen from Ref. [14] that the TSVH classifier includes quadratic loss and hinge loss function. If we define

$$u^+ = \frac{||\psi(\boldsymbol{B}_j) - \boldsymbol{C}^+||^2}{(R^+)^2}, \quad u^- = \frac{||\psi(\boldsymbol{A}_i) - \boldsymbol{C}^-||^2}{(R^-)^2}, \quad (4)$$

then the hinge loss functions L_h^+ and L_h^- of TSVH can be represented as

$$L_h^+ = \max\left\{0, \ (R^+)^2(1-u^+)\right\}, L_h^- = \max\left\{0, \ (R^-)^2(1-u^-)\right\}. \quad (5)$$

L_h^+ and L_h^- are shown as Fig. 3(a). The TSVH classifier with hinge loss requires one class of samples should be distributed inside of hyper-sphere and the other classes of samples should be distributed outside of the hyper-sphere as much as possible. However, once there are some noise around the boundary of hyper-sphere, then the hyper-sphere will deviate from the optimal position. Based on TSVH model, the new binary INSVHs model can be obtained by introducing pinball loss function of Pin-SVM. Just like Pin-SVM, INSVHs is insensitive to noise. The reason is that the binary INSVHs model changes the boundary attribute of hyper-sphere in TSVH. It adjusts some samples from outside to inside of hyper-sphere to reduce the adverse effect of noise around decision boundary.

The new pinball loss functions L_τ^+ and L_τ^- of binary INSVHs can be defined as:

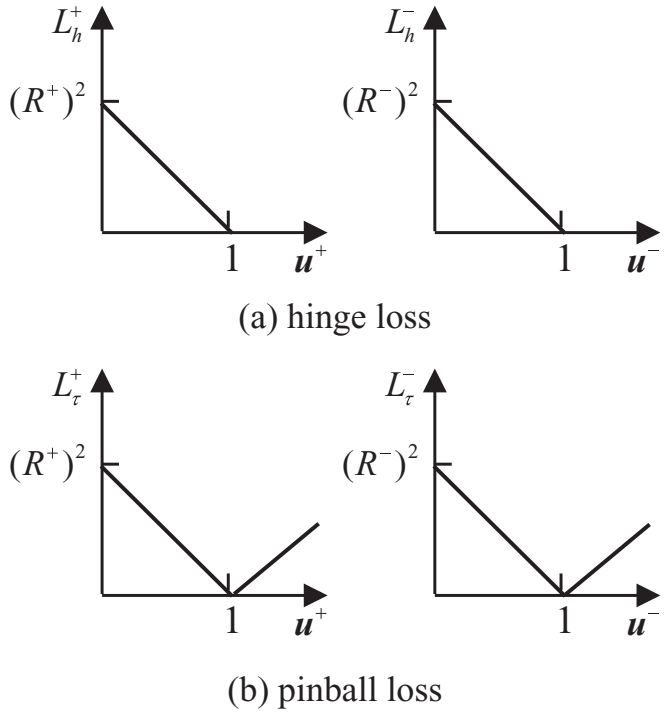

Fig. 3. Hinge loss and pinball loss.

$$L_\tau^+ = \begin{cases} (R^+)^2(1-u^+), & u^+ \le 1, \\ -\tau(R^+)^2(1-u^+), & u^+ > 1, \end{cases}$$
$$L_\tau^- = \begin{cases} (R^-)^2(1-u^-), & u^- \le 1, \\ -\tau(R^-)^2(1-u^-), & u^- > 1. \end{cases} \quad (6)$$

L_τ^+ and L_τ^- are shown as Fig. 3(b). If L_τ^+ and L_τ^- are replaced by L_h^+ and L_h^- in TSVH respectively, then the binary INSVHs model can be obtained. Its constraint condition can be represented as:

$$||\psi(\boldsymbol{B}_j) - \boldsymbol{C}^+||^2 \ge (R^+)^2 - \zeta_j, ||\psi(\boldsymbol{B}_j) - \boldsymbol{C}^+||^2 \le (R^+)^2 + \frac{1}{\tau}\zeta_j, \quad j = 1, 2, \cdots, m^-, \quad (7)$$

$$||\psi(\boldsymbol{A}_i) - \boldsymbol{C}^-||^2 \ge (R^-)^2 - \xi_i, ||\psi(\boldsymbol{A}_i) - \boldsymbol{C}^-||^2 \le (R^-)^2 + \frac{1}{\tau}\xi_i, \quad i = 1, 2, \cdots, m^+. \quad (8)$$

3.2. Local within-class sample density weight

For constraint condition (7), the binary INSVHs model mainly reduces the adverse effect of noise in class −1 rather than class +1. The isolated samples in class +1, especially the label noise far from decision boundary will affect the hyper-sphere Ω^+. In order to reduce the adverse effect of these samples, each sample can be added a corresponding weight. So, a novel sample weight calculation method is proposed in this section. It estimates the weight with local within-class sample density [21] for class +1. For constraint condition (8), the weight estimated with local within-class sample density for class −1 can also be obtained.

The following is the algorithm of local within-class sample density weight for class +1.

Step 1. in the mapping feature space, the Euclidean distance between any two samples in class +1 is calculated as:

$$\left(\widehat{d}_{i,i2}^+\right)^2 = \left|\left|\psi(\boldsymbol{A}_i) - \psi(\boldsymbol{A}_{i2})\right|\right|^2, \quad i, i2 = 1, 2, \cdots, m^+. \quad (9)$$

Step 2. according to $\left(\widehat{d}_{i,i2}^+\right)^2$ and predetermined k, the k-nearest neighbors set $(\Omega_k)_i^+$ in feature space for each sample $\boldsymbol{A}_i$ is determined.

Step 3. The mean radius $\overline{\delta}_i^+$ of $(\Omega_k)_i^+$, the total mean radius δ^+ and the relative Euclidean distance $\overline{d}_i^+$ between $\boldsymbol{A}_i$ and $(\Omega_k)_i^+$ are calculated as:

$$\overline{\delta}_i^+ = \frac{1}{k}\sum_{i2 \in (\Omega_k)_i^+} \widehat{d}_{i,i2}^+, \quad i = 1, 2, \cdots, m^+,$$
$$\delta^+ = \frac{1}{m^+}\sum_{i=1}^{m^+} \overline{\delta}_i^+, \quad (10)$$
$$\overline{d}_i^+ = \frac{1}{\delta^+}||\psi(\boldsymbol{A}_i) - \frac{1}{k}\sum_{i2 \in (\Omega_k)_i^+} \psi(\boldsymbol{A}_{i2})||^2.$$

Step 4. The density of $\psi(\boldsymbol{A}_i)$ in feature space can be calculated with the following formula.

$$d_i^+ = \frac{\delta^+}{\delta^+ + \overline{d}_i^+ \cdot \overline{\delta}_i^+}, \quad i = 1, 2, \cdots, m^+. \quad (11)$$

It can be seen that d_i^+ is constrained into [0,1]. When $\overline{\delta}_i^+ = 0$, d_i^+ reaches the maximum value 1 and all samples in $(\Omega_k)_i^+$ are clustered together. The bigger $\overline{\delta}_i^+$ is, the closer d_i^+ is to 0. And the samples in $(\Omega_k)_i^+$ are more dispersed. So, the bigger d_i^+ is, the more compacted the k-

nearest neighbors of A_i are. The above results show that the k-nearest neighbors with big d_i^+ are located in a clustered region. And A_i with big d_i^+ can be concerned as an important sample. On the contrary, the smaller d_i^+ is, the more dispersed the k-nearest neighbors of A_i are. Then, the importance of A_i reduces. So, d_i^+ reflects the distribution density of the k-nearest neighbors of A_i. Clearly, the density of the isolated sample especially label noise is small.

δ^+ means d_i^+ considers the whole distribution of sample set. In other words, for two sets with different distribution densities, the sample density can overcome the error caused by distribution density. $\overline{d}_i^+$ means d_i^+ considers the deviation between a sample and the center of its k-nearest neighbors. If there are two samples A_{i1} and A_{i2}, then d_{i1}^+ and d_{i2}^+ are equal. However, the distances between each of them and the center of k-nearest neighbors are different. It can be concerned that the closer the sample is to the center of k-nearest neighbors, the more compacted its k-nearest neighbors are. In order to solve this problem, d_i^+ is modified by $\overline{d}_i^+$. And $\overline{d}_i^+$ also considers the whole distribution of samples. In general, k can be chosen from 6 to 10 and is set as 8 in our experiment.

3.3. INSVHs formula

Because there are many types of steel plate surface defects, the defects recognition requires multi-class classification algorithm. It can be seen from Fig. 2 that the INSVHs model can easily realize multi-class classification. The binary INSVHs model builds two hyper-spheres: Ω^+ and Ω^-. Then, for N types of steel plate surface defects, the multi-class INSVHs model only needs to build N hyper-spheres.

According to Section 2.2, the hyper-sphere is Ω^n for the *n-th* type of defect and its training dataset can be described as:

$$\boldsymbol{A}^n = \{\boldsymbol{X}^o | o = 1, \cdots, n-1, n+1, \cdots, N\ \} \in \Re^{(m-m^n)\times d}, \boldsymbol{B}^n = \boldsymbol{X}^n \in \Re^{m^n \times d}. \tag{12}$$

The QPPs of multi-class INSVHs model can be defined as:

$$\begin{aligned} \min_{\boldsymbol{C}^n,\ R^n}\ & -p_r^n (R^n)^2 + p_e^n \sum_{j=1}^{m^n} \zeta_j + \frac{1}{2} \sum_{i=1}^{m-m^n} \xi_i \\ \text{s.t.}\quad & ||\psi\left(\boldsymbol{B}_j^n\right) - \boldsymbol{C}^n||^2 \geq (R^n)^2 - \zeta_j, \\ & ||\psi\left(\boldsymbol{B}_j^n\right) - \boldsymbol{C}^n||^2 \leq (R^n)^2 + \frac{1}{\tau}\zeta_j, \qquad n = 1, 2, \cdots, N, \\ & d_i^n\left(||\psi(\boldsymbol{A}_i^n) - \boldsymbol{C}^n||^2\right) = \xi_i, \\ & i = 1, 2, \cdots, m-m^n,\ \ j = 1, 2, \cdots, m^n, \end{aligned} \tag{13}$$

where $p_r^n > 0$ and $p_e^n > 0$ are penalty factors. d_i^n can be obtained with formula (11). The novel INSVHs model builds N QPPs in total. By optimizing each QPP, its corresponding hyper-sphere can be obtained. For the objective function of the *n-th* QPP, the first term is to maximize the radius of the hyper-sphere Ω^n. The term ensures the margin maximization between the two classes of samples. It can be seen from the first and the second constraint conditions that the second term of the objective function implies the minimization of pinball loss. On one hand, the pinball loss reduces the classifier's sensitivity to noise. On the other hand, minimizing pinball loss ensures its misclassification boundary is the same with that of hinge loss [10]. It can be seen from the third constraint condition that the third term in objective function minimizes the sum of squared distances between the center of hyper-sphere and samples in all classes except the *n-th* class. This makes the samples in other classes be close to the center of hyper-sphere Ω^n as much as possible, which minimizes the dispersion of samples. Moreover, the local within-class sample density d_i^n can weaken the adverse effect of isolated samples and label noise far away from decision boundary. When $\tau = 0$, the pinball loss function in INSVHs becomes hinge loss function. Then, the second constraint condition in (13) becomes $\zeta_j \geq 0$. When $\tau \neq 0$, τ can be adjusted to reduce the classifier's sensitivity to noise. Let us take the hyper-sphere of the *n-th* class to illustrate this question. The first constraint condition in (13) makes the *n-th* class of samples be outside of hyper-sphere Ω^n as much as possible. Once some noise is distributed near the boundary of Ω^n, the hyper-sphere will deviate from the optimal position. The second constraint condition in (13) adjusts the number of samples in the *n-th* class distributed inside or outside of Ω^n. The more samples in the *n-th* class are distributed inside of the hyper-sphere, the less adverse effect the noise has. So, the INSVHs model is insensitive to noise.

The hyper-spheres of TSVH and INSVHs are shown in Fig. 4 for one class of samples. Both of them are built for samples in $\boldsymbol{B}^n$("∘" in red) and have the large radius as much as possible. However, the center of hyper-sphere for TSVH is located in the region of $\boldsymbol{B}^n$, and the boundary of hyper-sphere (dotted curve in red) is close to the samples in $\boldsymbol{A}^n$("□" and "△" in blue). This means that the hyper-sphere is obtained by solving inequality constrained optimization problem for $\boldsymbol{A}^n$ and equality constrained optimization problem for $\boldsymbol{B}^n$. On the other hand, the center of hypersphere for INSVHs is located in the region of $\boldsymbol{A}^n$, and the boundary of hyper-sphere (solid curve in red) is constrained by the samples in $\boldsymbol{B}^n$. This hyper-sphere is obtained by solving inequality constrained optimization problem for $\boldsymbol{B}^n$ and equality constrained optimization problem for $\boldsymbol{A}^n$. It can be seen that TSVH costs much time to solve inequality constrained optimization problem for $\boldsymbol{A}^n$ with two classes of samples. However, INSVHs needs little time to solve the same problem for $\boldsymbol{B}^n$ with only one class of samples. Moreover, the hyper-sphere of TSVH is easily affected by the noise around the boundary in $\boldsymbol{A}^n$, but the hyper-sphere of INSVHs is insensitive to noise around the boundary.

The QPPs in (13) are convex quadratic programming problems. In Appendix A, we take the solution of QPPs in (13) as an example. Using the trick in Appendix A, we can determine $\boldsymbol{C}^n$ and R^n of Ω^n. Then, according to $\Omega^n (n = 1, 2, \cdots, N)$, the type of a new sample can be determined with the decision function (3).

4. Experiments

In order to testify the validity of the novel classification algorithm, SVM, Pin-SVM, TSVH and our INSVHs are compared in this section. For fairness, all classifiers are solved with quadratic programming function (Quadprog) in MATLAB. The penalty parameters for all classifiers are chosen from $\{2^t |\ t = -9, -8, \cdots, 8\}$, and the pinball loss parameter τ is chosen from $\{0.1,\ 0.2,\ 0.5,\ 1\}$. For nonlinear condition, the kernel function $K(,)$ adopts radial basis function and the kernel radius is chosen from $\{2^t |\ t = -10, -9, \cdots, 8\}$. The optimal parameters are obtained with grid searching from the combination of all parameters. Then we split the dataset into five subsets. And four of them are used to train the classifier and the remaining one is used to test the classifier. The training and testing tasks are repeated five times with cross validation method. The experimental results are determined by the average of five accuracies from cross validation. All experiments are done with MATLAB 7.11 on Windows 7 operating system with 3.2GHZ CPU and 4 GB memory. Six types of defects images are collected, which have been shown in Fig. 1. After a series of processing with the methods in Section 2.1, all defects images are divided into training dataset and testing dataset as Table 1.

The first experiment is done to show the theory of INSVHs model on 2-dimensional space. Firstly, two types of defects samples are randomly selected from training dataset and only 2-dimensional feature vectors are retained. Then, these 2-dimensional vectors are used to train the binary TSVH and INSVHs with linear kernel. The kernel function of linear classifiers is $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \boldsymbol{x}_1 \cdot \boldsymbol{x}_2$, and the parameter τ of INSVHs is set 0.5. Training results are shown in Fig. 5(a) and (b). It can be seen from Fig. 5(a) that the TSVH classifier builds two boundary hyper-spheres. For "+" in red, the red boundary hyper-sphere requires all "+" to be close to

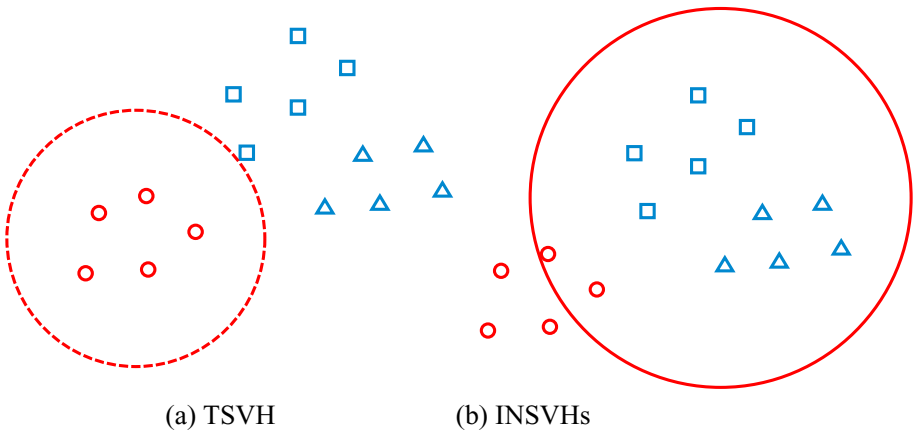

Fig. 4. The hyper-spheres of TSVH and INSVHs for one class of samples. Three classes of samples ("∘" in red, "□" in blue and "△" in blue). The hyper-spheres of class "∘" (dotted and solid curves in red).

Table 1
The number of samples for different types of defects.

Defects	The number of total samples	The number of training samples	The number of testing samples
Hole	590	413	177
Scratch	540	378	162
Bruise	560	392	168
Wave	550	385	165
Scarring	580	406	174
Scale	500	350	150

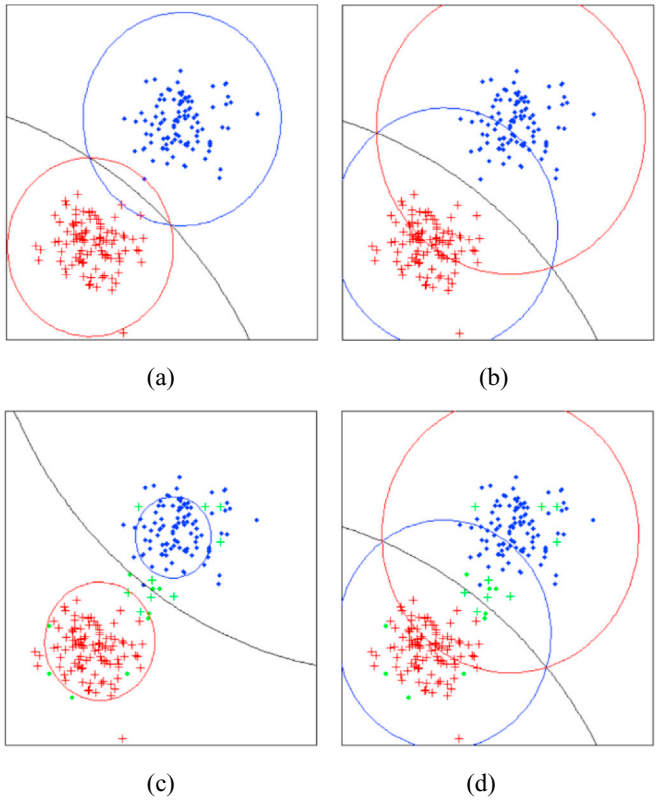

Fig. 5. Classification results of TSVH and INSVHs on 2-dimensional space. (a) and (c) for TSVH, (b) and (d) for INSVHs. Two classes of samples ("+" in red and "." in blue) and noise ("+" and "." in green). The hyper-sphere of class "+" (red curves), the hyper-sphere of class "." (blue curves), the decision boundary (black curves).

its center as much as possible, and requires "." in blue to be outside of it as much as possible. The hyper-sphere is controlled by "." on the boundary. If "." on the boundary is noise, the hyper-sphere is not optimal. It can be seen from Fig. 5(b) that the INSVHs classifier also builds two hyper-spheres. For "+" in red, the red hyper-sphere requires "." in blue to be close to the center as much as possible, while some samples with "+" distribute inside of the hyper-sphere, which means the pinball loss controls the samples inside or outside of the hyper-sphere with parameter τ. It can be seen from Fig. 5(a) and (b) that the decision boundaries are nearly the same, which means INSVHs and TSVH have the consistent classification results for standard samples. Fig. 5(c) and (d) show the results of INSVHs and TSVH for training dataset with noise. The corrupted samples are label noise far from decision boundary and around decision boundary, which are introduced artificially. It can be seen that the decision boundary of TSVH is seriously affected by noise. However, the decision boundary of INSVHs does not change clearly, which shows the pinball loss in INSVHs restrains the adverse effect of noise around decision boundary, and the local within-class density weight reduces the interference of label noise far from decision boundary.

Then, the experiments are done to testify the performance of parameter τ in INSVHs model. For better illustration, $d_i^n = 1$ and Gaussian noise with $\theta = 10\%$ is introduced into training dataset according to Ref. [10], where θ is the ratio of the number of noise to that of training samples. Moreover, all samples are normalized before training and testing. Firstly, the INSVHs model is trained with corrupted defects dataset. Then, the testing dataset is used to test the classification accuracy of INSVHs with different τ. The final classification results are shown in Table 2. Specially, $\tau = 0$ means the pinball loss in INSVHs becomes hinge loss. For different types of defects, the classification accuracy of INSVHs reaches the best when $\tau \neq 0$. For example, the classification accuracy of hole defects reaches the best when $\tau = 0.5$, which means the parameter τ adjusts some samples into the hyper-sphere to reduce the adverse effect of noise around decision boundary. So, for the dataset with noise, the pinball loss improves the classification accuracy of INSVHs. On the other

Table 2
Classification accuracy of INSVHs with different τ for corrupted defects dataset. The best accuracy for each defect is bolded.

Defects	Accuracy (%)				
	$\tau = 0$	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.5$	$\tau = 1$
Hole	94.92	96.05	**97.18**	96.61	96.05
Scratch	91.98	91.98	92.59	**93. 21**	92.59
Bruise	92.26	93.45	94.05	94.64	**95.24**
Wave	92.73	93.33	93.94	**95.15**	94.00
Scarring	93.10	94.25	94.83	**95.98**	95.40
Scale	93.33	94.67	**95. 33**	94.00	94.53

Table 3
Classification accuracy of INSVHs with different τ for original defects dataset. The best accuracy for each defect is bolded.

Defects	Accuracy (%)				
	$\tau = 0$	$\tau = 0.1$	$\tau = 0.2$	$\tau = 0.5$	$\tau = 1$
Hole	96.61	97.86	**98.31**	**98.31**	97.52
Scratch	**96.30**	94.93	95.56	95.56	**96.30**
Bruise	95.83	96.55	97.02	**97.62**	97.26
Wave	95.15	**96.97**	**96.97**	95.15	95.88
Scarring	95.98	97.36	**97.70**	96.55	96.55
Scale	96.00	**96.67**	94.00	95.33	96.00

Table 4
The classification result of INSVHs using d_i^n or not for corrupted defects dataset. The best accuracy for each defect is bolded.

Defects	Accuracy (%)	
	Not using d_i^n	Using d_i^n
Hole	96.05	**97.18**
Scratch	92.22	**94.44**
Bruise	93.45	**95.24**
Wave	92.73	**95.76**
Scarring	93.68	**96.55**
Scale	93.33	**95.33**

hand, for different types of defects sets, the τ corresponding to the best classification accuracy are different, which shows that the optimal τ can be obtained by searching. Table 3 shows the classification accuracy of INSVHs for original defects dataset. It can be seen that the classification accuracy reaches to the best for most types of defects when $\tau \neq 0$, which proves that the dataset obtained from production line has noise, and the pinball loss restrains the adverse effect of noise.

Then, the experiments are done to testify the performance of local within-class sample density weight in INSVHs model. For better illustration, $\tau = 0$ and the label noise with $\theta = 5\%$ is randomly introduced into training dataset. Similarly, all samples are normalized before training and testing. Firstly, the INSVHs model is trained with corrupted defects dataset. Then, the testing dataset is used to test classification accuracy for different types of defects. Finally, the classification results corresponding to using d_i^n or not are shown in Table 4. Clearly, the INSVHs model using d_i^n obtains better classification accuracy. On one hand, the weight d_i^n can enhance the effect of important samples. On the other hand, it can restrain the adverse effect of isolated samples especially far from decision boundary. So, for defects dataset with noise, the local within-class sample density weight improves the classification accuracy of INSVHs. Table 5 shows the classification accuracy for original defects dataset. It can be seen that classification accuracy of INSVHs using d_i^n is clearly better than that not using d_i^n, which shows the weight enhances the effect of important samples and restrains the interference of label noise hidden in the original defects dataset. In summary, the experimental results fully testify the performance of local within-class density weight in INSVHs model.

Finally, in order to fully testify the performance of INSVHs classifier, SVM, Pin-SVM, TSVH and INSVHs are compared. The label noise with $\theta = 5\%$ and Gaussian noise with $\theta = 10\%$ are introduced into training dataset. To begin with, in order to manifest the advantage of the novel proposed model, these four binary classifiers are tested on ten UCI datasets. These public UCI datasets [22] have been widely used by researchers all over the world as a primary source of machine learning datasets. The performance of the four binary classifiers on corrupted UCI datasets is shown in Table 6. It can be seen that the binary INSVHs classifier yields the best testing accuracy on 9 of 10 datasets, which shows our INSVHs model has excellent anti-noise property for different datasets. The accuracy of the four binary classifiers on original UCI datasets is shown in Table 7. Our binary INSVHs classifier also yields the best testing

Table 5
The classification result of INSVHs using d_i^n or not for original defects dataset. The best accuracy for each defect is bolded.

Defects	Accuracy (%)	
	Not using d_i^n	Using d_i^n
Hole	97.74	**98.31**
Scratch	95.68	**96.30**
Bruise	96.19	**96.43**
Wave	95.76	**96.36**
Scarring	95.98	**97.36**
Scale	96.00	**96.67**

Table 6
Classification accuracies of four classifiers on corrupted UCI datasets. The best accuracy on each dataset is bolded.

Datasets	INSVHs	Pin-SVM	TSVH	SVM
Heart	**77.98** ± 2.15	77.79 ± 1.94	76.67 ± 2.73	76.60 ± 2.27
Breast	**96.14** ± 1.26	95.77 ± 1.02	94.59 ± 1.54	95.35 ± 1.07
Ionosphere	**94.07** ± 2.20	93.70 ± 2.16	92.64 ± 2.45	92.45 ± 2.31
Haberman	**73.12** ± 2.17	73.07 ± 2.50	71.79 ± 3.01	71.69 ± 2.49
Spect	**82.35** ± 3.01	81.82 ± 3.72	81.96 ± 3.56	82.03 ± 4.95
German	**74.43** ± 2.36	73.28 ± 2.08	73.20 ± 2.67	73.16 ± 2.41
Magic	**82.29** ± 0.72	80.82 ± 0.64	80.22 ± 1.04	80.11 ± 0.87
Pima	77.32 ± 1.04	**77.35** ± 1.20	75.82 ± 2.26	75.70 ± 1.88
Spambase	**89.17** ± 1.27	89.10 ± 1.25	88.75 ± 1.64	88.36 ± 1.27
Transfusion	**78.27** ± 1.32	77.37 ± 1.14	77.01 ± 1.79	76.91 ± 1.90

accuracy on 7 of 10 datasets, which shows the novel model has advantage on classification accuracy for most datasets. Then, in order to solve the classification problem for multi-class defects dataset, SVM, Pin-SVM and TSVH are combined with binary tree. Our INSVHs itself can realize the multi-class classification. For the defects dataset with noise, the classification results of four classifiers are shown in Table 8. It can be seen that the classification accuracy of INSVHs is higher than that of Pin-SVM even though both of them adopt pinball loss. The reason is that the novel local within-class sample density weight in INSVHs restrains label noise and strengthens important samples. The classification accuracy of INSVHs and Pin-SVM is higher than that of SVM and TSVH, which also shows the pinball loss restrains the adverse effect of noise around decision boundary. Moreover, the training time of SVM and Pin-SVM is the longest because their QPPs are big. The QPPs of INSVHs and TSVH are small, so

Table 7
Classification accuracies of four classifiers on original UCI datasets. The best accuracy on each dataset is bolded.

Datasets	INSVHs	Pin-SVM	TSVH	SVM
Heart	**80.46** ± 1.38	80.22 ± 1.73	80.25 ± 1.95	80.20 ± 1.92
Breast	96.19 ± 0.51	**96.21** ± 0.74	96.19 ± 0.98	96.20 ± 0.84
Ionosphere	**94.32** ± 0.81	93.92 ± 1.20	93.91 ± 2.12	93.52 ± 1.96
Haberman	**73.08** ± 1.02	72.88 ± 1.10	72.31 ± 1.74	72.10 ± 1.53
Spect	86.36 ± 1.66	82.59 ± 1.56	85.98 ± 1.54	**86.63** ± 1.76
German	**75.53** ± 1.50	74.39 ± 1.56	75.45 ± 1.82	75.13 ± 1.87
Magic	**83.26** ± 1.17	83.15 ± 0.72	**83.26** ± 1.66	83.23 ± 1.24
Pima	**77.32** ± 1.20	76.83 ± 1.13	76.55 ± 1.34	76.42 ± 1.72
Spambase	89.27 ± 0.60	**89.87** ± 0.78	89.20 ± 1.25	89.20 ± 1.02
Transfusion	**79.06** ± 1.33	78.91 ± 1.44	78.35 ± 1.87	78.22 ± 1.60

Table 8
Classification results of four classifiers for corrupted defects dataset. The best accuracy for four classifiers is bolded.

Classifier	Total accuracy (%)	Training time (s)	Testing time (s)	Searching time (h)
SVM	91.35	598.1	0.7712	4.7873
Pin-SVM	93.14	601.6	0.9215	15.743
TSVH	92.25	289.5	0.8936	11.186
INSVHs	**95.35**	45.72	0.9007	7.9125

Table 9
Classification results of four classifiers for original defects dataset. The best accuracy for four classifiers is bolded.

Classifier	Total accuracy (%)	Training time (s)	Testing time (s)	Searching time (h)
SVM	94.77	584.3	0.7389	4.8053
Pin-SVM	96.27	592.6	0.9463	15.002
TSVH	95.88	292.5	0.8858	10.700
INSVHs	**97.26**	43.20	0.9013	7.2120

their training time is short. The training time of INSVH is shorter than that of TSVH because INSVHs only solves the inequality constrained problem for one class of samples. As far as the testing time concerned that four classifiers are nearly the same and meet the requirements of production line. The parameter searching time is determined by the number of the parameters and the training time. The searching time of SVM is the shortest because it only has two parameters. Though our INSVHs model has 4 parameters, its training time is very short. So, the searching time of INSVHs is smaller than that of Pin-SVM and TSVH. Table 9 shows the classification result of the above four methods for the original defects dataset. It can be seen that the INSVHs classifier has the best classification accuracy, which shows that the steel plate surface defects dataset obtained from real production line exists noise, and the pinball loss and local within-class sample density weight restrain the adverse effect of noise. For training time, the INSVHs classifier has obvious advantage. In summary, the INSVHs classifier has the excellent classification performance in accuracy and efficiency.

5. Conclusions

A novel multi-class classifier termed as INSVHs is proposed for steel plate surface defects in this paper. It uses the pinball loss to adjust some samples from outside of a hyper-sphere into inside of that hyper-sphere, which reduces its sensitivity to noise around decision boundary. Moreover, the INSVHs classifier introduces the local within-class sample density to restrain the adverse effect caused by label noise and isolated samples far from decision boundary, and to strengthen the effect of important samples. On the other hand, in order to improve the execution efficiency, the INSVHs model only optimizes the inequality constraints for one class of samples. Finally, a large number of comparative experiments are done on steel plate surface defects dataset. The experimental results show that the INSVHs classifier has perfect classification performance especially for corrupted defects dataset. How to solve the INSVHs model fast and dispose the unbalance of steel surface defects samples is the next research topic.

Acknowledgement

The authors would like to thank the reviewers for the helpful comments and suggestions. This work was supported by Liaoning Province Ministry of Education Scientific Study Project (No. 2017LNQN11 and No. 2016TSPY13), Liaoning Province PhD Start-up Fund (No. 201601291), Liaoning Province Natural Fund Project (20180550067).

Appendix A. The solution of INSVHs

The QPPs of our INSVHs model can be expressed as:

$$\begin{aligned}
&\min_{\boldsymbol{C}^n,\ R^n} \quad -p_r^n(R^n)^2 + p_e^n\sum_{j=1}^{m^n}\zeta_j + \frac{1}{2}\sum_{i=1}^{m-m^n}\xi_i \\
&\text{s.t.} \quad ||\psi\left(\boldsymbol{B}_j^n\right) - \boldsymbol{C}^n||^2 \geq (R^n)^2 - \zeta_j, \\
&\qquad ||\psi\left(\boldsymbol{B}_j^n\right) - \boldsymbol{C}^n||^2 \leq (R^n)^2 + \frac{1}{\tau}\zeta_j, \qquad n = 1,2,\cdots,N, \\
&\qquad d_i^n\left(||\psi\left(\boldsymbol{A}_i^n\right) - \boldsymbol{C}^n||^2\right) = \xi_i, \\
&\qquad i = 1,2,\cdots,m-m^n,\ \ j = 1,2,\cdots,m^n,
\end{aligned} \tag{A-1}$$

where $p_r^n > 0$ and $p_e^n > 0$. In order to solve the QPPs in (A-1), the operators $\alpha_j^n \geq 0$ and $\beta_j^n \geq 0$ are introduced. Then, the Lagrange function of QPP-n in (A-1) can be described as

$$\begin{aligned}
L = &-p_r^n(R^n)^2 + p_e^n\sum_{j=1}^{m^n}\zeta_j + \frac{1}{2}\sum_{i=1}^{m-m^n} d_i^n||\psi\left(\boldsymbol{A}_i^n\right) - \boldsymbol{C}^n||^2 - \\
&\sum_{j=1}^{m^n}\alpha_j^n\left(\left|\left|\psi\left(\boldsymbol{B}_j^n\right) - \boldsymbol{C}^n\right|\right|^2 - (R^n)^2 + \zeta_j\right) + \\
&\sum_{j=1}^{m^n}\beta_j^n\left(\left|\left|\psi\left(\boldsymbol{B}_j^n\right) - \boldsymbol{C}^n\right|\right|^2 - (R^n)^2 - \frac{1}{\tau}\zeta_j\right).
\end{aligned} \tag{A-2}$$

According to the KKT condition, we can get

$$\frac{\partial L}{\partial \boldsymbol{C}^n} = 2\sum_{j=1}^{m^n}\alpha_j^n\left(\psi\left(\boldsymbol{B}_j^n\right) - \boldsymbol{C}^n\right) - 2\sum_{j=1}^{m^n}\beta_j^n\left(\psi\left(\boldsymbol{B}_j^n\right) - \boldsymbol{C}^n\right) - \sum_{i=1}^{m-m^n} d_i^n\left(\psi\left(\boldsymbol{A}_i^n\right) - \boldsymbol{C}^n\right) = 0, \tag{A-3}$$

$$\frac{\partial L}{\partial (R^n)^2} = -p_r^n + \sum_{j=1}^{m^n}\alpha_j^n - \sum_{j=1}^{m^n}\beta_j^n = 0, \tag{A-4}$$

$$\frac{\partial L}{\partial \zeta_j} = p_e^n - \alpha_j^n - \frac{1}{\tau}\beta_j^n = 0, \tag{A-5}$$

$$\alpha_j^n\left(\left|\left|\psi\left(\boldsymbol{B}_j^n\right) - \boldsymbol{C}^n\right|\right|^2 - (R^n)^2 + \zeta_j\right) = 0, \tag{A-6}$$

$$\beta_j^n\left(\left|\left|\psi\left(\boldsymbol{B}_j^n\right) - \boldsymbol{C}^n\right|\right|^2 - (R^n)^2 - \frac{1}{\tau}\zeta_j\right) = 0. \tag{A-7}$$

According to (A-3) and (A-4), the center $C^n$of Ω^n can be obtained:

$$\boldsymbol{C}^n = \frac{\sum_{i=1}^{m-m^n} d_i^n \psi\left(\boldsymbol{A}_i^n\right) - 2\sum_{j=1}^{m^n}\left(\alpha_j^n - \beta_j^n\right)\psi\left(\boldsymbol{B}_j^n\right)}{\sum_{i=1}^{m-m^n} d_i^n - 2p_r^n}, \tag{A-8}$$

which also derives a by-product for p_r^n that $0 < p_r^n < \frac{1}{2}\sum_{i=1}^{m-m^n} d_i^n$. The operator $\lambda_j^n \geq 0$ is defined and meets $\lambda_j^n = \alpha_j^n - \beta_j^n$. Then, according to (A-2), (A-4), (A-5) and (A-8), we can deduce as follows:

$$\begin{aligned}
L &= \frac{1}{2}\sum_{i=1}^{m-m^n} d_i^n\left|\left|\psi\left(\boldsymbol{A}_i^n\right) - \boldsymbol{C}^n\right|\right|^2 - \sum_{j=1}^{m^n}\alpha_j^n\left|\left|\psi\left(\boldsymbol{B}_j^n\right) - \boldsymbol{C}^n\right|\right|^2 + \sum_{j=1}^{m^n}\beta_j^n\left|\left|\psi\left(\boldsymbol{B}_j^n\right) - \boldsymbol{C}^n\right|\right|^2 \quad = \frac{1}{2}\sum_{i=1}^{m-m^n} d_i^n\left|\left|\psi\left(\boldsymbol{A}_i^n\right) - \boldsymbol{C}^n\right|\right|^2 - \sum_{j=1}^{m^n}\lambda_j^n\left|\left|\psi\left(\boldsymbol{B}_j^n\right) - \boldsymbol{C}^n\right|\right|^2 \\
&= \frac{1}{2}\sum_{i=1}^{m-m^n} d_i^n\psi\left(\boldsymbol{A}_i^n\right)\cdot\psi\left(\boldsymbol{A}_i^n\right) + \frac{1}{2}\sum_{i=1}^{m-m^n} d_i^n\boldsymbol{C}^n\cdot\boldsymbol{C}^n - \sum_{i=1}^{m-m^n} d_i^n\psi\left(\boldsymbol{A}_i^n\right)\cdot\boldsymbol{C}^n - \quad \sum_{j=1}^{m^n}\lambda_j^n\psi\left(\boldsymbol{B}_j^n\right)\cdot\psi\left(\boldsymbol{B}_j^n\right) - \sum_{j=1}^{m^n}\lambda_j^n\boldsymbol{C}^n\cdot\boldsymbol{C}^n + 2\sum_{j=1}^{m^n}\lambda_j^n\psi\left(\boldsymbol{B}_j^n\right)\cdot\boldsymbol{C}^n \\
&= \frac{1}{2}\sum_{i=1}^{m-m^n} d_i^n\psi\left(\boldsymbol{A}_i^n\right)\cdot\psi\left(\boldsymbol{A}_i^n\right) - \sum_{j=1}^{m^n}\lambda_j^n\psi\left(\boldsymbol{B}_j^n\right)\cdot\psi\left(\boldsymbol{B}_j^n\right) + \frac{1}{2}\left(\sum_{i=1}^{m-m^n} d_i^n - \sum_{j=1}^{m^n}\lambda_j^n\right)\boldsymbol{C}^n\cdot\boldsymbol{C}^n - \quad \left(\sum_{i=1}^{m-m^n} d_i^n\psi\left(\boldsymbol{A}_i^n\right) - 2\sum_{j=1}^{m^n}\lambda_j^n\psi\left(\boldsymbol{B}_j^n\right)\right)\cdot\boldsymbol{C}^n \\
&= \frac{1}{2}\sum_{i=1}^{m-m^n} d_i^n\psi\left(\boldsymbol{A}_i^n\right)\cdot\psi\left(\boldsymbol{A}_i^n\right) - \sum_{j=1}^{m^n}\lambda_j^n\psi\left(\boldsymbol{B}_j^n\right)\cdot\psi\left(\boldsymbol{B}_j^n\right) - \frac{1}{2\sum_{i=1}^{m-m^n} d_i^n - 4p_r^n}\left(\sum_{i=1}^{m-m^n} d_i^n\psi\left(\boldsymbol{A}_i^n\right) - 2\sum_{j=1}^{m^n}\lambda_j^n\psi\left(\boldsymbol{B}_j^n\right)\right)^2 \\
&= \frac{1}{2}\sum_{i=1}^{m-m^n} d_i^n\psi\left(\boldsymbol{A}_i^n\right)\cdot\psi\left(\boldsymbol{A}_i^n\right) - \sum_{j=1}^{m^n}\lambda_j^n\psi\left(\boldsymbol{B}_j^n\right)\cdot\psi\left(\boldsymbol{B}_j^n\right) - \frac{1}{2\sum_{i=1}^{m-m^n} d_i^n - 4p_r^n}\left(\sum_{i=1}^{m-m^n}\sum_{i2=1}^{m-m^n} d_i^n d_{i2}^n\psi\left(\boldsymbol{A}_i^n\right)\cdot\psi\left(\boldsymbol{A}_{i2}^n\right) + \quad 4\sum_{j=1}^{m^n}\sum_{j2=1}^{m^n}\lambda_j^n\lambda_{j2}^n\psi\left(\boldsymbol{B}_j^n\right)\cdot\quad \psi\left(\boldsymbol{B}_{j2}^n\right) - 4\sum_{i=1}^{m-m^n}\right. \\
&\quad \left.\times\sum_{j=1}^{m^n} d_i^n\lambda_j^n\psi\left(\boldsymbol{A}_i^n\right)\cdot\psi\left(\boldsymbol{B}_j^n\right)\right) \\
&= \frac{1}{2}\sum_{i=1}^{m-m^n} d_i^n\psi\left(\boldsymbol{A}_i^n\right)\cdot\psi\left(\boldsymbol{A}_i^n\right) + \frac{1}{2\sum_{i=1}^{m-m^n} d_i^n - 4p_r^n}\sum_{i=1}^{m-m^n}\sum_{i2=1}^{m-m^n} d_i^n d_{i2}^n\psi\left(\boldsymbol{A}_i^n\right)\cdot\psi\left(\boldsymbol{A}_{i2}^n\right) - \quad \frac{2}{\sum_{i=1}^{m-m^n} d_i^n - 2p_r^n}\sum_{j=1}^{m^n}\sum_{j2=1}^{m^n}\lambda_j^n\lambda_{j2}^n\psi\left(\boldsymbol{B}_j^n\right)\cdot\quad \psi\left(\boldsymbol{B}_{j2}^n\right) \\
&\quad + \quad \sum_{j=1}^{m^n}\lambda_j^n\left(\frac{2}{\sum_{i=1}^{m-m^n} d_i^n - 2p_r^n}\sum_{i=1}^{m-m^n} d_i^n\psi\left(\boldsymbol{A}_i^n\right)\cdot\psi\left(\boldsymbol{B}_j^n\right) - \psi\left(\boldsymbol{B}_j^n\right)\cdot\psi\left(\boldsymbol{B}_j^n\right)\right).
\end{aligned} \tag{A-9}$$

Discarding the constant items in (A-9) and defining the kernel function $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = \psi(\boldsymbol{x}_1)\cdot\psi(\boldsymbol{x}_2)$, the dual QPP-$n$ of (A-1) can be simplified as following:

$$\begin{aligned}
\max_{\lambda_j^n}\ & -\frac{2}{\sum_{i=1}^{m-m^n} d_i^n - 2p_r^n}\sum_{j1=1}^{m^n}\sum_{j2=1}^{m^n}\lambda_{j1}^n\lambda_{j2}^n K\left(\boldsymbol{B}_{j1}^n, \boldsymbol{B}_{j2}^n\right) + \\
& \sum_{j=1}^{m^n}\lambda_j^n\left(\frac{2\sum_{i=1}^{m-m^n} d_i^n K\left(\boldsymbol{B}_j^n, \boldsymbol{A}_i^n\right)}{\sum_{i=1}^{m-m^n} d_i^n - 2p_r^n} - K\left(\boldsymbol{B}_j^n, \boldsymbol{B}_j^n\right)\right) \\
\text{s.t.}\ & \sum_{j=1}^{m^n}\lambda_j^n = p_r^n,\ -\tau p_e^n \leq \lambda_j^n \leq p_e^n, \\
& j = 1, 2, \cdots, m^n.
\end{aligned} \tag{A-10}$$

If $\boldsymbol{\lambda}^n = [\lambda_1^n, \lambda_2^n, \cdots, \lambda_{m^n}^n]^T$ and $\boldsymbol{e}^n$ is a column vector of ones with m^n dimensions, then the matrix form of QPP (A-10) is

$$\begin{aligned}
\min_{\lambda^n}\ & (\boldsymbol{\lambda}^n)^T\boldsymbol{H}^n\boldsymbol{\lambda}^n \\
\text{s.t.}\ & (\boldsymbol{e}^n)^T\boldsymbol{\lambda}^n = p_r^n, \\
& -\tau p_e^n\boldsymbol{e}^n \leq \boldsymbol{\lambda}^n \leq p_e^n\boldsymbol{e}^n,
\end{aligned} \tag{A-11}$$

where $\boldsymbol{H}^n \in \Re^{m^n \times m^n}$. It can be seen that QPP (A-11) can be solved with quadratic programming optimization. If $\boldsymbol{H}^n = (h^n_{j1,j2})$, then $h_{j1,j2}$ can be calculated with the following formula.

$$h_{j1,j2} = K\left(\boldsymbol{B}^n_{j1}, \boldsymbol{B}^n_{j2}\right) + \frac{\sum_{i=1}^{m-m^n} d^n_i - 2p^n_r}{4p^n_r}\left(K\left(\boldsymbol{B}^n_{j1}, \boldsymbol{B}^n_{j1}\right) + K\left(\boldsymbol{B}^n_{j2}, \boldsymbol{B}^n_{j2}\right)\right) - \frac{1}{2p^n_r}\sum_{i=1}^{m-m^n} d^n_i\left(K\left(\boldsymbol{A}^n_i, \boldsymbol{B}^n_{j1}\right) + K\left(\boldsymbol{A}^n_i, \boldsymbol{B}^n_{j2}\right)\right). \tag{A-12}$$

If Lagrange operator vectors $\boldsymbol{\alpha}^n = [\alpha^n_1, \alpha^n_2, \cdots, \alpha^n_{m^n}]^T$ and $\boldsymbol{\beta}^n = [\beta^n_1, \beta^n_2, \cdots, \beta^n_{m^n}]^T$ are defined, then the relationship of operators $\boldsymbol{\lambda}^n$, $\boldsymbol{\alpha}^n$ and $\boldsymbol{\beta}^n$ is

$$\boldsymbol{\alpha}^n - \boldsymbol{\beta}^n = \boldsymbol{\lambda}^n, p^n_e \boldsymbol{e}^n - \left(1 + \frac{1}{\tau}\right)\boldsymbol{\beta}^n = \boldsymbol{\lambda}^n. \tag{A-13}$$

So, when $-\tau p^n_e < \lambda^n_j < p^n_e$, $\alpha^n_j \neq 0$ and $\beta^n_j \neq 0$ can be deduced. The set S^n is defined as:

$$S^n = \left\{ j \middle| \alpha^n_j \neq 0,\ \beta^n_j \neq 0\ ,\ j = 1, 2, \cdots, m^n \right\}. \tag{A-14}$$

According to (A-6) and (A-7), $(R^n)^2 = \left|\left|\psi(\boldsymbol{B}^n_j) - \boldsymbol{C}^n\right|\right|^2$ can be obtained, where $j \in S^n$. For reliability's sake, $(R^n)^2$ can be calculated as follows:

$$(R^n)^2 = \frac{1}{|S^n|}\sum_{j \in S^n}\left|\left|\psi\left(\boldsymbol{B}^n_j\right) - \boldsymbol{C}^n\right|\right|^2, \tag{A-15}$$

where $|S^n|$ is the number of elements in S^n.

References

[1] F. Dupont, C. Odet, M. Cartont, Optimization of the recognition of defects in flat steel products with the cost matrices theory, NDT E Int. 30 (1) (1997) 3–10.
[2] Y. Yan, K. Song, Z. Xing, X. Feng, The strip steel surface defects classification method based on weak classifier adaptive enhancement, in: Third International Conference on Measuring Technology and Mechatronics Automation, 2011, pp. 958–961.
[3] F. Pernkopf, Detection of surface defects on raw steel blocks using Bayesian network classifiers, Pattern Anal. Appl. 7 (3) (2004) 333–342.
[4] K. Peng, X. Zhang, Classification technology for automatic surface defects detection of steel strip based on improved BP algorithm, in: Fifth International Conference on Natural Computation, 2009, pp. 110–114.
[5] E. Amid, S.R. Aghdam, H. Amindavar, Enhanced performance for support vector machines as multi-class classifiers in steel surface defects detection, World Acad. Sci. Eng. Technol. 6 (7) (2012) 1096–1100.
[6] Real-time steel inspection system based on support vector machine and multiple kernel learning, in: Y. Chen, L. Chen, X. Liu, S. Ding, H. Zhang, Y. Wang, T. Li (Eds.), Practical Applications of Intelligent Systems, Springer, Berlin, Heidelberg, 2011, pp. 185–190.
[7] M. Chu, R. Gong, A. Wang, Strip steel surface defects classification method based on enhanced twin Support vector machine, ISIJ Int. 54 (1) (2014) 119–124.
[8] M.X. Chu, A.N. Wang, R.F. Gong, M. Sha, Multi-class classification methods of enhanced LS-TWSVM for strip steel surface defects, J. Iron Steel Res. Int. 21 (2) (2014) 174–180.
[9] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[10] X. Huang, L. Shi, J.A.K. Suykens, Support vector machine classifier with pinball loss, IEEE Trans. Pattern Anal. Mach. Intell. 36 (5) (2014) 984–997.
[11] F.M. Schleif, P. Tino, Indefinite core vector machine, Pattern Recogn. 71 (2017) 187–195.
[12] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. 29 (5) (2007) 905–910.
[13] D.M.J. Tax, R.P.W. Duin, Support vector data description, Mach. Learn. 54 (1) (2004) 45–66.
[14] X. Peng, D. Xu, Twin support vector hypersphere (TSVH) classifier for pattern recognition, Neural Comput. Appl. 24 (5) (2014) 1207–1220.
[15] H. Cevikalp, Best fitting hyperplanes for classification, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1076–1088.
[16] L. Yi, G. Li, M. Jiang, An end-to-end steel strip surface defects recognition system based on convolutional neural networks, Steel Res. Int. 88 (2) (2017) 176–187.
[17] R. Gong, M. Chu, A. Wang, Y. Yang, A fast detection method for region of defects on strip steel surface, ISIJ Int. 55 (1) (2015) 207–212.
[18] C.H. Lin, J.S. Tsai, C.T. Chiu, Switching bilateral filter with a texture/noise detector for universal noise removal, IEEE Trans. Image Process. 19 (9) (2010) 2307–2320.
[19] L. Vincent, P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulations, IEEE Trans. Pattern Anal. Mach. Intell. 13 (6) (1991) 583–598.
[20] H. Hu, Y. Liu, M. Liu, L. Nie, Surface defects classification in large-scale strip steel image collection via hybrid chromosome genetic algorithm, Neurocomputing 181 (2016) 86–95.
[21] R. Gong, C. Wu, M. Chu, X. Liu, The strip steel surface defects recognition based on multiple support vector hyper-sphere with feature and sample weights, Steel Res. Int. 87 (12) (2016) 1678–1685.
[22] D. Dua, E.K. Taniskidou, UCI Machine Learning Repository, 2017. http://archive.ics.uci.edu/ml.