Pattern De-Noising Based on Support Vector Data Description

Jooyoung Park, Daesung Kang, Jongho Kim
Department of Control and Instrumentation Engineering
Korea University
Jochiwon, Chungnam, 339-700, Korea
E-mail: {parkj, mpkds, oyeasw}@korea.ac.kr

James T. Kwok, Ivor W. Tsang
Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
E-mail: {jamesk, ivor}@cs.ust.hk

Abstract— **The SVDD (support vector data description) is one of the most well-known one-class support vector learning methods, in which one tries the strategy of utilizing balls defined on the feature space in order to distinguish a set of normal data from all other possible abnormal objects. The major concern of this paper is to extend the main idea of the SVDD for the problem of pattern de-noising. Combining the projection onto the spherical decision boundary resulting from the SVDD together with a solver for the pre-image problem, we propose a new method for pattern de-noising. In the proposed method, we first solve the SVDD for the training data, then for each noisy test pattern, perform de-noising by projecting its feature vector onto the decision boundary on the feature space, and finally find the location of the de-noised pattern by obtaining the pre-image of the projection. The applicability of the proposed method is illustrated via an example dealing with noisy handwritten digits.**

I. Introduction

Recently, the support vector learning method has grown up as a viable tool in the area of intelligent systems [1], [2]. Among the important application areas for the support vector learning, we have the one-class classification problems [2], [4], [5], [6], [7], [8], [9], [10], [11]. In the problems of one-class classification, we are in general given only the training data for the normal class, and after the training phase is finished, we are required to decide whether each test vector belongs to normal class or abnormal class. The one-class classification problems are often called outlier detection problems or novelty detection problems. Obvious examples of this class include the fault detection for machines and the intrusion detection system for computers [2]. One of the most well-known support vector learning methods for the one-class problems is the SVDD (support vector data description) [4], [5]. In the SVDD, balls are used for expressing the region for the normal class. Among the methods having the same purpose with the SVDD are the so-called one-class SVM of Schölkopf *et al.* [6], [7], [8], the linear programming method of Campbell and Bennet [9], the information-bottleneck-principle based optimization approach of Crammer and Chechik [10], and the single-class minimax probability machine of Lanckriet *et al.* [11]. Since balls on the input domain can express only limited class of regions, the SVDD in general enhances its expressing power by utilizing balls on the feature space instead of the balls on the input domain. In this paper, we extend the main idea of the SVDD toward the use for the problem of pattern de-noising [12], [13]. Combining the projection onto the spherical decision boundary resulting from the SVDD together with a solver for the pre-image problem, we propose a new method for pattern de-noising. The proposed method consists of the following steps: First, we solve the SVDD for the training data. Second, for each noisy test pattern, we perform de-noising by projecting its feature vector onto the spherical decision boundary on the feature space. Finally in the third step, we recover the location of the de-noised pattern by obtaining the pre-image of the projection following the strategy of [13].

The remaining parts of this paper are organized as follows: In Section 2, preliminaries are provided regarding the SVDD. Our main results on the pattern de-noising based on the SVDD are presented in Section 3. In Section 4, the applicability of the proposed method is illustrated via an example dealing with noisy handwritten digits. Finally, in Section 5, concluding remarks are given.

II. Preliminaries

The SVDD method, which approximates the support (*i.e.*, existing region) of objects belonging to normal class, is derived as follows [4], [5]: Consider a ball B with the center $a \in \Re^d$ and the radius R, and the training data set D consisting of objects $x_i \in \Re^d$, $i = 1, \cdots, N$. Since the training data may be prone to noise, some part of the training data could be abnormal objects. The main idea of the SVDD is to find a ball that can achieve the two conflicting goals simultaneously: First, it should be as small as possible, and with equal importance, it should contain as many training data as possible. Obviously, satisfactory balls satisfying these objectives can be obtained by solving the following optimization problem:

$$\begin{array}{ll} \min & L_0(R^2, a, \xi) = R^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s. t.} & \|x_i - a\|^2 \leq R^2 + \xi_i,\ \xi_i \geq 0, \quad i = 1, \cdots, N. \end{array} \tag{1}$$

Here, the slack variable ξ_i represents the penalty associated with the deviation of the i-th training pattern outside the ball. The objective function of (1) consists of the two conflicting terms, *i.e.*, the square of radius, R^2, and the total penalty

$\sum_{i=1}^{N} \xi_i$. The constant C controls relative importance of each term; thus called the trade-off constant. The dual problem of the above can be derived as follows: First by introducing a Lagrange multiplier for each inequality condition, we obtain the following Lagrange function:

$$L = R^2 + C\sum_{i=1}^{N} \xi_i + \sum_{i=1}^{N} \alpha_i[||x_i - a||^2 - R^2 - \xi_i] - \sum_{i=1}^{N} \eta_i \xi_i, \quad (2)$$

where $\alpha_i \geq 0$, $\eta_i \geq 0$, $\forall i$. From the saddle point condition [1], [2], the optimal solution of (1) satisfies the following:

$$\begin{cases} \partial L/\partial(R^2) = 0; \\ \quad \text{thus } \sum_{i=1}^{N} \alpha_i = 1. \\ \partial L/\partial a = 0; \\ \quad \text{thus } a = (\sum_{i=1}^{N} \alpha_i x_i)/(\sum_{i=1}^{N} \alpha_i) = \sum_{i=1}^{N} \alpha_i x_i. \\ \partial L/\partial \xi_i = 0; \\ \quad \text{thus } \alpha_i \in [0, C], \forall i. \end{cases} \quad (3)$$

Substituting the above into L, the Lagrange function can be expressed in terms of the dual variables:

$$L = \sum_{i=1}^{N} \alpha_i \langle x_i, x_i \rangle - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \langle x_i, x_j \rangle. \quad (4)$$

Thus, the dual problem can now be written as follows:

$$\begin{array}{ll} \max_\alpha & \sum_{i=1}^{N} \alpha_i \langle x_i, x_i \rangle - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \langle x_i, x_j \rangle \\ \text{s. t.} & \sum_{i=1}^{N} \alpha_i = 1, \ \alpha_i \in [0, C], \quad \forall i \end{array} \quad (5)$$

Note that the above is equivalent to the following QP (quadratic programming problem, or quadratic program):

$$\begin{array}{ll} \min_\alpha & \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \langle x_i, x_j \rangle - \sum_{i=1}^{N} \alpha_i \langle x_i, x_i \rangle \\ \text{s. t.} & \sum_{i=1}^{N} \alpha_i = 1, \ \alpha_i \in [0, C], \quad \forall i \end{array} \quad (6)$$

Also, note that from the Kuhn-Tucker complementarity condition [1], the following should hold true at the optimal solution:

$$\alpha_i(||x_i - a||^2 - R^2 - \xi_i) = 0, \quad \forall i \quad (7)$$

From the equation (7), we can easily see that ultimately only the data points on the boundary or outside the ball can have the positive α_i values. These data points are called the support vectors. Once the α_i are obtained by solving (6), the center of the optimal ball can be obtained from (3). Also, the optimal value of R^2 can be found by applying the condition (7) to the support vectors on the ball boundary. After the training phase is over, we may decide whether a given test point $x \in \Re^d$ belongs to the normal class utilizing the following criterion:

$$\begin{array}{rl} & f(x) \\ \triangleq & R^2 - ||x - a||^2 \\ = & R^2 - (\langle x, x \rangle - 2\sum_{i=1}^{N} \alpha_i \langle x_i, x \rangle \\ & \quad + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \langle x_i, x_j \rangle) \\ \geq & 0. \end{array} \quad (8)$$

Obviously, balls can express very limited class of subsets. To express more complex decision regions in $\Re^d$, one can use the so-called feature map $\phi : \Re^d \to F$ and balls defined on the feature space F. More precisely, the problem of finding a reasonably small ball B_F in F that contains reasonably large portion of the (transformed) training data $D_F = \{\phi(x_i) | i = 1, \cdots, N\} \subset F$ can be handled by the following QP:

$$\begin{array}{ll} \min & L_F(R_F^2, a_F, \xi) = R_F^2 + C\sum_{i=1}^{N} \xi_i \\ \text{s. t.} & ||\phi(x_i) - a_F||^2 \leq R_F^2 + \xi_i, \ \xi_i \geq 0, \quad i = 1, \cdots, N. \end{array} \quad (9)$$

Proceeding similar to the above with the so-called kernel trick

$$\langle \phi(y), \phi(z) \rangle = K(y, z), \quad (10)$$

we can derive the following QP for the SVDD utilizing balls on the feature space:

$$\begin{array}{ll} \min_\alpha & \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j K(x_i, x_j) - \sum_{i=1}^{N} \alpha_i K(x_i, x_i) \\ \text{s. t.} & \sum_{i=1}^{N} \alpha_i = 1, \ \alpha_i \in [0, C], \quad \forall i \end{array} \quad (11)$$

If the Gaussian function

$$K(x, z) = \exp(-||x - z||^2/\sigma^2) \quad (12)$$

is chosen for the kernel K, we have $K(x, x) = 1$ for each $x \in \Re^d$. Thus, the above formulation can be further simplified as follows:

$$\begin{array}{ll} \min_\alpha & \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j K(x_i, x_j) \\ \text{s. t.} & \sum_{i=1}^{N} \alpha_i = 1, \ \alpha_i \in [0, C], \quad \forall i \end{array} \quad (13)$$

For simplicity, the use of the Gaussian kernel is assumed throughout this paper. Note that now the result corresponding to (3) contains

$$a_F = \sum_{i=1}^{N} \alpha_i \phi(x_i), \quad (14)$$

and for each support vector x_i on the decision boundary, its feature-space distance from the center a_F is the same with the radius of the ball B_F, thus the following holds true:

$$\begin{array}{rl} & R_F^2 - ||\phi(x_i) - a_F||^2 \\ = & R_F^2 - (1 - 2\sum_{i=1}^{N} \alpha_i K(x_i, x) \\ & \quad + \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j K(x_i, x_j)) \\ = & 0. \end{array} \quad (15)$$

Finally, the criterion for the normality can be summarized as follows:

$$\begin{array}{rcl} f_F(x) & \triangleq & R_F^2 - ||\phi(x) - a_F||^2 \\ & = & R_F^2 - 1 + 2\sum_{i=1}^{N} \alpha_i K(x_i, x) \\ & & \quad - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j K(x_i, x_j) \\ & \geq & 0. \end{array} \quad (16)$$

III. Main Results: Pattern De-Noising Based on SVDD

In the SVDD, the objective is to find the support of the normal objects, and anything outside the support is viewed as abnormal. On the feature space, the support is expressed by a reasonably small ball containing a reasonably large portion of the $\phi(x_i)$. The main idea of this paper is to utilize the ball-shaped support on the feature space for the purpose of correcting test inputs distorted by noise. More precisely, with the trade-off constant C of (9) set appropriately[1], we can find a region where the normal objects without noise generally reside. When an object (which was originally normal) is given as a test input x in a distorted form, the network resulting from the SVDD is supposed to judge that the distorted object x does not belong to the normal class. The role of the SVDD has been conventionally up to this point, and the problem of curing the distortion might be thought beyond the scope of the SVDD. However, here we observe that since the decision region of the SVDD is a simple ball B_F on the feature space F, it is quite easy to let the feature vector $\phi(x)$ of the distorted test input x move toward the center a_F of the ball B_F until it reaches the decision boundary so that it can be tailored enough to be normal. Of course, since the movement starts from the distorted feature $\phi(x)$, there are plenty of reasons to believe that the tailored feature $P\phi(x)$ still contain essential information about the original pattern. Thus, we claim that the tailored feature $P\phi(x)$ is the de-noised version of the feature vector $\phi(x)$. The above arguments together with additional step for finding the pre-image of $P\phi(x)$ comprise our proposal for a new de-noising strategy. In the following, we present the proposed method more precisely with mathematical details.

As mentioned before, the proposed method consists of three steps: In the first step, we solve the SVDD (13) for the given training data $D \overset{\triangle}{=} \{x_i \in \Re^d | i = 1, \cdots, N\}$. As a result, we find the optimal α_i along with a_F and R_F^2 obtained via (14) and (15). In the second step, we consider each test pattern x. When the decision function f_F of (16) yields a nonnegative value for x, the test input is accepted normal as it is, and the de-noising process is bypassed. In the otherwise case, the test input x is considered to be abnormal and distorted by noise. To recover the de-noised pattern, we move its feature vector $\phi(x)$ toward the center a_F up to the point where it touches the ball B_F. Thus, the outcome of this movement is the following:

$$P\phi(x) = a_F + \frac{R_F}{\|\phi(x) - a_F\|}(\phi(x) - a_F). \tag{17}$$

Obviously, this movement is a kind of the projection, and can be interpreted as performing de-noising in the feature space. Note that as a result of the projection, we have the obvious result

$$\|P\phi(x) - a_F\| = R_F. \tag{18}$$

Also, note that with

$$\lambda \overset{\triangle}{=} R_F / \|\phi(x) - a_F\|, \tag{19}$$

the equation (17) can be further simplified into

$$P\phi(x) = \lambda\phi(x) + (1-\lambda)a_F, \tag{20}$$

where λ can be computed from

$$\begin{aligned} \lambda^2 &= \tfrac{R_F^2}{\|\phi(x) - a_F\|^2} \\ &= \tfrac{R_F^2}{(1 - 2\sum_i \alpha_i K(x_i, x) + \sum_i \sum_j \alpha_i \alpha_j K(x_i, x_j))}. \end{aligned} \tag{21}$$

In the third and final step, we try to find the pre-image of the de-noised feature $P\phi(x)$. If the inverse map $\phi^{-1} : F \to \Re^d$ is well-defined and available, this final step attempting to get the de-noised pattern via $\hat{x} = \phi^{-1}(P\phi(x))$ will be trivial. However, the exact pre-image typically does not exist [12]. Thus, we need to seek an approximate solution instead. For this, we follow the strategy of [13], which uses a simple relationship between feature-space distance and input-space distance [14] together with the MDS (multi-dimensional scaling) [15]. Using the kernel trick (10) and the simple relation (20), we see that both $\|P\phi(x)\|^2$ and $\langle P\phi(x), \phi(x_i)\rangle$ can be easily computed as follows:

$$\begin{aligned} \|P\phi(x)\|^2 &= \lambda^2 + 2\lambda(1-\lambda)\textstyle\sum_{i=1}^N \alpha_i K(x_i, x) \\ &\quad + (1-\lambda)^2 \textstyle\sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j K(x_i, x_j), \end{aligned} \tag{22}$$

$$\langle P\phi(x), \phi(x_i)\rangle = \lambda K(x_i, x) + (1-\lambda)\sum_{j=1}^N \alpha_j K(x_i, x_j). \tag{23}$$

Thus, the feature-space distance between $P\phi(x)$ and $\phi(x_i)$ can be obtained via plugging the equations (22) and (23) into the following:

$$\begin{aligned} \tilde{d}^2(P\phi(x), \phi(x_i)) &\overset{\triangle}{=} \|P\phi(x) - \phi(x_i)\|^2 \\ &= \|P\phi(x)\|^2 - 2\langle P\phi(x), \phi(x_i)\rangle + 1. \end{aligned} \tag{24}$$

Now, note that for the Gaussion kernel, the following simple relationship holds true between $d(x_i, x_j) \overset{\triangle}{=} \|x_i - x_j\|$ and $\tilde{d}(\phi(x_i), \phi(x_j)) \overset{\triangle}{=} \|\phi(x_i) - \phi(x_j)\|$ [14]:[2]

$$\begin{aligned} \tilde{d}^2(\phi(x_i), \phi(x_j)) &= \|\phi(x_i) - \phi(x_j)\|^2 \\ &= 2 - 2K(x_i, x_j) \\ &= 2 - 2\exp(-\|x_i - x_j\|^2/\sigma^2) \\ &= 2 - 2\exp(-d^2(x_i, x_j)/\sigma^2). \end{aligned} \tag{25}$$

Since the feature-space distance $\tilde{d}^2(P\phi(x), \phi(x_i))$ is now available from the equation (24) for each training pattern x_i, we can easily obtain the corresponding input-space distance between the desired approximate pre-image $\hat{x}$ of $P\phi(x)$ and each x_i. Generally, the distances with neighbors are the most important in determining the location of any point. Hence here, we only consider the squared input-space distances between $P\phi(x)$ and its n nearest neighbors $\{\phi(x_{(1)}), \cdots, \phi(x_{(n)})\} \subset D_F$, and define

$$d^2 \overset{\triangle}{=} [d_1^2, d_2^2, \cdots, d_n^2]^T, \tag{26}$$

where d_i is the input-space distance between the desired pre-image of $P\phi(x)$ and $x_{(i)}$. In MDS [15], one attempts to find a

[1]In our experiments, $C = 1/(N \times 0.5)$ was used for the purpose of de-noising.

[2]Another possible way to get the input-space distance $d(x_i, x_j)$ is via using the geodesic distance between $\phi(x_i)$ and $\phi(x_j)$ on the sphere instead of the feature-space distance in (25).

representation of the objects that preserves the dissimilarities between each pair of them. Thus, we can use the MDS idea to embed $P\phi(x)$ back to the input space. For this, we first take the average of the training data $\{x_{(1)}, \cdots, x_{(n)}\} \subset D$ to get their centroid $\bar{x} = (1/n)\sum_{i=1}^{n} x_{(i)}$, and construct the $d \times n$ matrix

$$X \triangleq [x_{(1)}, x_{(2)}, \cdots, x_{(n)}]. \tag{27}$$

Here, we note that by defining the $n \times n$ centering matrix

$$H \triangleq I_n - (1/n)1_n 1_n^T,$$

where

$$I_n \triangleq \text{diag}[1, \cdots, 1] \in \Re^{n \times n}$$

and

$$1_n \triangleq [1, \cdots, 1]^T \in \Re^{n \times 1},$$

the matrix XH centers the $x_{(i)}$'s at their centroid, *i.e.*,

$$XH = [x_{(1)} - \bar{x}, \cdots, x_{(n)} - \bar{x}]. \tag{28}$$

The next step is to define a coordinate system in the column space of XH. When X (or XH) is of rank q, we can obtain the SVD (singular value decomposition) [16] of the $d \times n$ matrix XH as

$$\begin{aligned} XH &= [U_1 U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} \\ &= U_1 \Sigma_1 V_1^T \\ &= U_1 Z, \end{aligned} \tag{29}$$

where

$$U_1 = [e_1, \cdots, e_q]$$

is the $d \times q$ matrix with orthonormal columns e_i, and

$$Z \triangleq \Sigma_1 V_1^T = [z_1, \cdots, z_n]$$

is a $q \times n$ matrix with columns z_i being the projections of $x_{(i)} - \bar{x}$ onto the e_j's. Note that

$$\|x_{(i)} - \bar{x}\|^2 = \|z_i\|^2, \quad i = 1, \cdots, n, \tag{30}$$

and collect these into an n-dimensional vector, *i.e.*,

$$d_0^2 \triangleq [\|z_1\|^2, \cdots, \|z_n\|^2]^T. \tag{31}$$

As mentioned before, the location of the pre-image $\hat{x}$ is obtained by requiring $d^2(\hat{x}, x_{(i)}), i = 1, \cdots, n$ to be as close to those values in (26) as possible; thus we need to solve the LS (least squares) problem to find $\hat{x}$:

$$d^2(\hat{x}, x_{(i)}) \simeq d_i^2, \quad i = 1, \cdots, n. \tag{32}$$

Following the steps in [13] and [17], $\hat{z} \in \Re^{n \times 1}$ defined via $\hat{x} - \bar{x} = U_1 \hat{z}$ can be shown to satisfy

$$\hat{z} = -\frac{1}{2}\Sigma_1^{-1} V_1^T (d^2 - d_0^2). \tag{33}$$

Therefore, by transforming (33) back to the original coordinated system in the input space, the location of the recovered de-noised pattern turns out to be

$$\hat{x} = U_1 \hat{z} + \bar{x}. \tag{34}$$

IV. Experiments

For illustration of the proposed method, we report the de-noising results performed for the data set obtained from [18], which consists of 20×16 handwritten digits of 0 through 9, 39 examples of each class. For each of the ten digits, we chose the first $N = 29$ examples to form the training set, and the remaining 10 examples as the test set. The proposed de-noising method was applied to each digit separately. Two types of additive noise were added to the test set in order to get noisy test patterns of Fig. 1: The first one is the Gaussian noise $N(0, \sigma^2)$ with variance $\sigma^2 = 0.4$, and the second one is the so-called *salt and pepper* type noise with noise level $p = 0.4$, where $p/2$ is the probability that a pixel flips to black or white. As the first step for the de-noising with Gaussian kernel $K(x, z) = \exp(-\|x - z\|^2/\sigma^2)$, we solved the SVDD (13) with the trade-off constant and the variance of the Gaussian kernel set as $C = 1/(29 \times 0.5)$ and $\sigma^2 = 1/(N(N-1)\sum_{i=1}^{N}\sum_{j=1}^{N}\|x_i - x_j\|^2)$, respectively. Note that here the value of C was set to the effect that the support for the normal class resulting from the SVDD may cover approximately 50% of the training data. In the second step, the feature vector of each noisy test pattern in Fig. 2 was projected onto the boundary of B_F, which was obtained in the first step. Finally, in the third step, we used ten neighbors to recover the de-noised pattern $\hat{x}$ by solving the pre-image problem. Shown in Fig. 2 are the de-noised images, which seem to be reasonably good.

To compare with existing kernel-based de-noising methods, we also performed de-noising utilizing the methods of [12] and [13], respectively, for the same handwritten digit data. As an index for performance comparison, we used the average squared error, which is defined as follows:

$$E \triangleq \frac{1}{M}\sum_{k=1}^{M} \|t_k - \hat{t}_k\|^2,$$

where M is the number of the test patterns, t_k is the kth member of the test patterns, and $\hat{t}_k$ is the de-noising result for the k^{th} noisy test pattern. Shown in Tables 1 and 2 are the average squared errors for each de-noising method. The methods of [12] and [13] are both based on the kernel PCA (principal component analysis), and require the number of eigenvectors ($\#EV$) as a predetermined parameter. In the tables, we considered the cases with $\#EV$ being 5, 10, 15, and 20. The contents of the tables show that the proposed method yields better results than the other methods.

V. Concluding Remarks

In this paper, we addressed the problem of pattern de-noising based on the SVDD. Along with a brief review over the SVDD, we presented a new de-noising method, which utilizes the SVDD, the projection onto the balls in the feature space, and a method for finding the pre-image of the projection. Works yet to be done include more extensive comparative studies, which will reveal the strength and

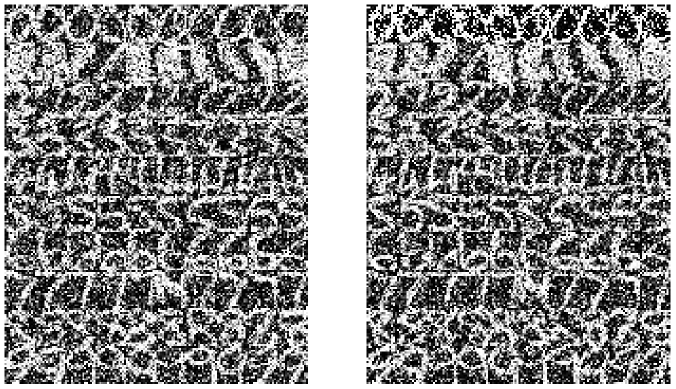

Fig. 1. Test images corrupted by Gaussian noise $\sigma^2 = 0.4$ (left) and *salt and pepper* noise $p = 0.4$ (right).

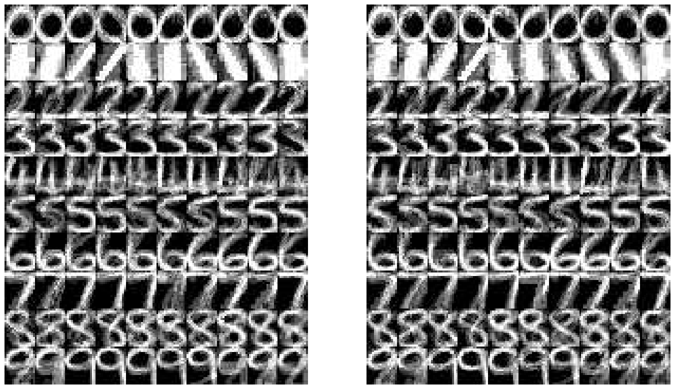

Fig. 2. De-noised results for the cases with Gaussian noise (left) and *salt and pepper* noise (right).

weakness of the proposed method, further refinement of the method for better performance, and the theoretical study about the convenience of the projection (20) in de-noising.

TABLE I

COMPARISON OF AVERAGE SQUARED ERRORS AFTER DE-NOISING THE TEST IMAGES CORRUPTED BY THE GAUSSIAN NOISE

	#EV=5	#EV=10	#EV=15	#EV=20
Method of Mika *et al.* [12]	47.7	45.4	44.1	43.5
Method of Kwok *et al.* [13]	47.7	45.1	44.1	43.5
Proposed method	42.2			

TABLE II

COMPARISON OF AVERAGE SQUARED ERRORS AFTER DE-NOISING THE TEST IMAGES CORRUPTED BY THE *salt and pepper* NOISE

	#EV=5	#EV=10	#EV=15	#EV=20
Method of Mika *et al.* [12]	48.0	45.9	44.9	44.3
Method of Kwok *et al.* [13]	47.9	45.2	44.5	44.0
Proposed method	43.0			

REFERENCES

[1] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge, U.K.: Cambridge University Press, 2000.

[2] B. Schölkopf and A. J. Smola, *Learning with Kernels*, Cambridge, MA: MIT Press, 2002.

[3] C. Bishop, "Novelty detection and neural networks validation," *IEE Proceedings on Vision, Image, and Signal Processing, Special Issue on Applications of Neural Networks*, 1994, vol. 141, pp. 217–222.

[4] D. Tax and R. Duin, "Support Vector Domain Description," *Pattern Recognition Letters*, vol. 20, pp. 1191–1199, 1999.

[5] D. Tax, *One-Class Classification, Ph.D. Thesis*, Delft University of Technology, 2001.

[6] B. Schölkopf, J. C. Platt, and A. J. Smola, *Kernel Method for Percentile Feature Extraction, Technical Report MSR-TR-2000-22*, Microsoft Research, 2000.

[7] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, pp. 1443–1471, 2001.

[8] G. Rätsch, S. Mika, B. Schölkopf, and K.-R. Müller, "Constructing boosting algorithms from SVMs: An application to one-class classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1–15, 2002.

[9] C. Campbell and K. P. Bennett, "A linear programming approach to novelty detection," *Advances in Neural Information Processing Systems*, vol. 13, pp. 395–401, Cambridge, MA: MIT Press, 2001.

[10] K. Crammer and G. Chechik, "A needle in a haystack: Local one-class optimization," *Proceedings of the Twentieth-First International Conference on Machine Learning*, Banff, Alberta, Canada, 2004.

[11] G. R. G. Lanckriet, L. El Ghaoui, and M.I. Jordan, "Robust novelty detection with single-class MPM," *Advances in Neural Information Processing Systems*, vol. 15, Cambridge, MA: MIT Press, 2003.

[12] S. Mika, B. Schölkopf, A. Smola, K. R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature space," *Advances in Neural Information Processing Systems*, vol. 11, pp. 536–542, Cambridge, MA: MIT Press, 1999.

[13] J. T. Kwok and I. W. Tsang, "The pre-image problem in kernel methods," *IEEE Transactions on Neural Networks*, vol. 15, pp. 1517–1525, 2004.

[14] C. K. I. Williams, "On a connection between kernel PCA and metric multidimensional scaling," *Machine Learning*, vol. 46, pp. 11–19, 2002.

[15] T. F. Cox and M. A. A. Cox, "Multidimensional Scaling," *Monographs on Statistics and Applied Probability*, vol. 88, *2nd Ed.*, London, U.K.: Chapman & Hall, 2001.

[16] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*, Upper Saddle River, NJ: Prentice Hall, 2000.

[17] J. C. Gower, "Adding a point to vector diagrams in multivariate analysis," *Biometrika*, vol. 55, pp. 582–585, 1968.

[18] `http://www.cs.toronto.edu/~roweis/data.html.`