LETTER Communicated by Klaus-Robert Müller

SVDD-Based Pattern Denoising

Jooyoung Park
parkj@korea.ac.kr
Daesung Kang
mpkds@korea.ac.kr
Department of Control and Instrumentation Engineering, Korea University, Jochiwon, Chungnam, 339-700, Korea

Jongho Kim
jongho6270.kim@samsung.com
Mechatronics and Manufacturing Technology Center, Samsung Electronics Co., Ltd., Suwon, Gyeonggi, 443-742, Korea

James T. Kwok
jamesk@cse.ust.hk
Ivor W. Tsang
ivor@cse.ust.hk
Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong

The support vector data description (SVDD) is one of the best-known one-class support vector learning methods, in which one tries the strategy of using 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 letter is to extend the main idea of SVDD to pattern denoising. Combining the geodesic projection to the spherical decision boundary resulting from the SVDD, together with solving the preimage problem, we propose a new method for pattern denoising. We first solve SVDD for the training data and then for each noisy test pattern, obtain its denoised feature by moving its feature vector along the geodesic on the manifold to the nearest decision boundary of the SVDD ball. Finally we find the location of the denoised pattern by obtaining the pre-image of the denoised feature. The applicability of the proposed method is illustrated by a number of toy and real-world data sets.

1 Introduction

Recently, the support vector learning method has become a viable tool in the area of intelligent systems (Cristianini & Shawe-Taylor, 2000; Schölkopf & Smola, 2002). Among the important application areas for support

© 2007 Massachusetts Institute of Technology

vector learning, we have the one-class classification problems (Campbell & Bennett, 2001; Crammer & Chechik, 2004; Lanckriet, El Ghaoui, & Jordan, 2003; Laskov, Schäfer, & Kotenko, 2004; Müller, Mika, Rätsch, Tsuda, & Schölkopf, 2001; Pekalska, Tax, & Duin, 2003; Rätsch, Mika, Schölkopf, & Müller, 2002; Schölkopf, Platt, & Smola, 2000; Schölkopf, Platt, Shawe-Taylor, Smola, & Williamson, 2001; Schölkopf & Smola, 2002; Tax, 2001; Tax & Duin, 1999). In one-class classification problems, we are 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 the normal or the abnormal class. One-class classification problems are often called outlier detection problems or novelty detection problems. Obvious examples of this class include fault detection for machines and the intrusion detection system for computers (Schölkopf & Smola, 2002).

One of the best-known support vector learning methods for the one-class problems is the SVDD (support vector data description) (Tax, 2001; Tax & Duin, 1999). 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 and others (Rätsch et al., 2002; Schölkopf et al., 2001; Schölkopf et al., 2000), the linear programming method of Campbell and Bennet (2001), the information-bottleneck-principle-based optimization approach of Crammer and Chechik (2004), and the single-class minimax probability machine of Lanckriet et al. (2003). Since balls on the input domain can express only a 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 letter, we extend the main idea of the SVDD toward the use for the problem of pattern denoising (Kwok & Tsang, 2004; Mika et al., 1999; Schölkopf et al., 1999). Combining the movement to the spherical decision boundary resulting from the SVDD together with a solver for the preimage problem, we propose a new method for pattern denoising that consists of the following steps. First, we solve the SVDD for the training data consisting of the prototype patterns. Second, for each noisy test pattern, we obtain its denoised feature by moving its feature vector along the geodesic to the spherical decision boundary of the SVDD ball on the feature space. Finally in the third step, we recover the location of the denoised pattern by obtaining the preimage of the denoised feature following the strategy of Kwok and Tsang (2004).

The remaining parts of this letter are organized as follows. In section 2, preliminaries are provided regarding the SVDD. Our main results on pattern denoising based on the SVDD are presented in section 3. In section 4, the applicability of the proposed method is illustrated by a number of toy and real-world data sets. Finally, in section 5, concluding remarks are given.

2 Preliminaries

The SVDD method, which approximates the support of objects belonging to normal class, is derived as follows (Tax, 2001; Tax & Duin, 1999). Consider a ball B with center $\mathbf{a} \in \mathbb{R}^d$ and radius R, and the training data set D consisting of objects $\mathbf{x}_i \in \mathbb{R}^d, i = 1, \ldots, N$. The main idea of SVDD is to find a ball that can achieve two conflicting goals (it should be as small as possible and contain as many training data as possible) simultaneously by solving

$$\begin{aligned} \min \quad & L_0(R^2, \mathbf{a}, \boldsymbol{\xi}) = R^2 + C \sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & \|\mathbf{x}_i - \mathbf{a}\|^2 \leq R^2 + \xi_i, \ \ \xi_i \geq 0, \quad i = 1, \ldots, N. \end{aligned} \tag{2.1}$$

Here, the slack variable ξ_i represents the penalty associated with the deviation of the ith training pattern outside the ball, and C is a trade-off constant controlling the relative importance of each term. The dual problem of equation 2.1 is

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{i=1}^{N} \alpha_i \langle \mathbf{x}_i, \mathbf{x}_i \rangle - \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \, \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle \\ \text{s.t.} \quad & \sum_{i=1}^{N} \alpha_i = 1, \ \ \alpha_i \in [0, C], \quad i = 1, \ldots, N. \end{aligned} \tag{2.2}$$

In order to express more complex decision regions in $\mathbb{R}^d$, one can use the so-called feature map $\phi : \mathbb{R}^d \to F$ and balls defined on the feature space F. Proceeding similar to the above and utilizing the kernel trick $\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle = k(\mathbf{x}, \mathbf{z})$, one can find the corresponding feature space SVDD ball B_F in F. Moreover, from the Kuhn-Tucker condition, its center can be expressed as

$$\mathbf{a}_F = \sum_{i=1}^{N} \alpha_i \phi(\mathbf{x}_i), \tag{2.3}$$

and its radius R_F can be computed by utilizing the distance between $\mathbf{a}_F$ and any support vector $\mathbf{x}$ on the ball boundary:

$$R_F^2 = k(\mathbf{x}, \mathbf{x}) - 2 \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j). \tag{2.4}$$

In this letter, we always use the gaussian kernel $k(\mathbf{x}, \mathbf{z}) = \exp(-\|\mathbf{x} - \mathbf{z}\|^2/s^2)$, and so $k(\mathbf{x}, \mathbf{x}) = 1$ for each $\mathbf{x} \in \mathbb{R}^d$. Finally, note that in this case, the SVDD formulation is equivalent to

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \quad \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{s.t.} \quad \sum_{i=1}^{N} \alpha_i = 1,\ \alpha_i \in [0, C], \quad i = 1, \ldots, N, \end{aligned} \tag{2.5}$$

and the resulting criterion for the normality is

$$f_F(\mathbf{x}) \overset{\triangle}{=} R_F^2 - \|\phi(\mathbf{x}) - \mathbf{a}_F\|^2 \geq 0. \tag{2.6}$$

3 Main Results

In SVDD, the objective is to find the support of the normal objects; anything outside the support is viewed as abnormal. In the feature space, the support is expressed by a reasonably small ball containing a reasonably large portion of the $\phi(\mathbf{x}_i)$'s. The main idea of this letter is to utilize the ball-shaped support on the feature space for correcting test inputs distorted by noise. More precisely, with the trade-off constant C 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 $\mathbf{x}$ in a distorted form, the network resulting from the SVDD is supposed to judge that the distorted object $\mathbf{x}$ does not belong to the normal class. The role of the SVDD has been conventional up to this point, and the problem of curing the distortion might be thought of as beyond the scope of the SVDD.

In this letter, we go one step further and move the feature vector $\phi(\mathbf{x})$ of the distorted test input $\mathbf{x}$ to the point $Q\phi(\mathbf{x})$ lying on the surface of the SVDD ball B_F so that it can be tailored enough to be normal (see Figure 1). Given that all the points in the input space are mapped to a manifold in the kernel-induced feature space (Burges, 1999), the movement is along the geodesic on this manifold, and so the point $Q\phi(\mathbf{x})$ can be considered as the geodesic projection of $\phi(\mathbf{x})$ onto the SVDD ball. Of course, since the movement starts from the distorted feature $\phi(\mathbf{x})$, there are plenty of reasons to believe that the tailored feature $Q\phi(\mathbf{x})$ still contains essential information about the original pattern. We claim that the tailored feature $Q\phi(\mathbf{x})$ is the denoised version of the feature vector $\phi(\mathbf{x})$. Pertinent to this claim is the discussion of Ben-Hur, Horn, Siegelmann, and Vapnik (2001) on the support vector clustering, in which the SVDD is shown to be a very efficient tool for clustering since the SVDD ball, when mapped back to

[1] In our experiments for noisy handwritten digits, $C = 1/(N \times 0.2)$ was used for the purpose of denoising.

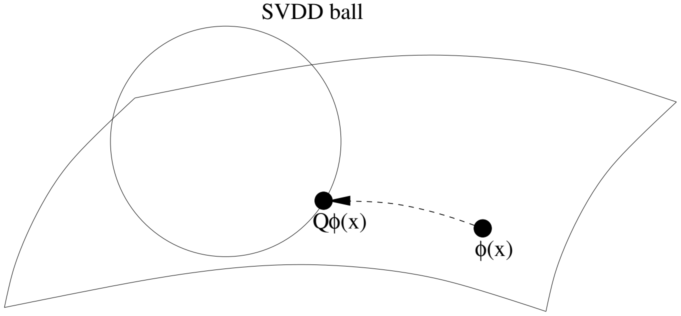

Figure 1: Basic idea for finding the denoised feature vector $Q\phi(\mathbf{x})$ by moving along the geodesic.

the input space, can separate into several components, each enclosing a separate cluster of normal data points, and can generate cluster boundaries of arbitrary shapes. These arguments, together with an additional step for finding the preimage of $Q\phi(\mathbf{x})$, comprise our proposal for a new denoising strategy. In the following, we present the proposed method more precisely with mathematical details.

The proposed method consists of three steps. First, we solve the SVDD, equation 2.5, for the given prototype patterns $D \overset{\triangle}{=} \{\mathbf{x}_i \in \mathbb{R}^d | i = 1, \ldots, N\}$. As a result, we find the optimal α_i's along with $\mathbf{a}_F$ and R_F^2 obtained via equations 2.3 and 2.4. Second, we consider each test pattern $\mathbf{x}$. When the decision function f_F of equation 2.6 yields a nonnegative value for $\mathbf{x}$, the test input is accepted as normal, and the denoising process is bypassed with $Q\phi(\mathbf{x})$ set equal to $\phi(\mathbf{x})$. Otherwise, the test input $\mathbf{x}$ is considered to be abnormal and distorted by noise. To recover the denoised pattern, we move its feature vector $\phi(\mathbf{x})$ along the geodesic defined on the manifold in the feature space, toward the SVDD ball B_F up to the point where it touches the ball. In principle, any kernel can be used here. However, as we will show, closed-form solutions can be obtained when stationary kernels (such as the gaussian kernel) are used.[2] In this case, it is obvious that all the points are mapped onto the surface of a ball in the feature space, and we can see from Figure 2 that the point $Q\phi(\mathbf{x})$ is the ultimate destination of this movement. For readers' convenience, we also include a three-dimensional drawing (see Figure 3) to clarify Figure 2.

In the following, the proposed method will be presented only for the gaussian kernel, where all the points are mapped to the unit ball in the feature space. Extension to other stationary kernels is straightforward. In order to find $Q\phi(\mathbf{x})$, it is necessary to solve the following series of subproblems:

[2] A stationary kernel $k(\mathbf{x}, \mathbf{x}')$ is a kernel that depends on only $\mathbf{x} - \mathbf{x}'$.

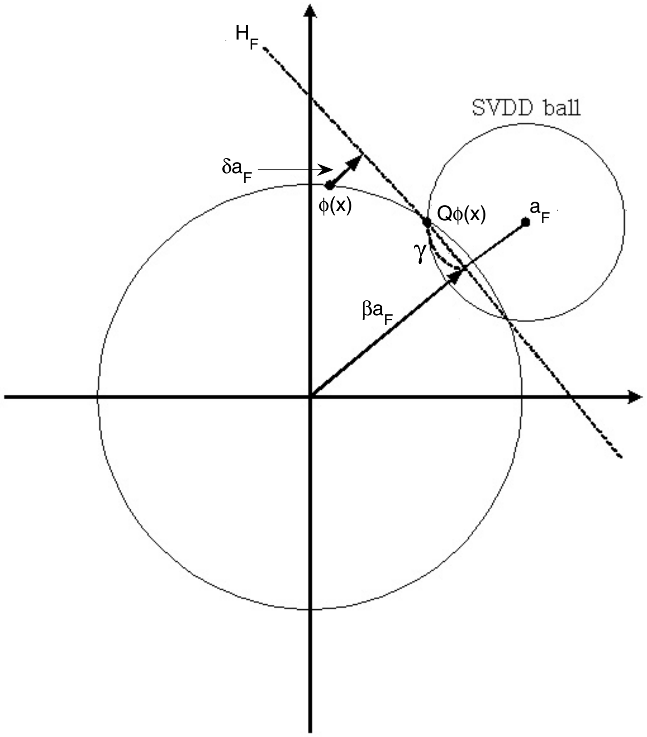

Figure 2: Proposed denoising procedure when a stationary kernel is used.

- **To find the separating hyperplane** H_F. From Figure 2, it is clear that for the SVDD problems utilizing stationary kernels, the center $\mathbf{a}_F$ of the SVDD ball has the same direction with the weight vector of the separating hyperplane H_F. In particular, when the gaussian kernel is used, the hyperplane H_F can be represented by

$$2\langle \mathbf{a}_F, \phi(\mathbf{x})\rangle = 1 + \|\mathbf{a}_F\|^2 - R_F^2. \tag{3.1}$$

  Further information needed for identifying the location of $Q\phi(\mathbf{x})$ includes the vectors $\beta\mathbf{a}_F$, $\delta\mathbf{a}_F$, and the distance γ shown in Figure 2.

- **To find vector** $\beta\mathbf{a}_F$. As shown in Figure 2, the vector $\beta\mathbf{a}_F$ lies on the hyperplane H_F. Thus, it should satisfy equation 3.1—that is,

$$2\langle \mathbf{a}_F, \beta\mathbf{a}_F\rangle = 1 + \|\mathbf{a}_F\|^2 - R_F^2. \tag{3.2}$$

  Therefore, we have

$$\beta = \frac{1 + \|\mathbf{a}_F\|^2 - R_F^2}{2\|\mathbf{a}_F\|^2} = \frac{1 + \boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} - R_F^2}{2\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}}, \tag{3.3}$$

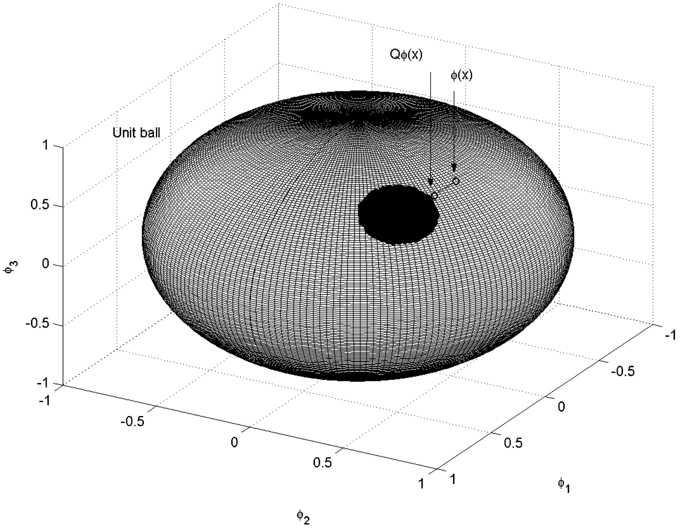

Figure 3: Denoised feature vector $Q\phi(\mathbf{x})$ shown in a (hypothetical) three-dimensional feature space. Here, since the chosen kernel is stationary, the projected feature vector $Q\phi(\mathbf{x})$, as well as the feature vector $\phi(\mathbf{x})$, should lie on a ball centered at the origin of the feature space. Also note that the location of $Q\phi(\mathbf{x})$ should be at the boundary of the intersection of the ball surface and the SVDD ball, which is colored black.

where $\boldsymbol{\alpha} \triangleq [\alpha_1 \cdots \alpha_N]^T$, and $\mathbf{K}$ is the kernel matrix with entries $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$.

- **To find distance** γ. Since $Q\phi(\mathbf{x})$ is on the surface of the unit ball, we have $\|Q\phi(\mathbf{x})\|^2 = 1$. Also from the Pythagorean theorem, $\|\beta\mathbf{a}_F\|^2 + \gamma^2 = \|Q\phi(\mathbf{x})\|^2$ holds. Hence, we have

$$\gamma = \sqrt{1 - \beta^2\|\mathbf{a}_F\|^2} = \sqrt{1 - \beta^2\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}}. \tag{3.4}$$

- **To find vector** $\delta\mathbf{a}_F$. Since $P\phi(\mathbf{x}) \triangleq \phi(\mathbf{x}) + \delta\mathbf{a}_F$ should lie on the hyperplane H_F, it should satisfy equation 3.1. Thus, the following holds:

$$2\langle\mathbf{a}_F, \phi(\mathbf{x}) + \delta\mathbf{a}_F\rangle = 1 + \|\mathbf{a}_F\|^2 - R_F^2. \tag{3.5}$$

Hence, we have

$$\delta = \frac{1 + \|\mathbf{a}_F\|^2 - R_F^2 - 2\langle\mathbf{a}_F, \phi(\mathbf{x})\rangle}{2\|\mathbf{a}_F\|^2} = \frac{1 + \boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} - R_F^2 - 2\mathbf{k}_\mathbf{x}\boldsymbol{\alpha}}{2\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha}}, \tag{3.6}$$

where $\mathbf{k}_\mathbf{x} \triangleq [k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_N)]^T$.

- **To find the denoised feature vector** $Q\phi(\mathbf{x})$. From Figure 2, we see that

$$Q\phi(\mathbf{x}) = \beta \mathbf{a}_F + \frac{\gamma}{\|\phi(\mathbf{x}) + (\delta - \beta)\mathbf{a}_F\|}(\phi(\mathbf{x}) + (\delta - \beta)\mathbf{a}_F). \tag{3.7}$$

Note that with

$$\lambda_1 \stackrel{\triangle}{=} \frac{\gamma}{\|\phi(\mathbf{x}) + (\delta - \beta)\mathbf{a}_F\|} \tag{3.8}$$

and

$$\lambda_2 \stackrel{\triangle}{=} \beta + \frac{\gamma(\delta - \beta)}{\|\phi(\mathbf{x}) + (\delta - \beta)\mathbf{a}_F\|}, \tag{3.9}$$

the above expression for $Q\phi(\mathbf{x})$ can be further simplified into

$$Q\phi(\mathbf{x}) = \lambda_1 \phi(\mathbf{x}) + \lambda_2 \mathbf{a}_F, \tag{3.10}$$

where λ_1 and λ_2 can be computed from

$$\lambda_1 = \frac{\gamma}{\sqrt{1 + 2(\delta - \beta)\mathbf{k}_{\mathbf{x}}\boldsymbol{\alpha} + (\delta - \beta)^2 \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}}}, \tag{3.11}$$

$$\lambda_2 = \beta + \frac{\gamma(\delta - \beta)}{\sqrt{1 + 2(\delta - \beta)\mathbf{k}_{\mathbf{x}}\boldsymbol{\alpha} + (\delta - \beta)^2 \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha}}}. \tag{3.12}$$

Obviously, the movement from the feature $\phi(\mathbf{x})$ to $Q\phi(\mathbf{x})$ is along the geodesic to the noise-free normal class and thus can be interpreted as performing denoising in the feature space. With this interpretation in mind, the feature vector $Q\phi(\mathbf{x})$ will be called the denoised feature of **x** in this letter. In the third and final steps, we try to find the preimage of the denoised feature $Q\phi(\mathbf{x})$. If the inverse map $\phi^{-1} : F \to \mathbb{R}^d$ is well defined and available, this final step attempting to get the denoised pattern via $\hat{\mathbf{x}} = \phi^{-1}(Q\phi(\mathbf{x}))$ will be trivial. However, the exact preimage typically does not exist (Mika et al., 1999). Thus, we need to seek an approximate solution instead. For this, we follow the strategy of Kwok and Tsang (2004), which uses a simple relationship between feature-space distance and input-space distance (Williams, 2002) together with the MDS (multi-dimensional scaling) (Cox & Cox, 2001). Using the kernel trick and the simple relation, equation 3.10, we see that $\langle Q\phi(\mathbf{x}), \phi(\mathbf{x}_i)\rangle$ can be easily computed as follows:

$$\langle Q\phi(\mathbf{x}), \phi(\mathbf{x}_i)\rangle = \lambda_1 k(\mathbf{x}_i, \mathbf{x}) + \lambda_2 \sum_{j=1}^{N} \alpha_j k(\mathbf{x}_i, \mathbf{x}_j). \tag{3.13}$$

Thus, the feature space distance between $Q\phi(\mathbf{x})$ and $\phi(\mathbf{x}_i)$ can be obtained by plugging equation 3.13 into

$$\begin{aligned}\tilde{d}^2(Q\phi(\mathbf{x}), \phi(\mathbf{x}_i)) &\stackrel{\triangle}{=} \|Q\phi(\mathbf{x}) - \phi(\mathbf{x}_i)\|^2 \\ &= 2 - 2\langle Q\phi(\mathbf{x}), \phi(\mathbf{x}_i)\rangle. \end{aligned} \tag{3.14}$$

Now, note that for the gaussian kernel, the following simple relationship holds true between $d(\mathbf{x}_i, \mathbf{x}_j) \triangleq \|\mathbf{x}_i - \mathbf{x}_j\|$ and $\tilde{d}(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)) \triangleq \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|$ (Williams, 2002):

$$\begin{aligned}\tilde{d}^2(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)) &= \|\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\|^2 \\ &= 2 - 2k(\mathbf{x}_i, \mathbf{x}_j) \\ &= 2 - 2\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/s^2) \\ &= 2 - 2\exp(-d^2(\mathbf{x}_i, \mathbf{x}_j)/s^2). \end{aligned} \tag{3.15}$$

Since the feature space distance $\tilde{d}^2(Q\phi(\mathbf{x}), \phi(\mathbf{x}_i))$ is now available from equation 3.14 for each training pattern $\mathbf{x}_i$, we can easily obtain the corresponding input space distance between the desired approximate preimage $\hat{\mathbf{x}}$ of $Q\phi(\mathbf{x})$ and each $\mathbf{x}_i$. Generally, the distances with neighbors are the most important in determining the location of any point. Hence, here we consider only the squared input space distances between $Q\phi(\mathbf{x})$ and its n nearest neighbors $\{\phi(\mathbf{x}_{(1)}), \ldots, \phi(\mathbf{x}_{(n)})\} \subset D_F$, and define

$$\mathbf{d^2} \triangleq [d_1^2, d_2^2, \ldots, d_n^2]^T, \tag{3.16}$$

where d_i is the input space distance between the desired preimage of $Q\phi(\mathbf{x})$ and $\mathbf{x}_{(i)}$. In MDS (Cox & Cox, 2001), one attempts to find a representation of the objects that preserves the dissimilarities between each pair of them. Thus, we can use the MDS idea to embed $Q\phi(\mathbf{x})$ back to the input space. For this, we first take the average of the training data $\{\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(n)}\} \subset D$ to get their centroid $\bar{\mathbf{x}} = (1/n)\sum_{i=1}^{n} \mathbf{x}_{(i)}$, and construct the $d \times n$ matrix,

$$\mathbf{X} \triangleq [\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \ldots, \mathbf{x}_{(n)}]. \tag{3.17}$$

Here, we note that by defining the $n \times n$ centering matrix $\mathbf{H} \triangleq \mathbf{I}_n - (1/n)\mathbf{1}_n\mathbf{1}_n^T$, where $\mathbf{I}_n \triangleq \mathrm{diag}[1, \ldots, 1] \in \mathbb{R}^{n\times n}$ and $\mathbf{1}_n \triangleq [1, \ldots, 1]^T \in \mathbb{R}^{n\times 1}$, the matrix $\mathbf{XH}$ centers the $\mathbf{x}_{(i)}$'s at their centroid:

$$\mathbf{XH} = [\mathbf{x}_{(1)} - \bar{\mathbf{x}}, \ldots, \mathbf{x}_{(n)} - \bar{\mathbf{x}}]. \tag{3.18}$$

The next step is to define a coordinate system in the column space of $\mathbf{XH}$. When $\mathbf{XH}$ is of rank q, we can obtain the SVD (singular value decomposition) (Moon & Stirling, 2000) of the $d \times n$ matrix $\mathbf{XH}$ as

$$\begin{aligned}\mathbf{XH} &= [\mathbf{U}_1\mathbf{U}_2]\begin{bmatrix}\boldsymbol{\Sigma}_1 & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{V}_1^T\\ \mathbf{V}_2^T\end{bmatrix}\\ &= \mathbf{U}_1\boldsymbol{\Sigma}_1\mathbf{V}_1^T\\ &= \mathbf{U}_1\mathbf{Z},\end{aligned} \tag{3.19}$$

where $\mathbf{U}_1 = [\mathbf{e}_1, \ldots, \mathbf{e}_q]$ is the $d \times q$ matrix with orthonormal columns $\mathbf{e}_i$, and $\mathbf{Z} \triangleq \boldsymbol{\Sigma}_1\mathbf{V}_1^T = [\mathbf{z}_1, \ldots, \mathbf{z}_n]$ is a $q \times n$ matrix with columns $\mathbf{z}_i$ being the projections of $\mathbf{x}_{(i)} - \bar{\mathbf{x}}$ onto the $\mathbf{e}_j$'s. Note that

$$\|\mathbf{x}_{(i)} - \bar{\mathbf{x}}\|^2 = \|\mathbf{z}_i\|^2, \quad i = 1, \ldots, n, \tag{3.20}$$

and collect these into an n-dimensional vector:

$$\mathbf{d}_\mathbf{0}^\mathbf{2} \triangleq [\|\mathbf{z}_1\|^2, \ldots, \|\mathbf{z}_n\|^2]^T. \tag{3.21}$$

The location of the preimage $\hat{\mathbf{x}}$ is obtained by requiring $d^2(\hat{\mathbf{x}}, \mathbf{x}_{(i)}), i = 1, \ldots, n$ to be as close to those values in equation 3.16 as possible; thus, we need to solve the LS (least squares) problem to find $\hat{\mathbf{x}}$:

$$d^2(\hat{\mathbf{x}}, \mathbf{x}_{(i)}) \simeq d_i^2, \quad i = 1, \ldots, n. \tag{3.22}$$

Now following the steps of Kwok and Tsang (2004) and Gower (1968), $\hat{\mathbf{z}} \in \mathbb{R}^{n\times 1}$ defined by $\hat{\mathbf{x}} - \bar{\mathbf{x}} = \mathbf{U}_1\hat{\mathbf{z}}$ can be shown to satisfy

$$\hat{\mathbf{z}} = -\frac{1}{2}\boldsymbol{\Sigma}_1^{-1}\mathbf{V}_1^T\left(\mathbf{d}^2 - \mathbf{d}_0^2\right). \tag{3.23}$$

Therefore, by transforming equation 3.23 back to the original coordinated system in the input space, the location of the recovered denoised pattern turns out to be

$$\hat{\mathbf{x}} = \mathbf{U}_1\hat{\mathbf{z}} + \bar{\mathbf{x}}. \tag{3.24}$$

4 Experiments

In this section, we compare the performance of the proposed method with other denoising methods on toy and real-world data sets. For simplicity, we denote the proposed method by SVDD.

4.1 Toy Data Set. We first use a toy example to illustrate the proposed method and compare its reconstruction performance with PCA. The setup is similar to that in Mika et al. (1999). Eleven clusters of samples are generated by first choosing 11 independent sources randomly in $[-1, 1]^{10}$ and then drawing samples uniformly from translations of $[-\sigma_0, \sigma_0]^{10}$ centered at each source. For each source, 30 points are generated to form the training data and 5 points to form the clean test data. Normally distributed noise, with variance σ_o^2 in each component, is then added to each clean test data point to form the corrupted test data.

We carried out SVDD (with $C = \frac{1}{N\times 0.6}$) and PCA for the training set, and then performed reconstructions of each corrupted test point using both the proposed SVDD-based method (with neighborhood size $n = 10$) and the standard PCA method.[3] The procedure was repeated for different numbers of principal components in PCA and for different values of σ_0. For the width s of the gaussian kernel, we used $s^2 = 2 \times 10 \times \sigma_0^2$ as in Mika et al. (1999). From the simulations, we found out that when the input space dimensionality d is low (as in this example, where $d = 10$), applying the proposed method iteratively (i.e., recursively applying the denoising to the previous denoised results) can improve the performance.

We compared the results of our method (with 100 iterations) to those of the PCA-based method using the mean squared distance (MSE), which is defined as

$$\mathrm{MSE} \triangleq \frac{1}{M}\sum_{k=1}^{M} \|\mathbf{t}_k - \hat{\mathbf{t}}_k\|^2, \tag{4.1}$$

where M is the number of test patterns, $\mathbf{t}_k$ is the kth clean test pattern, and $\hat{\mathbf{t}}_k$ is the denoised result for the kth noisy test pattern. Table 1 shows the ratio of $\mathrm{MSE}_{PCA}/\mathrm{MSE}_{SVDD}$. Note that ratios larger than one indicate that the proposed SVDD-based method performs better compared to the other one.

Simulations were also performed for a two-dimensional version of the toy example (see Figure 4a), and the denoised results were shown in Figures 4b and 4c. For PCA, we used only one eigenvector (if two eigenvectors were used, the result is just a change of basis and thus not useful). The observed MSE values for the reconstructions using the proposed and PCA-based methods were 0.0192 and 0.1902, respectively.

From Table 1 and Figures 4b and 4c, one can see that in the considered examples, the proposed method yielded better performance than the PCA-based method. The reason seems to be that here, the examples basically deal

[3] The corresponding Matlab program is posted online at http://cie.korea.ac.kr/ac_lab/pro01.html.

Table 1: Comparison of MSE Ratios After Reconstructing the Corrupted Test Points in $\mathbb{R}^{10}$.

	#EV=1	#EV=3	#EV=5	#EV=7	#EV=9
$\sigma_0 = 0.05$	193.6	91.90	37.37	12.44	2.528
$\sigma_0 = 0.10$	49.36	23.85	10.51	4.667	2.421
$\sigma_0 = 0.15$	22.71	11.34	5.613	3.241	2.419
$\sigma_0 = 0.20$	13.13	6.853	3.854	2.705	2.392

Note: Performance ratios, $\text{MSE}_{PCA}/\text{MSE}_{SVDD}$, being larger than one, indicate how much better SVDD did compared to PCA for different choices of σ_0, and different numbers of principal components (#EV) in reconstruction using PCA.

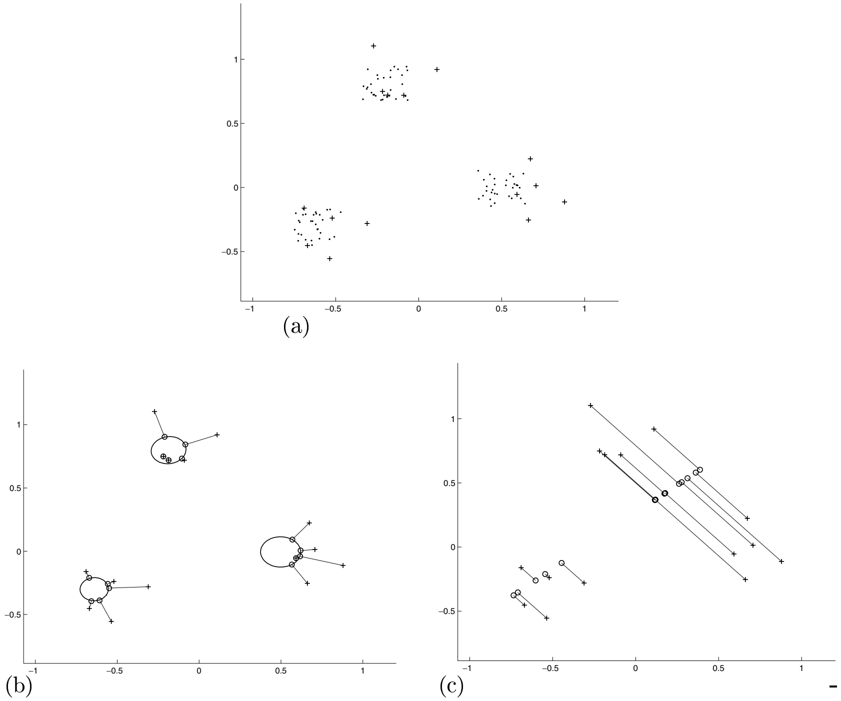

Figure 4: A two-dimensional version of the toy example (with $\sigma_0 = 0.15$) and its denoised results. Lines join each corrupted point (denoted +) with its reconstruction (denoted o). For SVDD, $s^2 = 2 \times 2 \times \sigma_0^2$ and $C = 1/(N \times 0.6)$. (a) Training data (denoted •) and *corrupted* test data (denoted +). (b) Reconstruction using the proposed method (with 100 iterations) along with the resultant SVDD balls. (c) Reconstruction using the PCA-based method, where one principal component was used.

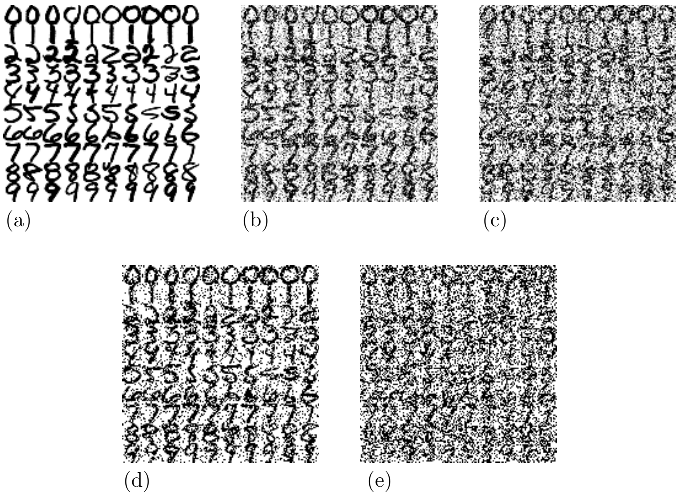

Figure 5: Sample USPS digit images. (a) Clean. (b) With gaussian noise ($\sigma^2 = 0.3$). (c) With gaussian noise ($\sigma^2 = 0.6$). (d) With salt-and-pepper noise ($p = 0.3$). (e) With salt-and-pepper noise ($p = 0.6$).

with clustering-type tasks, so any reconstruction method directly utilizing projection onto low-dimensional linear manifolds would be inefficient.

4.2 Handwritten Digit Data. In this section, we report the denoising results on the USPS digit database (http://www.kernel-machines.org), which consists of 16 × 16 handwritten digits of 0 to 9. We first normalized each feature value to the range [0, 1]. For each digit, we randomly chose 60 examples to form the training set and 100 examples as the test set (see Figure 5). Two types of additive noise were added to the test set.

The first is the gaussian noise $N(0, \sigma^2)$ with variance σ^2, and the second is the so-called salt-and-pepper noise with noise level p, where $p/2$ is the probability that a pixel flips to black or white. Denoising was applied to each digit separately. The width s of the gaussian kernel is set to

$$s_0^2 = \frac{1}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{N} \|\mathbf{x}_i - \mathbf{x}_j\|^2, \tag{4.2}$$

the average squared distance between training patterns. Here, the value of C was set to the effect that the support for the normal class resulting from the SVDD may cover approximately 80% (=100% − 20%) of the training data. Finally, in the third step, we used $n = 10$ neighbors to recover the denoised pattern $\hat{\mathbf{x}}$ by solving the preimage problem.

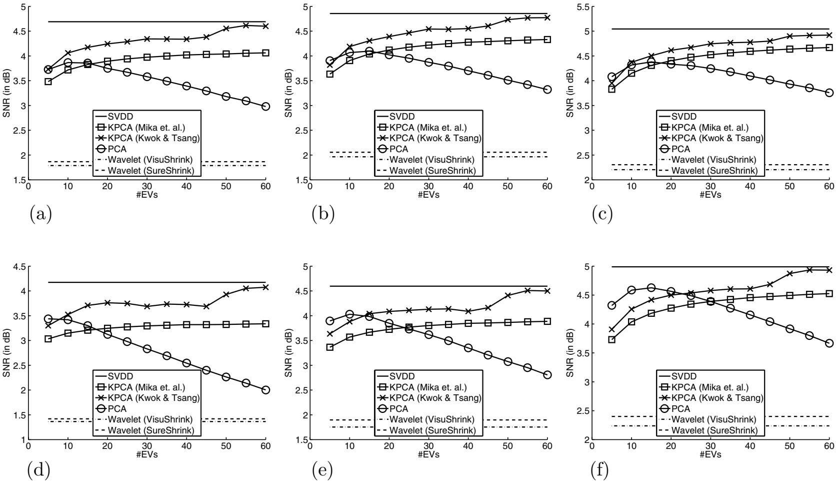

Figure 6: SNRs of the denoised USPS images. (Top) Gaussian noise with variance σ^2. (a) $\sigma^2 = 0.6$. (b) $\sigma^2 = 0.5$. (c) $\sigma^2 = 0.4$. (Bottom) Salt-and-pepper noise with noise level p. (d) $p = 0.6$. (e) $p = 0.5$. (f) $p = 0.4$.

The proposed approach is compared with the following standard methods:

- Kernel PCA denoising, using the preimage finding method in Mika et al. (1999)
- Kernel PCA denoising, using the preimage finding method in Kwok and Tsang (2004)
- Standard (linear) PCA
- Wavelet denoising (using the Wavelet Toolbox in Matlab).

For wavelet denoising, the image is first decomposed into wavelet coefficients using the discrete wavelet transform (Mallat, 1999). These wavelet coefficients are then compared with a given threshold value, and those that are close to zero are shrunk so as to remove the effect of noise in the data. The denoised image is then reconstructed from the shrunken wavelet coefficients by using the inverse discrete wavelet transform. The choice of the threshold value can be important to denoising performance. In the experiments, we use two standard methods to determine the threshold: VisuShrink (Donoho, 1995) and SureShrink (Donoho & Johnstone, 1995). Moreover, the Symlet6 wavelet basis, with two levels of decomposition, is used. The methods of Mika et al. (1999) and Kwok and Tsang (2004) are both based on kernel PCA and require the number of eigenvectors as a

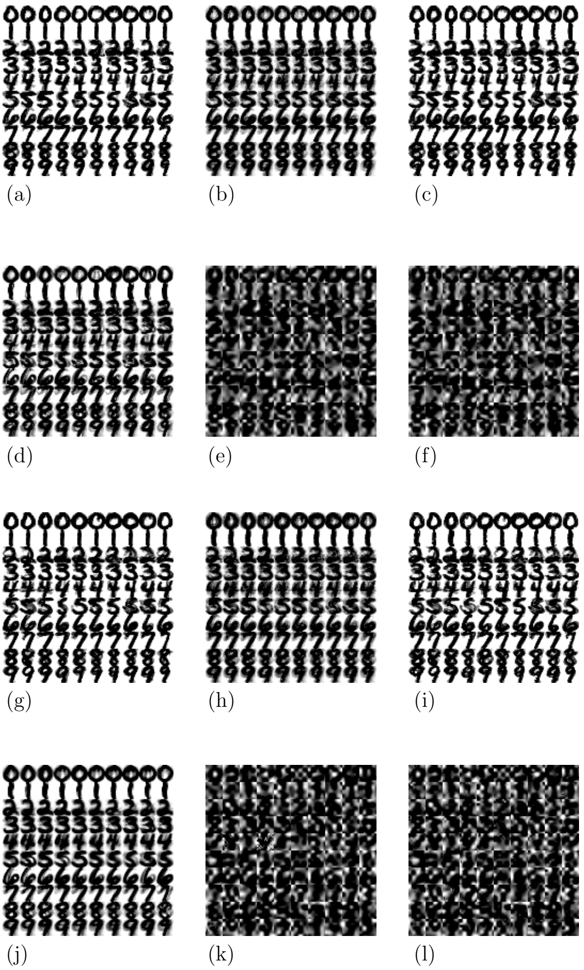

Figure 7: Sample denoised USPS images. (Top two rows) Gaussian noise ($\sigma^2 = 0.6$). (a) SVDD. (b) KPCA (Mika et al., 1999). (c) KPCA (Kwok & Tsang, 2004). (d) PCA. (e) Wavelet (VisuShrink). (f) Wavelet (SureShrink). (Bottom two rows) Salt-and-pepper noise (p = 0.6). (g) SVDD. (h) KPCA (Mika et al., 1999). (i) KPCA (Kwok & Tang, 2004). (j) PCA. (k) Wavelet (VisuShrink). (l) Wavelet (SureShrink).

predetermined parameter. In the experiments, the number of principal components is varied from 5 to 60 (the maximum number of PCA components that can be obtained on this data set). For SVDD, we set $C = \frac{1}{N\nu}$ with ν set to 0.2. For denoising using MDS and SVDD, 10 nearest neighbors are used to perform preimaging.

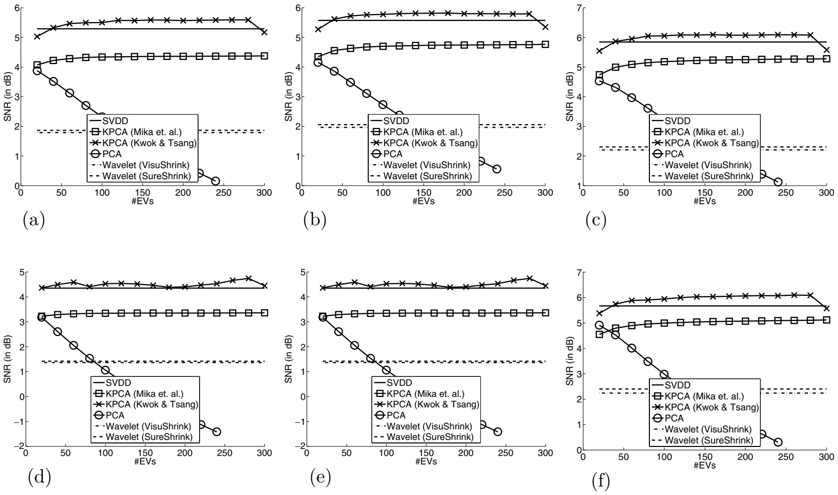

Figure 8: SNRs of the denoised USPS images when 300 samples are chosen from each digit for training. (Top) Gaussian noise with variance σ^2. (a) $\sigma^2 = 0.6$. (b) $\sigma^2 = 0.5$. (c) $\sigma^2 = 0.4$. (Bottom) Salt-and-pepper noise with noise level p. (d) $p = 0.6$. (e) $p = 0.5$. (f) $p = 0.4$.

To quantitatively evaluate the denoised performance, we used the average signal-to-noise ratio (SNR) over the test set images, where the SNR is defined as

$$10 \log_{10} \left(\frac{\text{var(clean image)}}{\text{var(clean image – new image)}} \right),$$

in decibel (dB). Figure 6 shows the (average) SNR values obtained for the various methods. SVDD always achieves the best performance. When more PCs are used, the performance of denoising using kernel PCA increases, while the performance of PCA first increases and then decreases as some noisy PCs are included, which corrupts the resultant images. Note that one advantage of the wavelet denoising methods is that they do not require training. But a subsequent disadvantage is that they cannot utilize the training set, and so both do not perform well here. Samples of the denoised images that correspond to the best setting of each method are shown in Figure 7.

As the performance of denoising using kernel PCA appears improving with the number of PCs, we also experimented with a larger data set so that even more PCs can be used. Here, we followed the same experimental setup except that 300 (instead of 60) examples were randomly chosen from

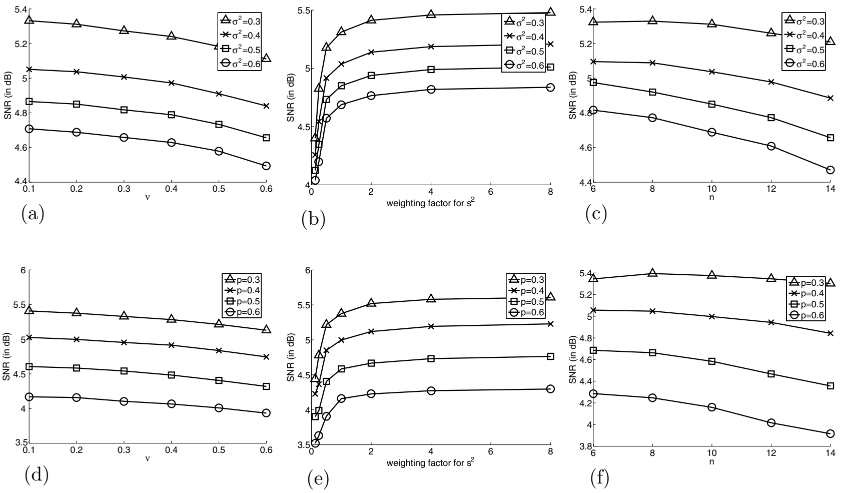

Figure 9: SNR results of the proposed method on varying each of ν, width of the gaussian kernel (s) and the neighborhood size for MDS (n). (Top) Gaussian noise with different σ^2's. (a) Varying ν. (b) Varying s as a factor of s_0 in equation 4.2. (c) Varying n. (Bottom) Salt-and-pepper noise with different p's. (d) Varying ν. (e) Varying s as a factor of s_0 in equation 4.2. (f) Varying n.

each digit to form the training set. Figure 8 shows the SNR values for the various methods. On this larger data set, denoising using kernel PCA does perform better than the others when a suitable number of PCs are chosen. This demonstrates that the proposed denoising procedure is comparatively more effective on small training sets.

In order to investigate the robustness of the proposed method, we also performed experiments using the 60-example training set for a wide range of ν, width of the gaussian kernel (s), and the neighborhood size for MDS (n). Results are reported in Figure 9. The proposed method shows robust performance around the range of parameters used.

In the previous experiments, denoising was applied to each digit separately, which means one must know what the digit is before applying denoising. To investigate how well the proposed method denoises when the true digit is unknown, we follow the same setup but combine all the digits (with a total of 600 digits) for training. Results are shown in Figures 10 and 11. From the visual inspection, one can see that its performance is slightly inferior to that of the separate digit case. Again, SVDD is still the best, though kernel PCA using the preimage method in Kwok and Tsang (2004) sometimes achieves better results as more PCs are included.

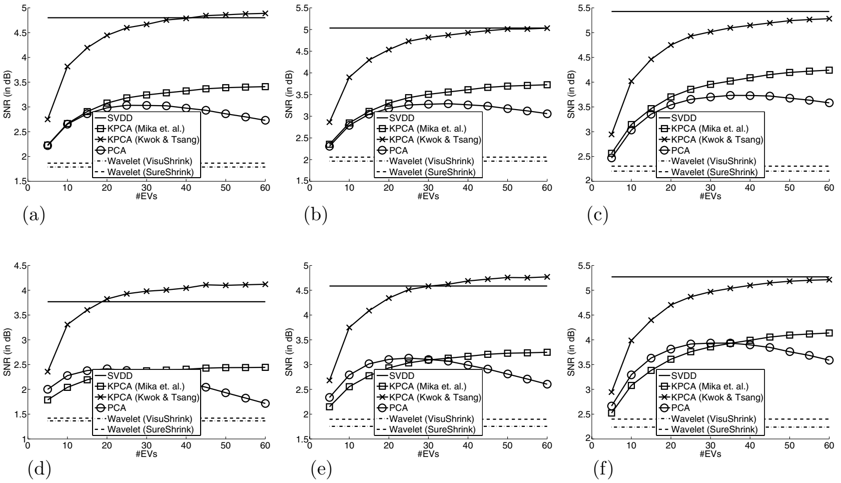

Figure 10: SNRs of the denoised USPS images. Here, the 10 digits are combined during training. (Top) Gaussian noise with variance σ^2's. (a) $\sigma^2 = 0.6$. (b) $\sigma^2 = 0.5$. (c) $\sigma^2 = 0.4$. (Bottom) Salt-and-pepper noise with noise level p's. (d) $p = 0.6$. (e) $p = 0.5$. (f) $p = 0.4$.

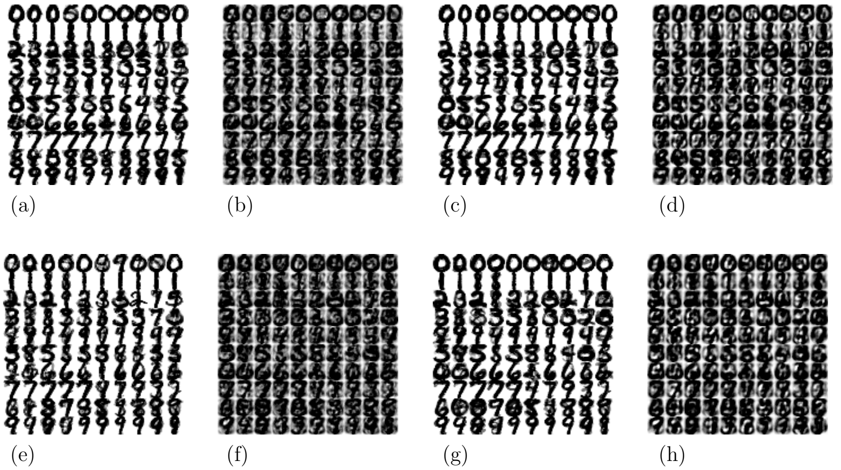

Figure 11: Sample denoised USPS images. The 10 digits are combined during training. Recall that wavelet denoising does not use the training set, and so their de-noising results are the same as those in Figure 7 and are not shown here. (Top) Gaussian noise (with $\sigma^2 = 0.6$). (a) SVDD. (b) KPCA (Mika et al. 1999). (c) KPCA (Kwok & Tsang, 2004). (d) PCA. (Bottom) Salt-and-pepper noise (with $p = 0.6$). (e) SVDD. (f) KPCA (Mika et al. 1999). (g) KPCA (Kwok & Tsang, 2004). (h) PCA.

5 Conclusion

We have addressed the problem of pattern denoising based on the SVDD. Along with a brief review over the SVDD, we presented a new denoising method that uses the SVDD, the geodesic projection of the noisy point to the surface of the SVDD ball in the feature space, and a method for finding the preimage of the denoised feature vectors. Work yet to be done includes more extensive comparative studies, which will reveal the strengths and weaknesses of the proposed method, and refinement of the method for better denoising.

References

Ben-Hur, A., Horn D., Siegelmann H. T., & Vapnik V. (2001). Support vector clustering. *Journal of Machine Learning Research, 2*, 125–137.

Burges, C. J. C. (1999). Geometry and invariance in kernel based methods. In A. J. Smola, P. L. Bartlett, B. Schölkopf, & D. Schuurmons (Eds.), *Advances in kernel methods—support vector learning*. Cambridge, MA: MIT Press.

Campbell, C., & Bennett, K. P. (2001). A linear programming approach to novelty detection. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems, 13*, (pp. 395–401). Cambridge, MA: MIT Press.

Cox, T. F., & Cox, M. A. A. (2001). *Multidimensional scaling* (2nd ed.). London: Chapman & Hall.

Crammer, K., & Chechik, G. (2004). A needle in a haystack: Local one-class optimization. In *Proceedings of the Twenty-First International Conference on Machine Learning*. Banff, Alberta, Canada.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press.

Donoho, D. L. (1995). De-noising by soft-thresholding. *IEEE Transactions on Information Theory, 41*(3), 613–627.

Donoho, D. L., & Johnstone, I. M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *Journal of the American Statistical Association, 90*(432), 1200–1224.

Gower, J. C. (1968). Adding a point to vector diagrams in multivariate analysis. *Biometrika, 55*(3), 582–585.

Kwok, J. T., & Tsang, I. W. (2004). The pre-image problem in kernel methods. *IEEE Transactions on Neural Networks, 15*(6), 1517–1525.

Lanckriet, G. R. G., El Ghaoui, L., & Jordan, M. I. (2003). Robust novelty detection with single-class MPM. In S. Becker, S. Thron, & K. Obermayer (Eds.), *Advances in neural information processing systems, 15*, (pp. 905–912). Cambridge, MA: MIT Press.

Laskov, P., Schäfer, C., & Kotenko, I. (2004). Intrusion detection in unlabeled data with quarter-sphere support vector machines. In *Proceeding of Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA) 2004* (pp. 71–82). Dortmund, Germany.

Mallat, S. G. (1999). *A wavelet tour of signal processing* (2nd ed.). San Diego, CA: Academic Press.

Mika, S., Schölkopf, B., Smola, A. J., Müller, K. R., Scholz, M., & Rätsch, G. (1999). Kernel PCA and de-noising in feature space. In M. S. Kearns, S. Solla, & D. Cohn (Eds.), *Advances in neural information processing systems, 11* (pp. 536–542). Cambridge, MA: MIT Press.

Moon, T. K., & Stirling, W. C. (2000). *Mathematical methods and algorithms for signal processing*. Upper Saddle River, NJ: Prentice Hall.

Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K., & Schölkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks, 12*(2), 181–201.

Pekalska, E., Tax, D. M. J., & Duin, R. P. W. (2003). One-class LP classifiers for dissimilarity representations. In S. Becker, S. Thrun, & K. Obermayer (Eds.) *Advances in neural information processing systems, 15* (pp. 761–768). Cambridge, MA: MIT Press.

Rätsch, G., Mika, S., Schölkopf, B., & Müller, K.-R. (2002). Constructing boosting algorithms from SVMs: An application to one-class classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(9), 1184–1199.

Schölkopf, B., Mika, S., Burges, C., Knirsch, P., Müller, K.-R., Rätsch, G., & Smola, A. J. (1999). Input space vs feature space in kernel-based methods. *IEEE Transactions on Neural Networks, 10*(5), 1000–1017.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computation, 13*(7), 1443–1471.

Schölkopf, B., Platt, J. C., & Smola, A. J. (2000). *Kernel method for percentile feature extraction* (Tech. Rep. MSR-TR-2000-22). Redmond, WA: Microsoft Research.

Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels*, Cambridge, MA: MIT Press.

Tax, D. (2001). *One-class classification*. Unpublished doctoral/dissertation, Delft University of Technology.

Tax, D., & Duin, R. (1999). Support vector data description. *Pattern Recognition Letters, 20*(11–13), 1191–1199.

Williams, C. K. I. (2002). On a connection between kernel PCA and metric multidimensional scaling. *Machine Learning, 46*(1–3), 11–19.

Received August 31, 2005; accepted October 30, 2006.