Pattern Recognition 44 (2011) 1698–1707

Contents lists available at ScienceDirect

Pattern Recognition

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

Dynamic pattern denoising method using multi-basin system with kernels

Kyu-Hwan Jung, Namhyoung Kim, Jaewook Lee *

Department of Industrial and Management Engineering, Pohang University of Science and Technology, 790-784, Pohang, Kyungbuk, South Korea

ARTICLE INFO

Article history:
Received 31 May 2010
Received in revised form
1 February 2011
Accepted 3 February 2011
Available online 3 March 2011

Keywords:
Kernel methods
Support vector machines
Pattern denoising
Image restoration
Dynamical system

ABSTRACT

In this paper, we propose a novel pattern denoising method that utilizes the topological property of a support that describes the distribution of normal patterns to denoise noisy patterns. The method first trains a support function which captures the domain of normal patterns and then construct a so-called multi-basin system associated with the trained support function. By moving noisy patterns along the trajectories of the multi-basin system, noise is removed while the pattern recovers its normality. The denoised pattern is obtained when the noisy pattern arrives at the attracting manifold generated by a set of normal patterns and this is the most similar normal pattern with the noisy pattern in the topological sense. Through simulations on some toy dataset and real image datasets, we show that the proposed framework effectively removes the noise while preserving the information contained in the noisy pattern.

1. Introduction

In the past decade, learning techniques with kernels have been widely studied to solve various problems in machine learning and have expedited the development of many successful intelligent systems in various applications. Among them, support (or pseudo-density) learning methods with kernels have been successfully applied to pattern recognition tasks such as outlier or novelty detection [1,2], data retrieval [3,4], clustering [5–9,14,13] and classification [10,16]. The advantages of these support learning methods with kernels over conventional support estimation methods are their flexibility to capture a support of arbitrary shape by utilizing the kernel mapping into the feature space and insensitivity to the outliers by following the structural risk minimization principle [6,17,14].

Generally, the role of a support function learned by existing methods that describes the domain of a data distributions is limited to separating novelties or outliers from normal patterns. However, when we have a prior knowledge that the outlier patterns are actually noised or corrupted version of normal patterns, we can reconstruct the denoised version of the noisy patterns by utilizing the domain information of normal patterns described by the support function and by associating outlier patterns with their corresponding normal patterns that are determined by the geometrical and topological affinities of the constructed support.

To accomplish this task, in this paper, we propose a novel pattern denoising method consisting of three steps: the first step to build a support function that describes a support of normal patterns, the second step to construct a so-called multi-basin system associated with the support function, and the final third step to denoise by moving given noisy pattern along the trajectory of multi-basin system. The denoised pattern obtained by this process is topologically most similar normal pattern with the original noisy pattern. We will show the effectiveness of the proposed method by demonstrating denoising performance on real image dataset such as USPS (United States Portal Service, [18]) handwritten digits, COIL20 object image library dataset [19] and C-Cube cursive character database [27] as well as a synthetic toy dataset to compare the results with various related methods.

The remainder of the paper is organized as follows: In Section 2, we provide the proposed dynamical denoising framework using a multi-basin system with theoretical analysis and give its detailed algorithms in Section 3. In Section 4, we demonstrate the effectiveness of the proposed method by comparing the experimental results with other denoising techniques in a number of toy and real-world problems. We conclude the paper in Section 5 with discussions.

2. The proposed method

Given a set of normal patterns $\mathcal{D}=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N\}$, the proposed method, which is detailed below, consists of three phases; the first phase for constructing a support function, the second phase for building the multi-basin system associated with the support function, and the final third step for applying the system to the noisy pattern to obtain its denoised version.

* Corresponding author.
E-mail address: jaewookl@postech.ac.kr (J. Lee).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.02.004

2.1. Learning a support of the data distribution

A *support function (or quantile function)* is roughly defined as a positive scalar function $q:\Re^n\to\Re^+$ where a level set of q estimates a support of a data distribution. (For a mathematical definition of a quantile, see [2].) With the aid of the support function, q, a support of a data distribution can then be decomposed into several separate connected components Λ_i, $i=1,\ldots,m$, i.e.

$$L_q(r) := \{\mathbf{x}\in\Re^n : q(\mathbf{x})<r\} = \Lambda_1\cup\cdots\cup\Lambda_m \tag{1}$$

where $r=q(\mathbf{x}^*)>0$ is a critical level value for some $\mathbf{x}^*\in\Re^n$, obtained by parameter estimation or tuning. Each Λ_i, $i=1,\ldots,m$ is also called a *feasible domain* of (1). Popular support functions that are trained from data and are shown to characterize the support (or quantile) of a class-conditional distribution include:

- A support function generated by the support vector domain description (SVDD) method with Gaussian kernel in [1,6,10],

$$q(\mathbf{x}) = 1-2\sum_{j\in J}\alpha_j e^{-v\|\mathbf{x}-\mathbf{x}_j\|^2} + \sum_{i,j\in J}\alpha_i\alpha_j e^{-v\|\mathbf{x}_i-\mathbf{x}_j\|^2} \tag{2}$$

  where $\mathbf{x}_j$ and α_j for $j\in J$ are support vectors and their corresponding coefficients are constructed by optimizing the SVDD model and a critical level value $r=q(\mathbf{x}_i)$ for any support vector $\mathbf{x}_j$. v denotes the parameter of the Gaussian kernel $k(\mathbf{x}_i,\mathbf{x}_j)=e^{-v\|\mathbf{x}_i-\mathbf{x}_2\|^2}$.
- A Gaussian process support function generated by the Gaussian process clustering in [9,14],

$$q(\mathbf{x}) = \mathbf{k}(\mathbf{x})^T\mathbf{C}^{-1}\mathbf{k}(\mathbf{x}) \tag{3}$$

  where $\mathbf{C}$ is a positive definite covariance matrix with elements $C_{ij}=C(\mathbf{x}_i,\mathbf{x}_j;\Theta)$ and $\mathbf{k}(\mathbf{x})=(C(\mathbf{x},\mathbf{x}_1),\ldots,C(\mathbf{x},\mathbf{x}_N))^T$, One commonly used covariance function is

$$C(\mathbf{x}_i,\mathbf{x}_j;\Theta) = v_0\exp\left\{-\frac{1}{2}\sum_{m=1}^{n} l_m(x_i^m-x_j^m)^2\right\}+v_1+\delta_{ij}v_2$$

  where a set of hyper-parameters, $\Theta=\{v_0,v_1,v_2,l_1,\ldots,l_n\}$ and a critical level value r, can be determined by maximizing the marginal likelihood [14].
- Parzen window support function generated by kernel density estimation technique as in [15].

$$q(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n} k\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right) \tag{4}$$

  where h is the bandwidth parameter and k bounded function with compact support, i.e.

$$\int_{R^d} k(\mathbf{x})\,d\mathbf{x}=1 \quad \lim_{\|\mathbf{x}\|\to\infty}\|\mathbf{x}\|^d k(\mathbf{x})=0 \tag{5}$$

$$\int_{R^d}\mathbf{x}k(\mathbf{x})\,d\mathbf{x}=0 \quad \int_{R^d}\mathbf{x}\mathbf{x}^T k(\mathbf{x})\,d\mathbf{x}=cI \tag{6}$$

  with constant c. Even though each support functions successfully applied to various pattern recognition techniques (SVDD: [6–13], GP:[14], Parzen window:[15]), support function learned by SVDD generally outperforms other support learning method in its robustness and complexity. As shown in Eq. (2), the support function of SVDD is expressed by summation of kernel functions evaluated only with support vectors, the number of which is generally small portion of training set. On the contrary, the support functions of GP and Parzen window are expressed with all training points and this leads to higher computational burden in computing the optimal parameters and the mode of estimated support function. In [9], the clustering performance of SVDD and GP support function is directly compared and SVDD shows better performance both in accuracy and time. Therefore, we used support function learned by SVDD unless otherwise stated.

2.2. Multi-basin system associated with a learned support function

Multi-basin strategy aims at searching a 'normal' point on a feasible domain (or a connected component) of (1) corresponding to (and similar to) a noisy data point. The key idea is as follows: if there exists a proper dynamical system such that all the feasible domains are its attractors (whose nearby trajectories approach it asymptotically), then we can find a normal point in a connected component which is a denoised version of given noisy point by finding an attractor of the corresponding dynamical system.

To implement this idea, we build the following non-hyperbolic dynamical system (cf. [20]) associated with the support function, q, and its critical level value, r:

$$\frac{d\mathbf{x}}{dt} = F_q(\mathbf{x}) := -q_r(\mathbf{x})\nabla q(\mathbf{x}) \tag{7}$$

where $q_r(\cdot)$ is defined by

$$q_r(\mathbf{x}) = \frac{e^{\alpha(q(\mathbf{x})-r)}}{1+e^{\alpha(q(\mathbf{x})-r)}} \simeq 1_{\{q(\mathbf{x})\geq r\}}$$

where $\alpha>1$ is a constant that affects the asymptotic behavior of the transformation and $1_{\{q(\mathbf{x})\geq r\}}$ is the indicator function having value one if $q(\mathbf{x})>r$ and zero otherwise. This choice of q_r combined with the assumption that q is twice differentiable guarantees the existence of a unique solution (or trajectory) $\mathbf{x}(\cdot):\Re\to\Re^n$ for each initial condition $\mathbf{x}(0)$. (Note that the support function q of our interest as such in (2) or (3) is twice differentiable.)

A connected component, say Λ, of $F_q^{-1}(0)$, consisting of non-isolated equilibrium points is called an *equilibrium manifold* of system (7). An equilibrium manifold Λ is called an *attracting manifold*, if δ can be chosen such that

$$\mathbf{x}_0\in N_\delta(\Lambda)\Rightarrow\lim_{t\to\infty}\mathbf{x}(t)\in\Lambda$$

where $N_\delta(\Lambda)=\{x\in\Re^n : \|x-y\|<\delta,\ \forall y\in\Lambda\}$ is an open neighborhood of Λ with size δ. Notice that each connected component of $L_q(r)=\{\mathbf{x}\in\Re^n : q(\mathbf{x})<r\}$ corresponds to an equilibrium manifold of system (7).

Formally, the *basin* of an attracting equilibrium manifold, Λ_s, is defined as

$$B(\Lambda_s) := \{x\in\Re^n : \lim_{t\to\infty}\mathbf{x}(t)\in\Lambda_s\}.$$

The next theorem establishes a result showing the relationship between feasible domains of (1) and attractors of (7), which is a distinguished feature of system (7).

Theorem 1. *Let Λ be a feasible domain of $L_q(r)=\{\mathbf{x}\in\Re^n : q(\mathbf{x})<r\}$ (i.e. a feasible connected component of $L_q(r)$). Then Λ is an attracting manifold of system* (7).

Proof. See Appendix

Theorem 1 asserts that we can locate a normal point corresponding to a noisy data point by applying system (7) and by locating an attractor to which it converges.

The next theorem establishes so-called the multi-basin theorem for system (7): the whole data space is composed of multiple basins of attracting manifolds. (See Fig. 1.)

Theorem 2 (*Multi-basin theorem*). *Suppose that all the stable equilibrium vectors of system $\dot{\mathbf{x}}=-\nabla q(\mathbf{x})$ are inside the feasible domain $L_q(r)=\{\mathbf{x}\in\Re^n : q(\mathbf{x})<r\}$. Then the whole data space $\Re^n$ can*

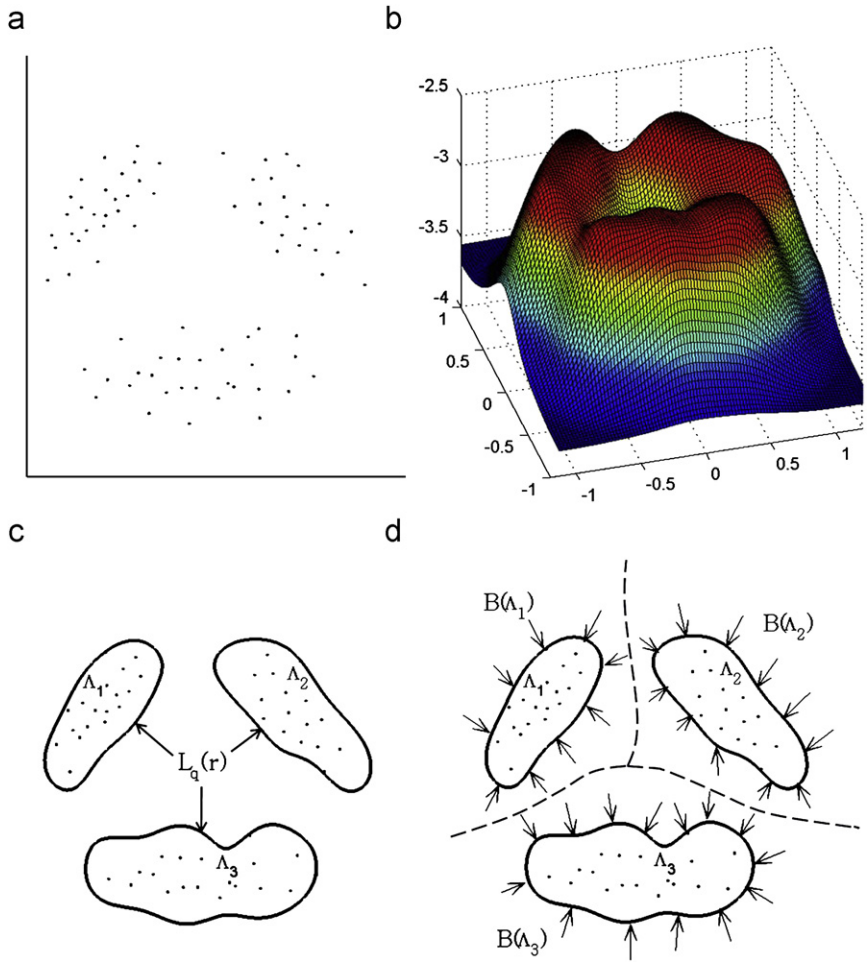

Fig. 1. Illustration of the multi-basin systems. (a) Data points, (b) support function value of input space, (c) support level set and (d) attracting manifolds.

be decomposed into several separate basins of the attracting manifolds of system (7), *i.e.*

$$\Re^n = \bigcup_{i=1}^{m} B(\Lambda_i) \tag{8}$$

Proof. See Appendix

2.3. Denoising via a constructed multi-basin system

As is explained above, we start by building a support function, say $q(\cdot)$, that estimates a support of data distribution. Since the constructed support function is able to measure the degree of normality, we can determine whether a new pattern is normal or not. Next to remove the noise and recover its normality, we construct a multi-basin system associated with the support function q. The constructed multi-basin system decomposes the whole data space into the multiple disjoint basins of attracting manifolds where each attracting manifold attracts every pattern resides within its basin (i.e. a noisy test pattern $\tilde{\mathbf{x}}$ is attracted to the normal region). Then the noisy test pattern $\tilde{\mathbf{x}}$, when applied to the multi-basin system, moves along the trajectory until it converges to a boundary point, say $\hat{x}_D$, of its attracting manifold. The obtained point $\hat{x}_D$ becomes a resulting normal pattern that turns out to be very similar to the noisy pattern $\tilde{\mathbf{x}}$.

Fig. 2 illustrates the method presented in the previous section: after training a support function that describes the distribution of normal patterns, we construct a multi-basin system associated with the support function. Since a noisy pattern $\tilde{\mathbf{x}}$ resides within a basin $B(\Lambda)$ of its corresponding attracting manifold Λ, it moves toward Λ until it arrives at a boundary point, $\hat{x}_D$, on Λ. The resulting boundary point is then the denoised version of $\tilde{\mathbf{x}}$ obtained by the proposed method.

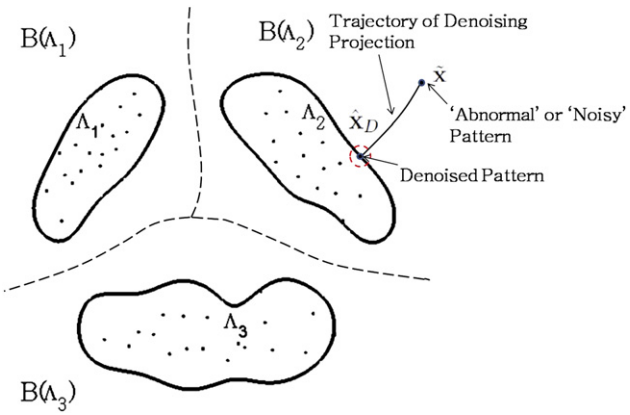

Fig. 2. Illustration of the proposed denoising framework. See the text for more detail.

3. Algorithm and implementation

The simplified algorithm for the proposed method is given as follows.

Algorithm 1. Attracting Manifold Methods for Dynamical Pattern Denoising

//A1. Training Support Function//
Given a set of normal patterns $\mathcal{D}=\{\mathbf{x}_k\}_{k=1}^{N}$ and critical level value r
Train a support function $q(\mathbf{x})$

//A2. Constructing Multi-Basin System//
Construct the multi-basin system associated with $q(\mathbf{x})$ by defining F_q in (7).

//A3. Numerical Integration for Denoising//
For a given noisy pattern $\tilde{\mathbf{x}}$
Set $\hat{\mathbf{x}}_0=\tilde{\mathbf{x}}$ and $t=0$
while $q_r(\hat{\mathbf{x}}_{t+1})>r$ **do**
  Numerically integrate the dynamical system (7) one step as $\hat{\mathbf{x}}_{t+1}=\hat{\mathbf{x}}_t+F_q(\hat{\mathbf{x}}_t)$;
  $t=t+1$;
end while
$\hat{\mathbf{x}}_{in}=\hat{\mathbf{x}}_{t+1}$;
$\hat{\mathbf{x}}_{out}=\hat{\mathbf{x}}_t$;

//A4. Linear Search to Find Optimal Denoised Pattern//
Set $\rho=0$, $\hat{\mathbf{x}}_{temp}=\hat{\mathbf{x}}_{out}$;
while $q(\hat{\mathbf{x}}_{temp})>r$ **do**
  $\rho=\rho+0.1$;
  $\hat{\mathbf{x}}_{temp}=(1-\rho)\hat{\mathbf{x}}_{out}+\rho\hat{\mathbf{x}}_{in}$;
end while
$\hat{\mathbf{x}}_D=\hat{\mathbf{x}}_{temp}$;

In algorithm part *A*1, we train support function by selecting a support learning algorithms stated in Section 2.1. After training the support function, we construct multi-basin system in algorithm part *A*2 by selecting a numerical optimization technique. The theoretical analysis selecting numerical optimization technique can be found in [9]. While looping algorithm part *A*3, we obtain a pattern almost surely reside within the support boundary. To obtain a most similar normal pattern with the original noisy pattern, we move backward until it is most close to the boundary in algorithm part *A*4.

Some remarks of the proposed methods related to some implementation issues are given below.

- The proposed denoising framework can be generally combined with any support functions learning method which captures well the support of normal patterns.
- The proposed framework is not restricted to be used with specific kernels. As long as it can be adopted to the candidate support function, we can use any type of kernels.
- We do not need any prior knowledge of the noise. This is important since most the image denoising algorithms assume specific noise model and show poor performance when the assumption is not satisfied [21,22].
- The run time complexity of the proposed method is linear to the number of training normal patterns. The only necessary computation is the evaluation of support function value and its gradient which involves N_f times of kernel function evaluation. Here, N_f is the number of vectors that appear in the trained support function where for SVDD, N_f is the number of support vector as shown in Eq. (2) and for GP, N_f is the number of training patterns as shown in (3).
- We do not necessarily integrate the system (7) until it reaches to the attracting manifold all the time. When the task is just to recognize and categorize a noisy image visually, we can gradually denoise a noisy pattern only up to the necessary level by monitoring the results which will save the denoising time. We will show the intermediated denoised patterns in Section 4 to visually investigate this point.
- The step length parameter ρ in algorithm part *A*4 controls both the speed and quality. Even though bigger value results in mixtured denosied pattern and smaller value increases line search time, we found through empirical analysis that the performance is not significantly sensitive if the value is set near to 0.1.

We note that a denoising method using SVDD (D-SVDD) has been proposed in [23]. They train SVDD to find a ball which encloses the normal patterns in the feature space and project the noisy pattern onto the surface of the ball. The preimage of this projected point is the denoised pattern in the input space. However, to find the preimage, they use singular value decomposition which costs $O(N^3)$ where N is the number of training normal patterns. They also use multi-dimensional scaling which introduces another control variable *nb*, the number of neighborhood to find MDS solution. As shown in [23], *nb* apparently affects the denoising performance and to use MDS, the feasible kernel-type is restricted to only invariant kernels such as Gaussian kernel. In the experiment, we included D-SVDD as one of the compared methods with the proposed denoising framework.

4. Experiments

In this section, we demonstrate the experimental results by applying the proposed method to simple toy example and real image denoising problems. We compare the denoising performance of the proposed method with other denoising methods both visually and quantitatively.

4.1. Toy example

To demonstrate the effectiveness of the proposed method visually, we illustrate the denoising procedure on simple 2D toy example. We generated 2D points which are considered to be normal patterns and a number of test points which are distant from training points are generated as the corrupted patterns (see Fig. 3(a)). We then trained support functions which describes the region of normal patterns using SVDD and GP. The support of both functions and contour of the level value is shown in Fig. 3(b) and (d), respectively. For both support functions, four SEVs are constructed which attracts the noisy pattern to the region of normal pattern. We used simple steepest descent to apply the proposed denoising method and the trajectory of denoising process for each noisy pattern is shown in Fig. 3(c) and (e). We can observe that the proposed method effectively denoise the test noisy patterns using both support functions.

4.2. USPS dataset

The dataset used in this experiment is well-known USPS handwritten digit dataset [18]. The USPS dataset contains 16×16 grayscale images of numbers written on the postal materials and thereby the dimension of each pattern is 256 when it is vectorized. Originally the value used in gray scale is 0–255 where 0 is for black and 255 for white. We normalized the values to be 0–1 for training. There are 7261 training patterns and 2007 for test. For each of the 10 digits, we randomly chose 50 patterns from the training set and combined them to be 500 patterns to train the support function and 10 patterns from the test set (100 patterns in total) to make noisy patterns. To generate noisy test patterns, we added Gaussian noise with zero mean and standard deviation $\sigma=0.5$. We used SVDD to train the support function

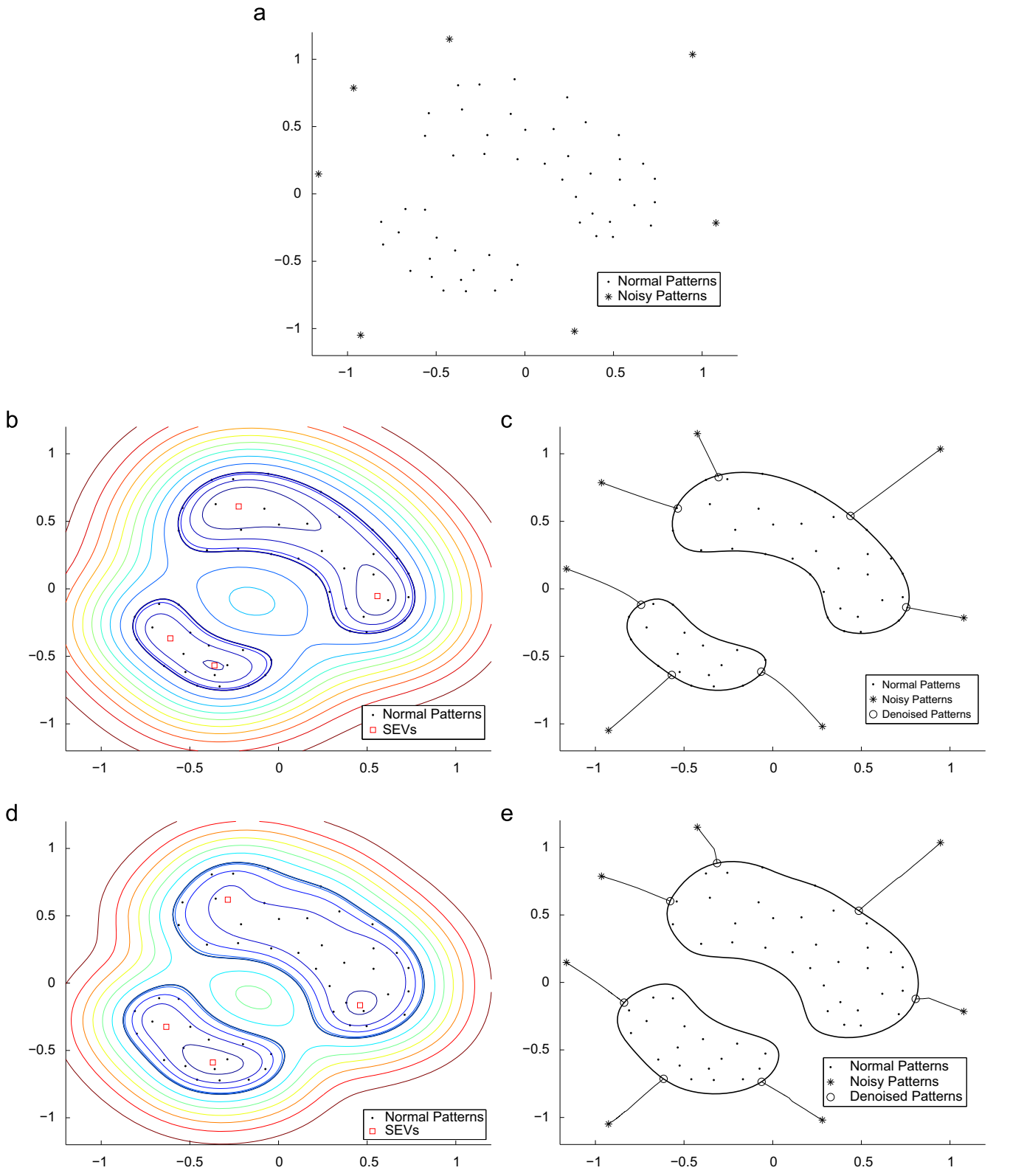

Fig. 3. Illustration of the proposed method on simple 2D toy example. (a) The normal and noisy patterns, (b) the multi-basin system constructed using SVDD, (c) the denoising trajectory and denoised pattern using SVDD support function, (d) the multi-basin system constructed using GP and (e) the denoising trajectory and denoised pattern using GP support function.

and with Gaussian kernel $k(x_i,x_j)=\exp(v\|x_i-x_j\|^2)$. GP is excluded as the support function here since it generally fails to capture the support of high-dimensional dataset and hence shows poor performance. The Gaussian kernel parameter was set to $v=2$ by considering the average variance of data in each dimension as used in [24] and regularization constant was set to and $C=0.7$ to allow some bounded support vectors because there are misleading non-typical patterns which degrades the quality of support function. We used same kernel parameter for both D-SVDD and the proposed method.

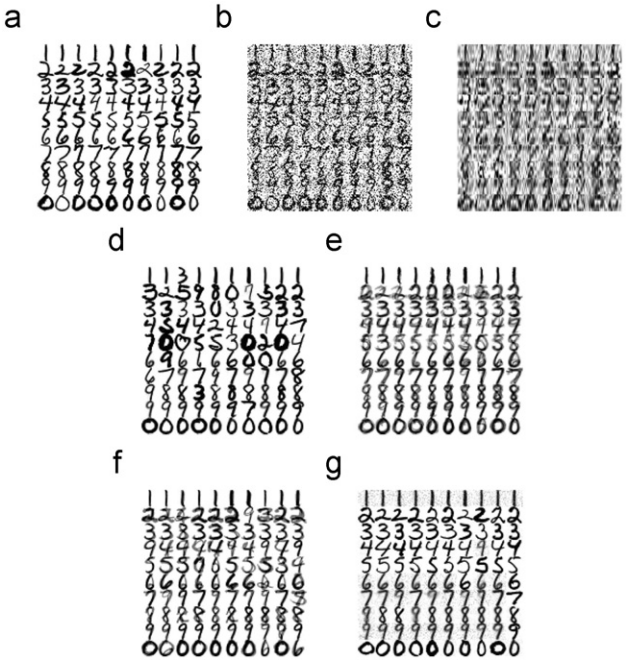

Fig. 4. Denoised images of USPS dataset. (a) Original USPS dataset, (b) noised images, (c) wavelet, (d) KPCA (Mika), (e) KPCA (Kwok), (f) D-SVDD and (g) multi-basin (proposed).

To compare the performance with that of the other denoising methods as baseline, we also experimented with the following methods:

- Kernel PCA proposed by Mika et al. [24].
- Kernel PCA proposed by Kwok & Tsang [25].
- Wavelet method [26].
- SVDD-based Denoising [23].

Since Kernel PCA (KPCA) methods require us to determine the number of eigenvectors (EVs) to be used for approximation, we varied the number of EVs accordingly where the maximum possible number of EV is the number of training data. For wavelet, fixed form threshold was used with estimation of level noise and thresholding mode was soft [26]. The approximation coefficients were kept and we denoised the images using global thresholding option.

The experimental results are presented visually in Fig. 4. In the figure, (a) is the original test images ordered vertically 1–0 before adding noise and (b) is their corrupted version with Gaussian noise. The result with wavelet in (c) is hardly recognized as the corresponding digits and shows significant level of artifacts. (d) is the denoised images by KPCA (Mika) using 500 eigenvectors and even though the images are cleaned to be easily recognizable, they are frequently recognized as different digit from their original one which is very severe drawback for denoising purpose. In (e), the denoised images by KPCA (Kwok) using 500 eigenvectors seems to be mixture of similar digits and visual quality is not satisfactory. The denoised images by D-SVDD is given in (f). We used $nb=2$ number of neighborhood and even though the noises are effectively reduced, the denoised images also seem to be mixture of numbers. In (g), the denoised image by proposed method is given and we can observe that the denoised images are easily recognizable as the corresponding numbers even though some level of noises are still remained. In Fig. 5 a sample of images which are on the denoising trajectory of proposed method is given. The trajectory starts from the leftmost noised image and ends at the rightmost denoised image. As shown in the figure, the noise is gradually removed as the pattern moves along the trajectory and the third image from the left is already easily recognizable as the correct number.

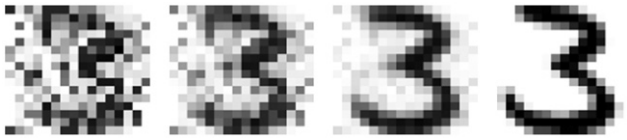

Fig. 5. The sample images on the trajectory of denoising procedure using proposed dynamical pattern denoising method.

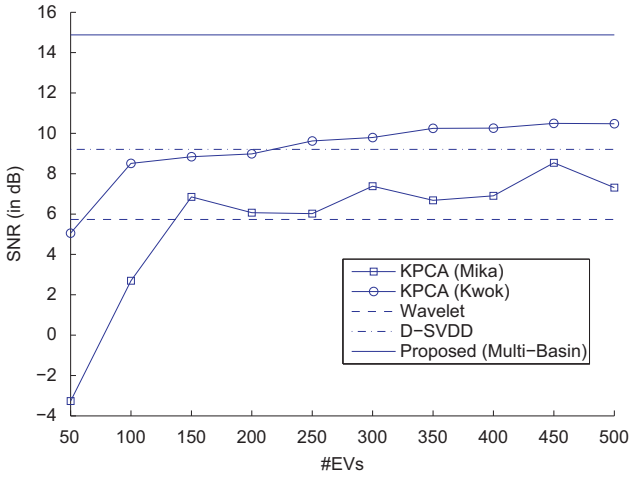

Fig. 6. Comparison of signal-to-ratio of denoising methods on USPS dataset.

To quantitatively compare the performance with other methods, we computed the average signal-to-noise ratio(SNR) for each denoising method where SNR in decibel(dB) is defined as

$$10\log_{10}\left(\frac{\text{var(original image)}}{\text{var(original image}-\text{denoised image)}}\right). \tag{9}$$

The results are given in Fig. 6. As it can be seen in figure, the proposed method shows significantly better results than the other methods. The performance of KPCA increases as the number of EV is increased while the performance gain is marginal after the number of EV is more that 150. KPCA (Kwok) outperforms than KPCA (Mika) and D-SVDD showed similar performance with KPCA (Kwok). Wavelet shows poor performance both visually and quantitatively since there is no training phase for Wavelet and, therefore, it cannot utilize the information of normal patterns in the training set.

4.3. COIL20 dataset

The COIL-20 dataset is a database of images of 20 objects recorded under 72 different viewing angles. Each pattern is 32×32 grayscale image and thereby the dimension of each pattern is 1024 when vectorized. We randomly chose two images (2 angles) from each object for test set and used remaining 70 images (70 different angle) of each object as the training set. The Gaussian noise with mean zero and standard deviation $\sigma=0.6$ is added to the test images. The sample test images and its denoised images using various methods are shown in Fig. 7. (a) is the original object images and (b) is its noised version with Gaussian noise. (c) is the denoised image by wavelet which is not satisfactory. (d) and (e) are the denoised images by KPCA (Mika) and KPCA (Kwok), respectively, using 70 EVs. While KPCA (Mika) generates clean image in different angle, KPCA (Kwok) generates

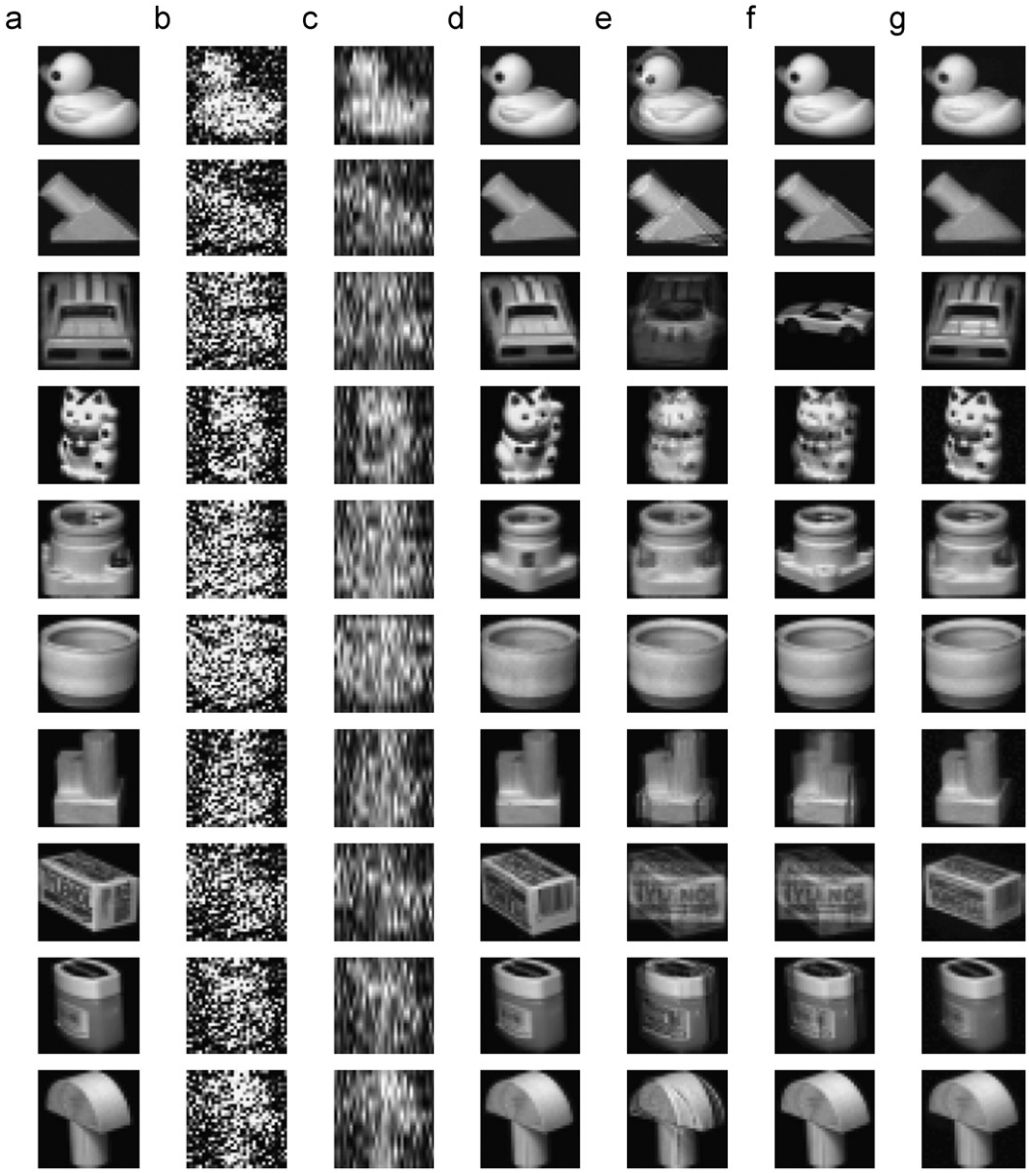

Fig. 7. Denoised images of COIL20 dataset. (a) Original object images, (b) noised images, (c) wavelet, (d) KPCA (Mika), (e) KPCA (Kwok), (f) D-SVDD and (g) multi-basin (proposed).

overlapped images. The result of D-SVDD in (f) is generally satisfactory while there are some images in the different angle. (g) is the denoising result of the proposed method where we used $v=1.5$ and $C=0.5$ with the same parameter selection strategy used in USPS case. As shown in the figure, the noise is effectively removed and the resultant images are most visually most satisfactory.

In Fig. 8 a sample of images on the denoising trajectory is given. As shown in the figure, the noise is effectively removed as the pattern moves along the trajectory and the object is sufficiently recognizable at the level of third image from left. In practice, we can save the denoising time by monitoring the intermediate images of the denoising process and terminating the process when pre-determined criterion is satisfied.

For quantitative analysis of the denoising performance, we compared the average SNR. As shown in the Fig. 9, the proposed method outperforms other denoising methods while the performance of KPCA converges to that of proposed method when the number of EV used is close to the maximum. D-SVDD shows generally satisfactory performance while we needed to tune the parameter *nb* for this performance.

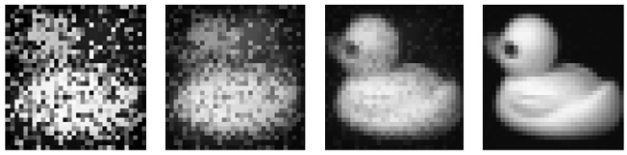

Fig. 8. The sample images on the trajectory of denoising procedure using proposed dynamical pattern denoising method.

4.4. Cursive character dataset

In this experiment, we used C-Cube cursive character database which can be downloaded from http://ccc.idiap.ch. The cursive characters in this database were extracted from handwritten words which had been collected from postal service in United States. The words were separated into individual characters by

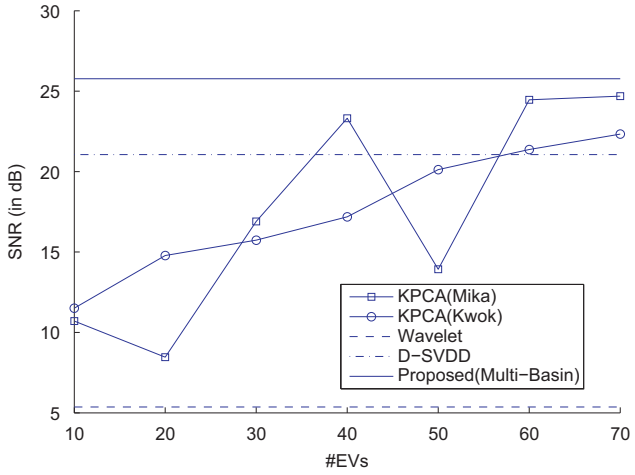

Fig. 9. Comparison of signal-to-ratio of denoising methods on COIL20 dataset.

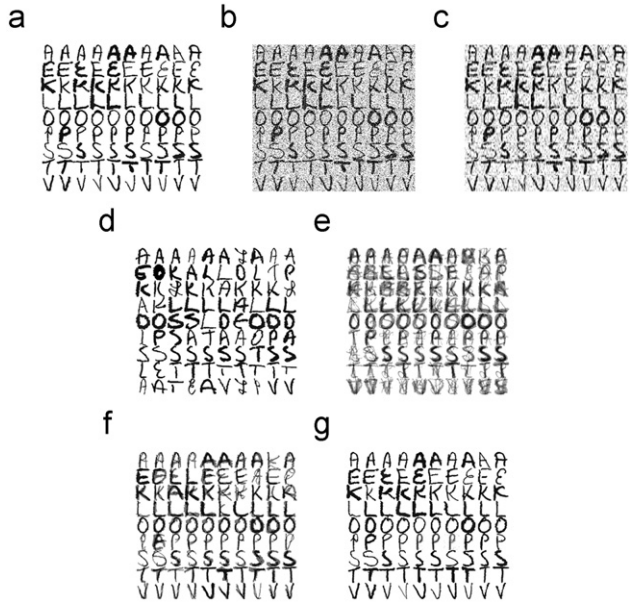

Fig. 10. Denoised images of C-Cube dataset. (a) Original object images, (b) noised images, (c) wavelet, (d) KPCA (Mika), (e) KPCA (Kwok), (f) D-SVDD and (g) multi-basin (proposed).

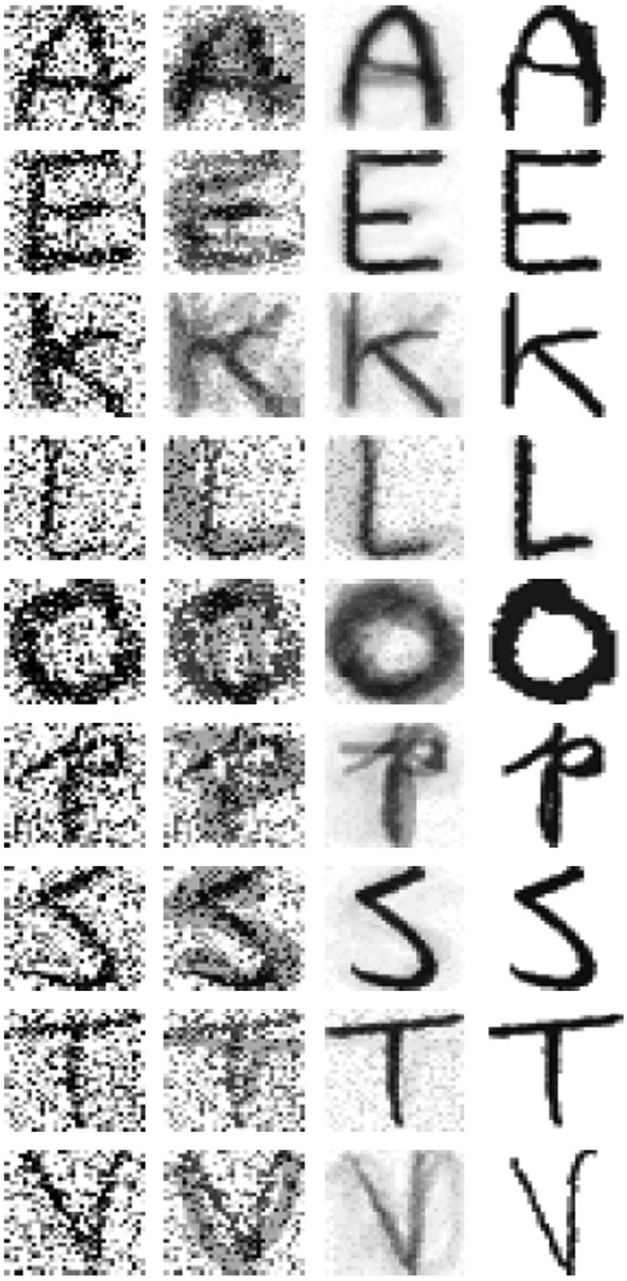

Fig. 11. The sample images on the trajectory of denoising procedure using proposed dynamical pattern denoising method.

desloping, deslanting and segmentation process and as a result, database of 57 293 characters (38 160 for training set and 19 133 for test set) was obtained. Detailed information on C-Cube including the distribution of characters and preliminary experimental results are available in [27,28]. Since the each character of C-Cube have different size and encoded in binary bitmap, we normalized the characters to be grayscale images of 32×32 pixel sizes. To fit in the scale of experiment in our experimental setting, we arbitrarily selected nine capital characters 'A', 'E', 'K', 'L', 'O', 'P', 'S', 'T' and 'V'. For each character, we used 300 examples from training set (in total 2700 examples) to train support function with $v=2$ and $C=1$. We randomly chose 50 examples of each character from test set to generate noised image with Gaussian noise level $\sigma=0.6$. The sample test images and its denoised versions using various methods are shown in Fig. 10. (a) is the original cursive characters and (b) is its noised version with Gaussian noise. (c) is the denoised image by wavelet. (d) and (e) are the denoised image by KPCA (Mika) and KPCA (Kwok), respectively, using 300 EVs. While KPCA (Mika) generates clean wrong characters, KPCA (Kwok) generates overlapped characters. While the result of D-SVDD in (f) is recognizable as corresponding character, the images are still overlapped. (g) is the denoising result of the proposed method. As shown in the figure, the noise is effectively removed and the resultant images are most visually most satisfactory.

In Fig. 11 samples images of each character on the denoising trajectory is given. As we can observe in the figure, the noise is effectively removed when the pattern moves along the denoising trajectory and the characters are sufficiently clear at the level of third image from left.

For quantitatively analyzing the denoising performance, average SNR values are compared. As shown in the Fig. 12, the proposed method significantly outperforms other denoising methods while the performance of wavelet is slightly better than the other methods in this dataset followed by D-SVDD.

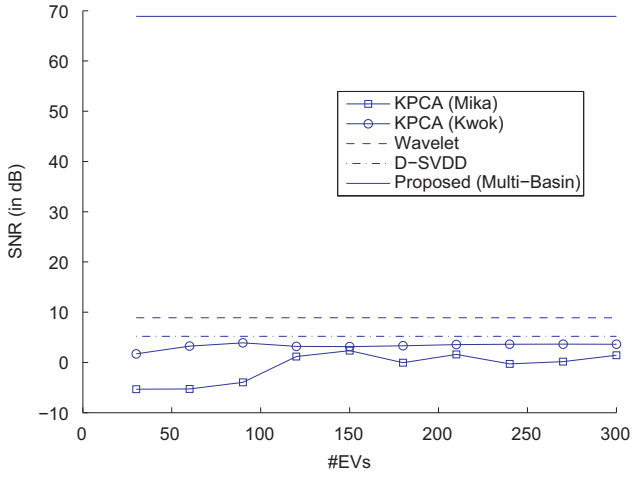

Fig. 12. Comparison of signal-to-ratio of denoising methods on C-Cube dataset.

Altogether with the qualitative and quantitative investigation in three real image dataset, we can observe that the proposed method effectively removes noise of test pattern and accurately finds the most similar normal pattern with the given noisy pattern.

5. Conclusions

In this paper, we proposed a new denoising framework using topological behavior of a noisy pattern in the multi-basin system constructed by support functions. To do that, we first constructed a support function trained from a given normal dataset and built a multi-basin system associated with the learned support function. Then we moved a noisy pattern along the trajectory of the multi-basin system until it arrives to the boundary of the attracting manifold. Through the simulations, we demonstrated that the proposed denoising method effectively reduces the noise while preserving the information contained in the noisy data. Further works include refinement of numerical procedure to find denoised point and using other type of density estimators as the support function. The effect of training set size and choice of kernel will be also studied.

Acknowledgments

This work was supported partially by the Korea Research Foundation (KRF) Grant funded by the Korean Government (No. 2008-314-D00483) and partially by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0025530).

Appendix A. Proof of Theorem 1

Proof. Let Λ be a feasible connected component of the set $L_q(r)$ and $\mathbf{x}\in\Lambda$. Since $q(\mathbf{x})<r$, we have $q_r(\mathbf{x})=0$ and so $\mathbf{x}$ becomes an equilibrium point of system (7). Without loss of generality (WLOG), we may assume that the set $S=\{\mathbf{x}\in\Re^n : r\le q(\mathbf{x})\le r'\}$ for some $r'>r$ does not contain any equilibrium points of system (7) by Morse–Sard's theorem. Then we can choose a $\delta>0$ such that $\Lambda\subset N_\delta(\Lambda)\subset L_q(r')=\{\mathbf{x}\in\Re^n : q(\mathbf{x})<r'\}$. Now suppose that a point $\mathbf{x}_0\in N_\delta(\Lambda)\backslash\Lambda$, i.e. $q_r(\mathbf{x}_0)\simeq 1_{\{q(\mathbf{x})\ge r\}}>0$, WLOG. If $\mathbf{x}(\cdot):\Re\to\Re^n$ is the trajectory of system (7) starting from $\mathbf{x}(0)=\mathbf{x}_0$, then

$$\frac{dq(\mathbf{x}(t))}{dt}=\nabla q(\mathbf{x})^T\frac{d\mathbf{x}(t)}{dt}=-q_r(\mathbf{x}(t))\|\nabla q(\mathbf{x}(t))\|^2<0$$

which implies that the trajectory $\mathbf{x}(t)$ starting from $\mathbf{x}_0\in S$ points into the region that decreases the function value, q. Also we note that on the region $S=\{\mathbf{x}\in\Re^n : r\le q(\mathbf{x})\le r'\}$, $F_q(\mathbf{x})=-\nabla q(\mathbf{x})$ since $q_r(\mathbf{x})\simeq 1_{\{q(\mathbf{x})\ge r\}}$, so it can be shown that system (7) is completely stable, i.e. every trajectory converges to one of the equilibrium points by the complete stability property of $\dot{\mathbf{x}}=-\nabla q(\mathbf{x})$ in [7,8]. Therefore, the fact that S does not contain equilibrium points implies that every trajectory $\mathbf{x}(t)$ starting from $\mathbf{x}_0\in N_\delta(\Lambda)$ should converge to Λ. This proves that Λ is an attracting manifold of system (7). □

Appendix B. Proof of Theorem 2

Proof. From the complete stability property of system $\dot{\mathbf{x}}=-\nabla q(\mathbf{x})$ in [8], we know that

$$\Re^n=\bigcup_{i=1}^{l}B(\mathbf{s}_i) \tag{10}$$

where $\{\mathbf{s}_i : i=1,\ldots,l\}$ is the set of all the (isolated) stable equilibrium vectors and $B(\mathbf{s}_i)$ is the basin of attractions for $\mathbf{s}_i$. Since all the stable equilibrium vectors $\mathbf{s}_i$ are inside the region $L_q(r)$ and $F_q(\mathbf{x})=-\nabla q(\mathbf{x})$ for all $\mathbf{x}\in\Re^n\backslash L_q(r)$, the result follows. □

References

[1] D.M.J. Tax, R.P.W. Duin, Support vector domain description, Pattern Recognition Letters 20 (1999) 1191–1199.
[2] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, Neural Computation 13 (2001) 1443–1471.
[3] C. Lai, D.M.J. Tax, R.P.W. Duin, E. Pekalska, P. Paclik, On combining one-class classifiers for image database retrieval, in: Lecture Notes in Computer Science, vol. 2364, 2002, pp. 212–221.
[4] T. Onoda, H. Murata, S. Yamada, Non-relevance feedback document retrieval based on one-class SVM and SVDD, in: Proceeding of 2006 IEEE World Congress on Computational Intelligence, 2006, pp. 2191–2198.
[5] J. Yang, V. Estivill-Castro, S.K. Chalup, Support vector clustering through proximity graph modelling, in: Proceedings of the Ninth International Conference on Neural Information Processing (ICONIP'02), 2002, pp. 898–903.
[6] A. Ben-Hur, D. Horn, H.T. Siegelmann, V. Vapnik, Support vector clustering, Journal of Machine Learning Research 2 (2001) 125–137.
[7] J. Lee, D. Lee, An improved cluster labeling method for support vector clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2005) 461–464.
[8] J. Lee, D. Lee, Dynamic characterization of cluster structures for robust and inductive support vector clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (2006) 1869–1874.
[9] K.H. Jung, D. Lee, J. Lee, Fast support-based clustering method for large-scale problems, Pattern Recognition 43 (2010) 1975–1983.
[10] D. Lee, J. Lee, Domain described support vector classifier for multi-class classification problems, Pattern Recognition 40 (2007) 41–51.
[11] D. Lee, J. Lee, Equilibrium-based support vector machine for semi-supervised classification, IEEE Transactions on Neural Networks 18 (2) (2007) 578–583.
[12] D. Lee, K.H. Jung, J. Lee, Constructing sparse kernel machines using attractors, IEEE Transactions on Neural Networks 20 (4) (2009) 721–729.
[13] D. Lee, J. Lee, Dynamic dissimilarity measure for support-based clustering, IEEE Transactions on Knowledge and Data Engineering 22 (6) (2010) 900–905.
[14] H.-C. Kim, J. Lee, Clustering based on gaussian processes, Neural Computation 19 (11) (2007) 3088–3107.
[15] D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 603–619.
[16] F. Camastra, A. Verri, A novel kernel method for clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (3) (2005) 461–464.
[17] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.
[18] J.J. Hull, A database for handwritten text recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1994) 550–554.

[19] S.A. Nene, S.K. Nayar, H. Murase, Columbia Object Image Library (COIL-20), Technical Report CUCS-005-96, 1996.
[20] J. Lee, H.D. Chiang, Theory of stability regions of a class of non-hyperbolic dynamical systems and its applications to constraint satisfaction problems, IEEE Transactions on Circuits and Systems—Part I: Fundamental Theory and Applications 49 (2) (2002) 196–209.
[21] M.C. Motwani, M.C. Gadiya, R.C. Motwani, Survey of image denoising techniques, in: Proceedings of GSPx, Santa Clara, CA, 2004.
[22] A. Buades, B. Coll, J.M. Morel, On image denoising methods, SIAM Multiscale Modeling and Simulation 4 (2005) 490–530.
[23] J. Park, D. Kang, J. Kim, J.T. Kwok, I.W. Tsang, SVDD-based pattern denoising, Neural Computation 19 (7) (2007) 1919–1938.
[24] S. Mika, B. Schölkopf, A. Smola, K.R. Müller, M. Scholz, G. Rätsch, Kernel PCA and de-noising in feature spaces, Advances in Neural Information Processing Systems 11 (1999) 536–542.
[25] J.T. Kwok, I.W. Tsang, The pre-image problem in kernel methods, IEEE Transactions on Neural Networks 15 (2004) 1517–1525.
[26] D.L. Donoho, De-noising by soft-thresholding, IEEE Transactions on Information Theory 41 (3) (1995) 613–627.
[27] F. Camastra, M. Spinetti, A. Vinciarelli, Offline cursive character challenge: a new benchmark for machine learning and pattern recognition algorithms, in: Proceedings of 18th International Conference on Pattern Recognition (ICPR06), 2006, pp. 913–916.
[28] F. Camastra, A SVM-based cursive character recognizer, Pattern Recognition 40 (12) (2007) 3721–3727.

Kyu-Hwan Jung received B.S. in industrial engineering from Pohang University of Science and Technology (POSTECH) in 2005 and is expected to receive a Ph.D. candidate in the Department of Industrial and Management Engineering at POSTECH in 2010. He is interested in pattern recognition, support vector machine, and their applications to data mining, pattern denoising, and image segmentation. E-mail: onlyou7@postech.ac.kr

Namhyoung Kim received B.S. in industrial engineering from Pohang University of Science and Technology (POSTECH) in 2008 and is currently pursuing a Ph.D. in the Department of Industrial and Management Engineering at POSTECH. She is currently interested in pattern recognition, pattern denoising, and data mining. E-mail: skagud@postech.ac.kr

Jaewook Lee is an associate professor in the Department of Industrial and Management Engineering at Pohang University of Science and Technology (POSTECH), Pohang, Korea. He received the B.S. degree from Seoul National University, and the Ph.D. degree from Cornell University in 1993 and 1999, respectively. His research interests include pattern recognition, neural networks, global optimization, nonlinear systems, and their applications to data mining and financial engineering. E-mail: jaewookl@postech.ac.kr