Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 847062, 10 pages
http://dx.doi.org/10.1155/2013/847062

Research Article

Optimizing Kernel PCA Using Sparse Representation-Based Classifier for MSTAR SAR Image Target Recognition

Chuang Lin,[1] Binghui Wang,[1] Xuefeng Zhao,[2] and Meng Pang[1]

[1] *School of Software, Dalian University of Technology, Dalian 116620, China*
[2] *School of Civil Engineering, Dalian University of Technology, Dalian 116024, China*

Correspondence should be addressed to Xuefeng Zhao; zhaoxf@dlut.edu.cn

Received 10 January 2013; Revised 1 April 2013; Accepted 7 April 2013

Academic Editor: Yingwei Zhang

Different kernels cause various class discriminations owing to their different geometrical structures of the data in the feature space. In this paper, a method of kernel optimization by maximizing a measure of class separability in the empirical feature space with sparse representation-based classifier (SRC) is proposed to solve the problem of automatically choosing kernel functions and their parameters in kernel learning. The proposed method first adopts a so-called data-dependent kernel to generate an efficient kernel optimization algorithm. Then, a constrained optimization function using general gradient descent method is created to find combination coefficients varied with the input data. After that, optimized kernel PCA (KOPCA) is obtained via combination coefficients to extract features. Finally, the sparse representation-based classifier is used to perform pattern classification task. Experimental results on MSTAR SAR images show the effectiveness of the proposed method.

1. Introduction

Recently, kernel learning or kernel machine has aroused broad interest in pattern recognition and kernel learning areas. For classification problem based supervised kernel learning, different kernel geometrical structures give different class discriminations. However, separability of the data in the feature space could be even worse if an inappropriate kernel is chosen since the geometrical structure of the mapped data in the feature space is totally determined by the kernel matrix, so the selection of kernel influences greatly the performance of kernel learning and thus optimizing kernel can be regarded as an effective way to improve classification performance. Considering that optimized kernel parameters of kernel function cannot change the geometrical structures of kernel in the feature space [1, 2], so it cannot improve the performance of kernel learning. In this sense, Scholkopf et al. [3] proposed an empirical kernel map which maps original input data space into a subspace of the empirical feature space. Since the training data have the same geometrical structure in both the empirical feature space and the feature space, and the former is easier to access than the latter, it is easier to study the adaptability of a kernel to the input data and to improve it in the former space. Cristianini et al. [4] and Lanckrict et al. [5] have proposed methods of choosing kernel by optimizing the measure of data separation in the feature space for the first time. Cristianini et al. and Lanckrict et al., respectively, employ the alignment and margin as the measure of data separation to evaluate the adaptability of a kernel to input data. Zhang et al. proposed several variants of KPCA [6, 7] to perform fault diagnosis and nonlinear processes. Then, they utilized the improved kernel learning techniques to deal with statistical analysis of nonlinear fault detection [8], large-scale fault diagnosis processes [9], and the monitoring of dynamic processes [10].

Simultaneously, sparse representation has gained great interest in pattern recognition and computer vision areas recently. Wright et al. [11] presented a sparse representation based classification method [12] and applied it to real-world face recognition problems [11, 12]. With varying expression and illumination, as well as occlusion and disguise, it was very effective and robust for face recognition.

The paper is organized as follows: in Section 2, we first introduce the concept of data-dependent kernel and

empirical feature space. Then, we optimize the kernel in the empirical feature space by seeking the optimal combination coefficients of data-dependent kernel based on Fisher criterion. In Sections 3 and 4, the optimized kernel PCA (KOPCA) is adopted for MSATR SAR images to obtain dimensionality reduced empirical feature so as to employ sparse representation-based classifier for pattern classification. Finally, in Section 5, experiments are carried out on MSTAR SAR images to demonstrate the improvement in the performance of the data classification algorithms after using the optimized kernel and sparse representation-based classifier.

2. Kernel Optimization in the Empirical Feature Space

2.1. Data-Dependent Kernel. Since different kernels create different geometrical structures of the data in the feature space and lead to different class discriminations [13], there does not exist a kernel function that can be adaptive to all datasets in kernel learning. Therefore, data-dependent based kernel is necessary to be chosen to deal with the problem. In this paper, we employ a data-dependent kernel which is proposed by Amari and Wu [14] to be the objective kernel function to conduct kernel optimization. There is a need to explain that the data-dependent kernel is a conformal transformation to a basic kernel.

Given a set of N training samples $x_1, \ldots, x_N \in \mathbf{R}^d$, the data-dependent kernel is defined as follows:

$$k(x, y) = q(x)\, q(y)\, k_0(x, y), \tag{1}$$

where $x, y \in \mathbf{R}^d$, $k_0(x, y)$ is a basic kernel such as a polynomial kernel or Gaussian kernel. $q(\cdot)$ is a positive real valued factor function and different $q(\cdot)$ make the data-dependent kernel different properties; Amari and Wu [14] expand the spatial resolution in the margin of a SVM by

$$q(x) = \sum_{i \in \mathrm{SV}} \alpha_i k_1(x, a_i), \tag{2}$$

where $k_1(x, a_i) = \exp(-\gamma \|x - a_i\|)$, $a_i \in \mathbf{R}^d$ is the ith support vector, and SV is a set of support vector. The set $\{\mathbf{a} \mid \mathbf{a} = [a_1, a_2, \ldots, a_n]\}$ called the "empirical cores," can be determined according to the distribution of the training data. γ is a free parameter. $\alpha_i \{i = 0, 1, \ldots, n\}$ are the positive combination coefficients which are considered as contribution weights corresponding to the a_i. Meanwhile, the data-dependent kernel is a kernel function as it satisfies the Mercer condition [3].

Let K and K_0 be the Kernel Matrices of $k(x, y)$ and $k_0(x, y)$, respectively. Thus, it is easy to see that

$$K = \left[q(x_i)\, q(x_j)\, k_0(x_i, x_j)\right]_{N \times N} = QK_0Q, \tag{3}$$

where $K = [k(x_i, x_j)]_{N \times N}$, $K_0 = [k_0(x_i, x_j)]_{N \times N}$ and Q is a diagonal matrix with elements $\{q(x_1), q(x_2), \ldots, q(x_N)\}$.

We denote vectors $[q(x_1), q(x_2), \ldots, q(x_N)]^T$ and $[\alpha_0, \alpha_1, \ldots, \alpha_n]^T$ as $\mathbf{q}$ and $\boldsymbol{\alpha}$, respectively. Then, we have

$$\mathbf{q} = \begin{bmatrix} 1 & k_1(x_1, a_1) & \cdots & k_1(x_1, a_n) \\ 1 & k_1(x_2, a_1) & \cdots & k_1(x_2, a_n) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & k_1(x_N, a_1) & \cdots & k_1(x_N, a_n) \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = K_1 \boldsymbol{\alpha}. \tag{4}$$

2.2. Empirical Feature Space. The different kernels cause various class discriminations owing to their different geometrical structures of the data in the feature space. It is often not so convenient or easy to compute in feature space. Hence, the concept of empirical feature space is introduced.

Let $x_i \{i = 1, 2, \ldots, N\}$ be a d-dim training dataset and $X^T = [x_1, x_2, \ldots, x_N]$. $K = [k(x_i, x_j)]_{N \times N}$ denotes the kernel matrix with rank r. Since K is a symmetric positive semidefinite matrix, it can be decomposed as

$$K_{N \times N} = P_{N \times r} \Lambda_{r \times r} P^T_{r \times N}, \tag{5}$$

where Λ is a diagonal matrix with r positive eigenvalues of K in descending order, and P contains r eigenvectors corresponding to the positive eigenvalues.

On the basis of above, we can define the map from input data space to $\mathbf{R}^r$ Euclidean space and gain the so-called empirical kernel map Φ_r defined in [3], that is,

$$\Phi_r : X \to \mathbf{R}^r,$$
$$x \longmapsto \Lambda^{-1/2} P^T \left(k(x, x_1), k(x, x_2), \ldots, k(x, x_N)\right)^T. \tag{6}$$

The embedding space $\Phi_r(X) \subset \mathbf{R}^r$ is called empirical feature space.

We can prove that the training data has the same geometric structure in both the empirical feature space and feature space. Let $Y = KP\Lambda^{-1/2}$, then the dot product matrix $\{\Phi_r(x_i)\}$ in the empirical feature space can be calculated as

$$YY^T = KP\Lambda^{-1/2}\Lambda^{-1/2}P^TK^T = K. \tag{7}$$

Notice that $K^T = K$, $K = [\Phi(x_i) \cdot \Phi(x_j)]_{N \times N}$, and the result of YY^T is exactly the dot product matrix of $\{\Phi(x_i)\}$ in the feature space; therefore, we say that the empirical feature space preserves the geometric structure in the feature space.

2.3. Fisher Criterion Based Kernel Optimization. As illustrated in Section 2.2, the training data has the same geometric structure in both the empirical feature space and feature space and it is easier to access the empirical feature space than the feature space, so it is better to measure class separability in the empirical feature space. In this paper, we choose the acquainted Fisher criteria to measure class separability:

$$J = \frac{\operatorname{tr} S_b}{\operatorname{tr} S_w}, \tag{8}$$

where S_b is the between-class scatter matrix, S_w is the within-class scatter matrix, and tr is the trace of given matrix. J is Fisher criteria to measure the class separability. Notice that J measures the class separability in the feature space and J is independent of the projections in the common projection subspace, so it is a satisfying choice to be the task of kernel optimization.

Up to now, the kernel optimization problem is transformed to maximize Fisher scalar J. Let the number of each class i ($i = 1, 2, \ldots, C$) be N_i, that is, class i has N_i training samples and N denotes the number of all training samples. What is more, let m_i, m_0 denote the center of each training samples in class i ($i = 1, 2, \ldots, C$) and the center of all training samples, respectively, that is, $m_i = (1/N_i)\sum_{j=1}^{N_i} y_j$, $m_0 = (1/N)\sum_{k=1}^{N} y_k$ and y_i ($i = 1, 2, \ldots, N$) be the images of the training samples in the empirical feature space, that is, $y_i = \Phi_r(x_i)$. Then, we can define

$$\begin{aligned} \operatorname{tr} S_b &= \frac{1}{N}\sum_{i=1}^{C} N_i (m_i - m_0)^T (m_i - m_0), \\ \operatorname{tr} S_w &= \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N_i} \left(y_i^j - m_i\right)^T \left(y_i^j - m_i\right), \end{aligned} \tag{9}$$

where y_i^j means the jth training sample in the ith class.

For the convenience of calculation and representation, we rewrite the kernel matrix K as

$$K = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1C} \\ K_{21} & K_{22} & \cdots & K_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ K_{C1} & K_{C2} & \cdots & K_{CC} \end{bmatrix}, \tag{10}$$

where K_{ij} ($i = 1, 2, \ldots, C$; $j = 1, 2, \ldots, C$) represent the submatrices of K and the size of K_{ij} is $N_i \times N_j$.

Let the following matrices B and W be called "between-class" and "within-class" kernel scatter matrices, respectively,

$$\begin{aligned} B = &\begin{bmatrix} \frac{1}{N_1}K_{11} & 0 & \cdots & 0 \\ 0 & \frac{1}{N_2}K_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{N_C}K_{CC} \end{bmatrix} \\ &- \frac{1}{N}\begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1C} \\ K_{21} & K_{22} & \cdots & K_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ K_{C1} & K_{C2} & \cdots & K_{CC} \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} W = &\operatorname{diag}(k_{11}, k_{22}, \ldots, k_{NN}) \\ &- \begin{bmatrix} \frac{1}{N_1}K_{11} & 0 & \cdots & 0 \\ 0 & \frac{1}{N_2}K_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{N_C}K_{CC} \end{bmatrix}. \end{aligned} \tag{11}$$

We can also employ B_0 and W_0 to denote "between-class" and "within-class" kernel scatter matrices corresponding to the basic kernel K_0.

Now we establish the relation between Fisher scalar J and the proposed kernel scatter matrices. Let 1_N be the N-dim vector whose elements are equal to 1. Then we can get

$$J = \frac{\mathbf{q}^T B_0 \mathbf{q}}{\mathbf{q}^T W_0 \mathbf{q}}. \tag{12}$$

The proof is given in the appendix.

To maximize J, we adopt the general gradient method and use formula (4). Define

$$\begin{aligned} J_1 &= \mathbf{q}^T B_0 \mathbf{q} = (K_1\boldsymbol{\alpha})^T B_0 K_1 \boldsymbol{\alpha} = \boldsymbol{\alpha}^T K_1^T B_0 K_1 \boldsymbol{\alpha}, \\ J_2 &= \mathbf{q}^T W_0 \mathbf{q} = (K_1\boldsymbol{\alpha})^T W_0 K_1 \boldsymbol{\alpha} = \boldsymbol{\alpha}^T K_1^T W_0 K_1 \boldsymbol{\alpha}. \end{aligned} \tag{13}$$

Then,

$$\begin{aligned} \frac{\partial J_1}{\partial \boldsymbol{\alpha}} &= 2K_1^T B_0 K_1 \boldsymbol{\alpha}, \\ \frac{\partial J_2}{\partial \boldsymbol{\alpha}} &= 2K_1^T W_0 K_1 \boldsymbol{\alpha}. \end{aligned} \tag{14}$$

Thus,

$$\begin{aligned} \frac{\partial J}{\partial \boldsymbol{\alpha}} &= \frac{\partial (J_1/J_2)}{\partial \boldsymbol{\alpha}} = \frac{(\partial J_1/\partial \boldsymbol{\alpha})\, J_2 - J_1 (\partial J_2/\partial \boldsymbol{\alpha})}{J_2^2} \\ &= \frac{2}{J_2^2}\left(K_1^T B_0 K_1 J_2 - K_1^T W_0 K_1 J_1\right)\boldsymbol{\alpha} \\ &= \frac{2}{J_2}\left(K_1^T B_0 K_1 - K_1^T W_0 K_1 J\right)\boldsymbol{\alpha}. \end{aligned} \tag{15}$$

Denote $M_0 = K_1^T B_0 K_1$ and $N_0 = K_1^T W_0 K_1$, maximize J is equivalent to $\partial J/\partial \boldsymbol{\alpha} = 0$, that is,

$$M_0 \boldsymbol{\alpha} = N_0 J \boldsymbol{\alpha}. \tag{16}$$

Considering the fact that it is almost impossible to make W_0 invertible because of the limited amount of training samples in real-world applications, the general gradient descent method is adopted to get an approximate value of the optimal $\boldsymbol{\alpha}$. The updating equation to maximize J is defined as follows:

$$\boldsymbol{\alpha}^{(n+1)} = \boldsymbol{\alpha}^{(n)} + \eta\left(\frac{1}{J_2}M_0 - \frac{J}{J_2}N_0\right)\boldsymbol{\alpha}^{(n)}. \tag{17}$$

To guarantee the convergence of formula (17), η is defined as the function of iterations, that is,

$$\eta(q) = \eta_0\left(1 - \frac{q}{Q}\right), \tag{18}$$

where η_0 is a predefined initial value, Q denotes total number of iterations, and q represents the current iteration number.

After we calculate the combination coefficients of $\boldsymbol{\alpha}$, then we can get $\mathbf{q}$, as $\mathbf{q} = K_1\boldsymbol{\alpha}$, and thus, the optimizing kernel or data-dependent kernel, K, is easy to achieve.

3. Optimizing Kernel PCA (KOPCA)

In this section, we will employ the optimized kernel function mentioned above to construct the optimizing kernel PCA and extract feature in the empirical feature space.

Given a set of N training samples $x_1, x_2, \ldots, x_N \in \mathbf{R}^d$ and an empirical feature mapping Φ_r, let the input data space X be mapped into the empirical feature space $\mathbf{R}^r : \Phi_r : X \to \mathbf{R}^r$, $x \mapsto \Phi_r(x)$. The covariance operator on the empirical feature space $\mathbf{R}^r$ can be constructed by

$$S^{\Phi_r} = \frac{1}{N}\sum_{i=1}^{N}(y_i - m_0)(y_i - m_0)^T, \tag{19}$$

where y_i and m_0 are defined the same as above, that is, $y_i = \Phi_r(x_i)$, $m_0 = (1/N)\sum_{k=1}^{N} y_k$. It is easy to proof that all nonzero eigenvalues of S^{Φ_r} are positive, and every eigenvector β of S^{Φ_r} can be linearly expanded by

$$\beta = \sum_{i=1}^{N}\theta_i y_i. \tag{20}$$

To get these expansion coefficients, we denote $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_m]$, $Y^T = [y_1, y_2, \ldots, y_N]$ and form an $N \times N$ Gram matrix $G = YY^T$, whose elements are determined by optimizing kernel, that is, $G_{ij} = y_i^T y_j = (\Phi_r(x_i), \Phi_r(x_j)) = k(x_i, x_j)$. Note that the kernel matrix $K = [k(x_i, x_j)]_{N\times N}$ is the same as what is defined in formula (7).

Centralize G by

$$G_c = G - 1_N \times G - G \times 1_N + 1_N \times G \times 1_N, \tag{21}$$

where 1_N is defined the same as above, that is, 1_N is N-dim vector whose elements are equal to 1.

Let the eigenvectors of G_c be $\gamma_1, \gamma_2, \ldots, \gamma_m$ corresponding to the m largest positive eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$. Then, the m eigenvectors of S^{Φ_r}, $\beta_1, \beta_2, \ldots, \beta_m$, corresponding to the m largest positive eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$, are

$$\beta_j = \frac{1}{\sqrt{\lambda_j}}Y^T\gamma_j. \tag{22}$$

After the projection of a mapped sample $y_t = \Phi_r(x_t)$ onto the eigenvectors $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_m]$, we can obtain optimizing kernel PCA transformed feature vector $\mathbf{z}$ by

$$\mathbf{z} = \boldsymbol{\beta}^T y_t. \tag{23}$$

Meanwhile, the jth optimizing kernel PCA component:

$$\begin{aligned} z_j = \beta_j^T y_t &= \frac{1}{\sqrt{\lambda_j}}\gamma_j^T Y y_t = \frac{1}{\sqrt{\lambda_j}}\gamma_j^T \begin{bmatrix} y_1^T \\ y_2^T \\ \vdots \\ y_N^T \end{bmatrix} y_t \\ &= \frac{1}{\sqrt{\lambda_j}}\gamma_j^T [k(x_1, x_t), k(x_2, x_t), \ldots, k(x_N, x_t)]^T. \end{aligned} \tag{24}$$

Up to now, the essence of optimizing kernel PCA has been revealed. That is, we first maximize a measure of class separability in the empirical feature space by virtue of Fisher criterion to form needful data-dependent kernel and then take advantage of optimizing kernel PCA to extract feature in the empirical feature space.

4. Sparse Representation-Based Classifier (SRC)

Let A_i $(i = 1, 2, \ldots, C)$ be the matrix formed by the training samples of the ith class, that is, $A_i = [x_{i1}, x_{i2}, \ldots, x_{iN_i}] \in \mathbf{R}^{d\times N_i}$. And define a new matrix A for the total training set with C classes as the concatenation of the N training samples: $A = [A_1, A_2, \ldots, A_C] \in \mathbf{R}^{d\times N}$.

Given a test sample y from the ith class, then y can be approximately represented by the linear span of the training samples in the corresponding class, that is,

$$\begin{aligned} y &= \mu_{i1}x_{i1} + \mu_{i2}x_{i2} + \cdots + \mu_{iN_i}x_{iN_i} = \sum_{j=1}^{N_i}\mu_{ij}x_{ij} \\ &= A_i\boldsymbol{\mu}_i, \end{aligned} \tag{25}$$

where μ_{ij} $(j = 1, 2, \ldots, N_i)$ are the corresponding coefficients, and we denote $\boldsymbol{\mu}_i = [\mu_{i1}, \mu_{i2}, \ldots, \mu_{iN_i}]^T$.

Then, the linear representation of y can be rewritten in terms of A as

$$y = A\boldsymbol{\mu}, \tag{26}$$

where $\boldsymbol{\mu} = [0, \ldots, 0, \boldsymbol{\mu}_i^T, 0, \ldots, 0]^T \in \mathbf{R}^N$ is a coefficient vector whose entries are zero except those associated with the ith class.

Hereto, we should take the number of row and column of A into consideration. If the row number d is bigger than column number N, the system of equations $y = A\boldsymbol{\mu}$ is overdetermined and the correct $\boldsymbol{\mu}$ can usually be found as its unique solution. Nevertheless, this is not what we need since sparse representation involves an underdetermined system of linear equations $y = A\boldsymbol{\mu}$, where $d < N$ as it is motivated by the following fact: given a test sample y, the representation is naturally sparse if training sample size (column number) is large enough, and if the sparser the coefficient vector $\boldsymbol{\mu}$ is, the easier it will be to accurately reconstruct the identity of the test sample y [12].

Consequently, it means that the dimension of feature vector (row number) must be smaller than the training sample

size (column number). Considering, before we use sparse representation, we have obtained dimensionality reduced empirical feature in Section 2, and just in time, it can meet requirements.

The above discussion motivates us to seek the sparest solution by solving the following optimization problem:

$$\left(\ell^0\right) \quad \widehat{\boldsymbol{\mu}}_0 = \operatorname{argmin} \|\boldsymbol{\mu}\|_0, \quad \text{s.t. } y = A\boldsymbol{\mu}, \tag{27}$$

where $\|\cdot\|_0$ denotes the ℓ^0-norm, which counts the number of nonzero entries in a vector.

However, solving ℓ^0 optimization problem in formula (27) is NP hard and time-consuming. Recent research of spares representation and compressed sensing [15, 16] proves that if the solution $\widehat{\boldsymbol{\mu}}_0$ is sparse enough, the solution of the ℓ^0 optimization problem is equivalent to finding the solution of the ℓ^1 optimization problem:

$$\left(\ell^1\right) \quad \widehat{\boldsymbol{\mu}}_1 = \arg\min \|\boldsymbol{\mu}\|_1, \quad \text{s.t. } y = A\boldsymbol{\mu}. \tag{28}$$

This problem can be solved in polynomial time by standard linear programming algorithms [17].

After obtaining the sparest solution $\widehat{\boldsymbol{\mu}}_1$, we can form a sparse representation-based classifier (SRC) in the following way. For each class i ($i = 1, 2, \ldots, C$), let $\delta_i : \mathbf{R}^N \to \mathbf{R}^N$ be a function which selects the coefficients associated with the ith class, then $\delta_i(\boldsymbol{\mu})$ is a vector whose only nonzero entries are the entries in $\boldsymbol{\mu}$ that are associated with class i. Making use of the coefficients associated with the ith class, one can reconstruct the given test sample y as $\widehat{y}_i = A\delta_i(\widehat{\boldsymbol{\mu}}_1)$. $\widehat{y}_i$ is often called the prototype of class i with respect to the sample y. The residual between y and its prototype $\widehat{y}_i$ of class i is defined as follows:

$$r_i(y) = \|y - \widehat{y}_i\|_2 = \|y - A\delta_i(\widehat{\boldsymbol{\mu}}_1)\|_2. \tag{29}$$

Then the SRC decision rule is to minimize the residual, that is, if $r_k(y) = \min_i r_i(y)$, y is assigned to class k. It is necessary to explain that our implementation minimizes the ℓ^1-norm via the basis pursuit denoising (BPDN) algorithm for linear programming based on [17–19].

5. Experimental Results

In this section, experiments are designed to evaluate the performance of the proposed algorithm. The first experiment is adopted to show that the class separability is probably worse in the feature space than that in the input space in some cases and demonstrate that the proposed kernel optimization algorithm can enlarge class separability. The second experiment is carried out on MSTAR SAR images using KOPCA compared with conventional KPCA to extract features and use nearest neighbor (NN) classifier to implement pattern classification. Simultaneously, sparse representation-based classifier (SRC) is applied to verify its superiority and effectiveness to deal with pattern classification compared with other classifiers. In order to verify the sparsity via BPDN, we randomly choose a test sample and show its representation coefficients on the training set.

5.1. Kernel Optimization on Synthetic Gaussian Distributed Dataset. Before concentrating on optimizing the kernel in the empirical feature space, we use two simple datasets called Gaussian distribution data generated by computer to get intuition about the embedding of data in the feature space into the empirical feature space. More information about data embedding can be found in [20]. Figure 1(a) shows a 2-class 2-dim dataset containing 400 samples, whose coordinates are uncorrelated. Each class contains 200 samples and both are Gaussian distributions with parameters: $\mu_x = -2$, $\mu_y = 0$, $\sigma_x = 2$, $\sigma_y = 1$ and $\mu_x = 2$, $\mu_y = 0$, $\sigma_x = 1$, $\sigma_y = 2$, respectively. Seeing this figure, there is some overlap between the two classes. Figure 1(b) shows the projection of the data into the empirical feature space onto the first two significant dimensions corresponding to the first two largest eigenvalues of K, when the polynomial kernel function $k_0(x, y) = (x, y)^d$ with $d = 3$ is used. Figure 1(c) shows the corresponding projection when the Gaussian kernel function $k_0(x, y) = \exp(-(1/\sigma)\|x - y\|^2)$ with $\sigma = 1.0 \times 10^5$ is employed. Both the two basic kernels are mentioned in formula (1). It is seen from Figures 1(b) and 1(c) that the class separability is worse in the feature space than that in the input space when adopting both the polynomial kernel and Gaussian kernel. Therefore, it is important to conduct kernel optimization. We will carry out an experiment later to demonstrate that when applying the kernel optimization algorithm in Section 2.3, the measure for class separability is surely enlarged.

In this experiment, we set parameter γ of the function $k_1(\cdot)$ in formula (2) as $\gamma = 1e6$ for the given polynomial kernel $k_0(x, y) = (x, y)^3$ and the given Gaussian kernel $k_0(x, y) = \exp(-1.0 \times 10^{-5}\|x - y\|^2)$. One-third of the synthetic data are randomly chosen to form the "empirical core" set $\{a_i\}$. The initial learning rate η_0 and total iteration number Q are set 0.1 and 200, respectively, in both the polynomial kernel and Gaussian kernel. Figures 2(a) and 2(b) show the projection of the data into empirical feature space onto the first two significant dimensions corresponding to the first two largest eigenvalues of K, when the polynomial kernel and Gaussian kernel are used as mentioned above. It is seen from Figure 2 that the proposed kernel optimization algorithm substantially improves the class separability of the data in the empirical feature space and, hence, in the feature space.

5.2. KOPCA Criterion on MSTAR SAR Dataset. This experiment is conducted on MSTAR SAR image provided by Defense Advanced Research Project Agency and Air Force Research Laboratory (DARPA/AFRL). The data is the MSTAR public release subset in order to initiate Moving and Stationary Target Acquisition and Recognition project which has provided a unique opportunity to promote and assess progress in SAR ATR algorithm development.

Since the characteristic in SAR image changes greatly with different aspect angles, a great many of images within one target-class were collected, where the poses lie between 0 and 360 degree.

The vehicle in MSTAR SAR Dataset contains BMP2 (sn-c21, sn-9563, sn-9566) tracked Armored Personnel Carrier, BTR70 (sn-c71) wheeled Armored Personnel Carrier, and

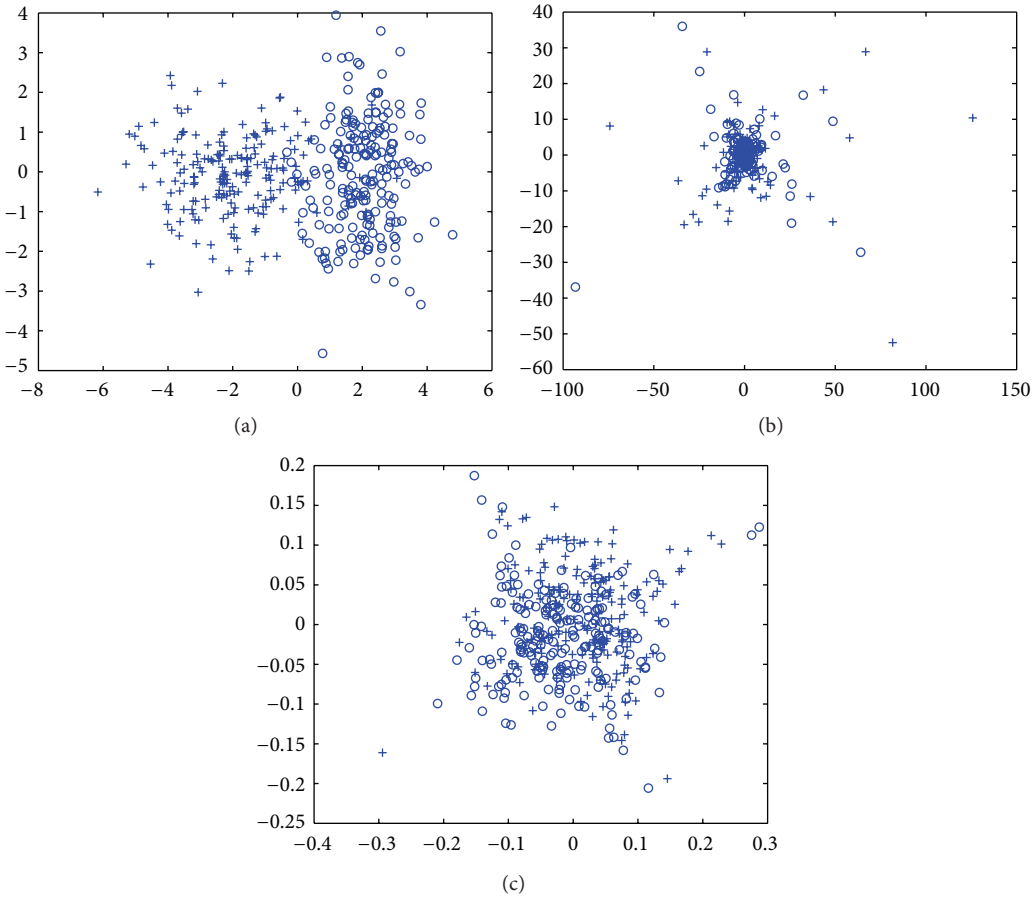

Figure 1: 2-Dim dataset and its projections in the feature space onto the first two significant dimensions. (a) Two classes of data samples with two Gaussian distributions. (b) 2-Dim projection in the feature space for polynomial kernel with $d = 3$. (c) 2-Dim projection in the empirical feature space for Gaussian kernel with $\sigma = 1.0 \times 10^5$.

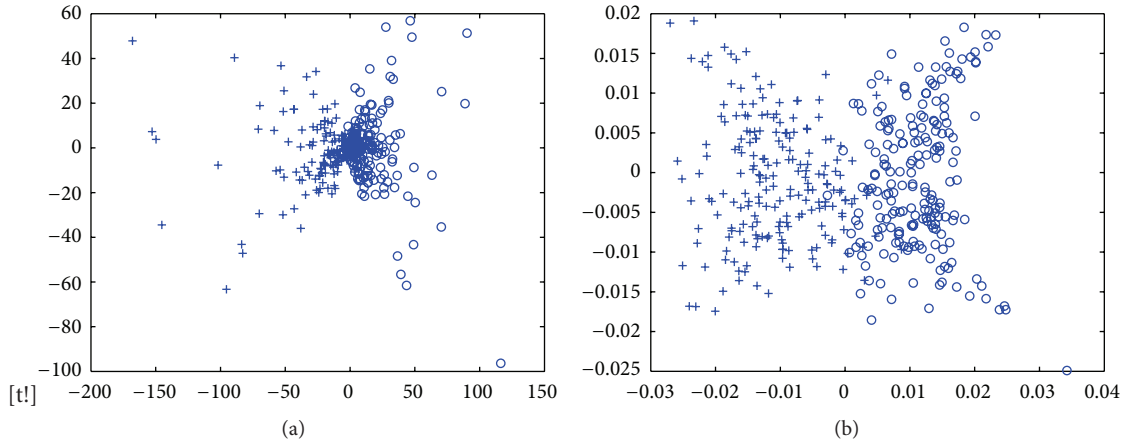

Figure 2: Improvement of class separability via kernel optimization algorithm. (a) 2-Dim projection in the empirical feature space for polynomial basic kernel $d = 3$. (b) 2-Dim projection in the empirical feature space for Gaussian basic kernel with $\sigma = 1.0 \times 10^5$.

T72 (sn-132, sn-812, sn-s7) Main Battle Tank. Different serial numbers in one-target class mean that vehicles are variant with small differences in configuration, articulation under extended operating condition (EOC) [21]. Therefore, scattering centers of SAR images change so intensively that recognition ability decreases greatly, and in this sense, recognizing variants in SAR images is difficult.

In this experiment, we select images of BMP2sn-c21, BTR70sn-c71 and T72sn-132 in 17 depression angle as the training samples (numbers of each class are 233, 233, 233). The

Table 1: Recognition rates of KPCA and KOPCA with polynomial kernel using NN classifier.

Polynomial kernel (d)	1	2	3	4	5	6	7	8	9	10
KPCA: NN	67.56	68.00	67.11	66.67	66.22	65.78	65.33	65.11	64.44	64.44
KOPCA: NN	77.78	77.56	77.56	77.11	77.11	76.67	76.00	75.56	75.33	74.67

Table 2: Recognition rates of KPCA and KOPCA with Gaussian kernel using NN classifier.

Gaussian kernel (σ)	$1e1$	$1e2$	$1e3$	$1e4$	$1e5$	$1e6$	$1e7$	$1e8$	$1e9$	$1e10$
KPCA: NN	62.00	62.00	62.00	66.11	68.00	67.11	62.00	61.33	34.22	31.11
KOPCA: NN	75.11	75.33	75.33	77.56	80.00	77.78	76.67	75.56	45.33	44.89

testing samples are selected as BMP2 sn-9563, BMP2 sn-9566, and T72 sn-812, T72sn-s7 in 15 depression angle (numbers of each class are 195, 196, 195, 191). The testing targets have small configuration differences to the training targets. There is a need to explanation; in this paper, KOPCA extracts features of all MSTAR images with different aspect angles directly and the process does not need to form different aspect windows and before recognition, images are chipped into $48 * 48$ pixels.

We set parameter γ of the function $k_1(\cdot)$ in formula (2) as $\gamma = 1e6$. The kernel functions are chosen as the polynomial kernel $k_0(x, y) = (x, y)^d$ where d is from 1 to 10, and the Gaussian kernel $k_0(x, y) = \exp(-\|x - y\|^2/\sigma)$ where σ is from $1e1$ to $1e10$. One-third of the training data are chosen to form the "empirical core" set $\{a_i\}$. The initial learning rate η_0 and total iteration number Q are set 0.1 and 200, respectively, in both the polynomial kernel and Gaussian kernel. Moreover, the feature dimension and empirical feature dimension are set 100 in both KPCA and KOPCA criterion. In order to reflect the performance of optimizing kernel in real-world application, the simplest nearest neighbor (NN) classifier is selected.

Suppose the distance between two samples x_i and x_j is defined by $d(x_i, x_j) = \|x_i - x_j\|_1$, where $\|\cdot\|_1$ denotes l1-norm.

Then, if a test sample x satisfies $d(x, x_j) = \min_i d(x, x_i)$, and x_i belongs to class k, then x belongs to class k.

Tables 1 and 2 show the recognition rates of KPCA and KOPCA with polynomial kernel and Gaussian kernel using the nearest neighbor (NN) classifier, respectively. From them, we can see that using the proposed data-dependent kernel optimization algorithm with KPCA criterion, recognition rate can be increased 10% ~15% in both polynomial kernel and Gaussian kernel compared with conventional KPCA in the whole process. The class separability in the empirical feature space is improved, and thus, recognition rate is improved.

Now, we will conduct another experiment to discuss sparse representation-based classifier (SRC). In the first part, we validate its effectiveness to deal with pattern classification task after extracting features via KOPCA (KOPCA: SRC) when compared with other classifiers, such as K-nearest neighbor (KNN) classifier, support vector classifier (SVC), and linear regression classifier (LRC) [22]. In the second part, in order to verify the sparsity of sparse representation-based classifier (SRC) via BPDN, we randomly choose a testing sample and show its representation coefficients on the training set.

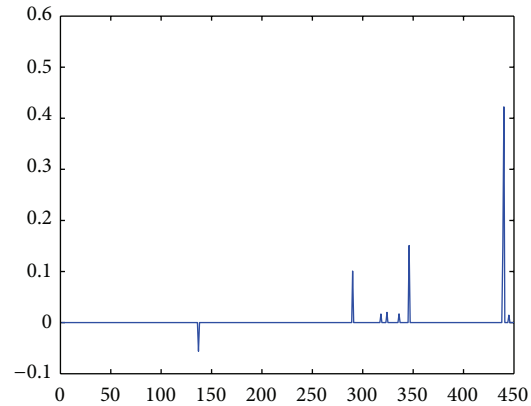

Figure 3: Representation coefficients to a chosen testing sample in the third class.

Tables 3 and 4, respectively, show the recognition rates of KOPCA with KNN, SVC, LRC, and SRC classifiers corresponding to polynomial kernel and Gaussian kernel.

From Table 3, we see that sparse representation-based classifier (SRC) outperforms other classifiers no matter what the order of polynomial kernel is. In the whole experiment process, the recognition rate of KOPCA: SRC achieves higher than 95% while others are lower than 95%. Only KOPCA: LRC is close to KOPCA: SRC, the others are lower than 10%, even 20% compared with it. Meanwhile, notice that there exists an interesting phenomenon, that is, though the order of polynomial kernel varies from 1 to 10, variations of recognition rates of all the algorithms are less than 10%. However, it is inapplicable to Gaussian kernel. From Table 4, we learn that KOPCA: SRC has the superiority and effectiveness, that is, (1) when parameter σ of Gaussian kernel is between $1e1$ and $1e5$, recognition rate of KOPCA: SRC is slightly lower than KOPCA: LRC, the difference is no more than 2%, but it is better than KOPCA: KNN and KOPCA: SVM. (2) Though, KOPCA: LRC performs well, but it has limitations when σ is equal or greater than $1e6$ since the recognition rate degrades quickly and significantly, while KOPCA: SRC remains stable and superior. (3) The performance of all these algorithms decreases rapidly when σ is greater than $1e8$ while KOPCA:

Table 3: Recognition rates of KOPCA with polynomial kernel using different classifier.

Polynomial kernel (d)	1	2	3	4	5	6	7	8	9	10
KOPCA: KNN	85.11	85.33	87.78	83.56	83.33	84.22	82.89	80.89	82.00	81.33
KOPCA: SVC	74.00	91.33	88.44	88.44	88.00	86.22	86.00	85.56	83.33	82.67
KOPCA: LRC	88.67	92.44	93.78	94.22	94.44	93.56	93.56	92.89	92.22	91.56
KOPCA: SRC	98.44	98.44	98.22	98.22	98.00	97.56	98.22	98.00	97.56	97.78

Table 4: Recognition rates of KOPCA with Gaussian kernel using different classifier.

Gaussian kernel (σ)	1e1	1e2	1e3	1e4	1e5	1e6	1e7	1e8	1e9	1e10
KOPCA: KNN	85.33	86.67	86.67	86.44	84.44	85.11	85.56	85.33	50.00	43.11
KOPCA: SVC	89.33	89.78	89.78	89.78	89.78	89.33	89.78	58.44	33.33	33.33
KOPCA: LRC	96.89	96.89	96.68	96.68	96.89	55.78	55.78	55.78	55.33	35.78
KOPCA: SRC	94.89	94.89	94.89	94.89	95.33	95.33	95.56	95.33	68.00	55.78

SRC can still hold about 70%, the others are equal or lower than 50%.

The basis pursuit (RB) method is introduced to optimize l1-norm based minimization problem in our experiment. Here, we randomly choose a testing sample in the third class. Intuitively, most nonzero representation coefficients for the testing sample lie in the range from 301 to 450 (since each class has 150 training samples, so the index for the third class is from 301 to 450). Fortunately, our result demonstrates it. From Figure 3, we see that the representation coefficients are sparse with respect to the basis, that is, the training set. Moreover, the nonzero coefficients are mostly located in the range from 301 to 450. As the end, we are to say that BPDN algorithm is fast enough to perform our SAR images' recognition and classification. The BPDN software package that we use is from the "L1 Homotopy" homepage: http://users.ece.gatech.edu/~sasif/homotopy/. The running time is in second level, which is as the same level as LRC and SVC.

Through the complete discussion above, we can come to the conclusion that optimizing kernel PCA can indeed enhance the class separability in the empirical feature space compared to conventional KPCA and thus improve recognition rate. In the meantime, sparse representation-based classifier is robust and of high efficiency for classification compared to other nice classifiers.

6. Conclusion

In this paper, we proposed an efficient pattern classification method named kernel optimized PCA with sparse representation classifier (KOPCA: SRC). After conducting several experiments, we can come to the following conclusions with the experimental results.

(1) We have proposed a new space called the empirical feature space, in which the data is embedded in a way that the geometrical structure of the data in the feature space is preserved.

(2) We have presented a general form of data-dependent kernel and derived an effective algorithm for optimizing kernel by maximizing class separability of the dataset in the empirical feature space via Fisher criterion.

(3) We have for the first time applied sparse representation-based classifier for pattern classification on MSTAR SAR image and experiment results reveal that it is more effective and robust than existing classifiers.

Appendix

Proof. Note the empirical feature mapping $\Phi_r : X \to \mathbf{R}^r$, and $y_i = \Phi_r(x_i)$, we know the dot product matrix K has exactly r positive eigenvalues.

Let $Y^T = [y_1, y_2, \ldots, y_N]$, $Y_i^T = [y_1^i, y_2^i, \ldots, y_{N_i}^i]$, $i = 1, \ldots, C$. Then, we have

$$m_i = \frac{1}{N_i}\sum_{j=1}^{N_i} y_j = \frac{1}{N_i} Y_i{}^T 1_{N_i},$$
$$m_0 = \frac{1}{N}\sum_{k=1}^{N} y_k = \frac{1}{N} Y^T 1_N. \tag{A.1}$$

As the empirical feature space preserves the dot product, that is,

$$\begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_C \end{bmatrix} \begin{bmatrix} Y_1^T & Y_2^T & \cdots & Y_C^T \end{bmatrix}$$

$$= \begin{bmatrix} Y_1Y_1^T & Y_1Y_2^T & \cdots & Y_1Y_C^T \\ Y_2Y_1^T & Y_2Y_2^T & \cdots & Y_2Y_C^T \\ \vdots & \vdots & \ddots & \vdots \\ Y_CY_1^T & Y_CY_2^T & \cdots & Y_CY_C^T \end{bmatrix}$$

$$= YY^T = K = \begin{bmatrix} K_{11} & K_{12} & \cdots & K_{1C} \\ K_{21} & K_{22} & \cdots & K_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ K_{C1} & K_{C2} & \cdots & K_{CC} \end{bmatrix}, \tag{A.2}$$

therefore

$$\begin{aligned} \operatorname{tr} S_b &= \frac{1}{N}\sum_{i=1}^{C} N_i (m_i - m_0)^T (m_i - m_0) \\ &= \frac{1}{N}\sum_{i=1}^{C} N_i {m_i}^T m_i - {m_0}^T m_0 \\ &= \frac{1}{N}\sum_{i=1}^{C} N_i \frac{1}{N_i} 1_{N_i}^T Y_i \frac{1}{N_i} Y_i^T 1_{N_i} - \frac{1}{N} 1_N^T Y \frac{1}{N} Y^T 1_N \\ &= \frac{1}{N}\sum_{i=1}^{C} \frac{1}{N_i} 1_{N_i}^T Y_i Y_i^T 1_{N_i} - \frac{1}{N^2} 1_N^T Y Y^T 1_N \\ &= \frac{1}{N}\sum_{i=1}^{C} \frac{1}{N_i} 1_{N_i}^T K_{ii} 1_{N_i} - \frac{1}{N^2} 1_N^T K 1_N \\ &= \frac{1}{N} \left[1_{N_1}^T 1_{N_2}^T \cdots 1_{N_C}^T \right] \\ &\quad \times \begin{bmatrix} \frac{1}{N_1}K_{11} & 0 & \cdots & 0 \\ 0 & \frac{1}{N_2}K_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{N_C}K_{CC} \end{bmatrix} \begin{bmatrix} 1_{N_1} \\ 1_{N_2} \\ \vdots \\ 1_{N_C} \end{bmatrix} \\ &\quad - \frac{1}{N} 1_N^T \begin{bmatrix} \frac{1}{N}K_{11} & \frac{1}{N}K_{12} & \cdots & \frac{1}{N}K_{1C} \\ \frac{1}{N}K_{21} & \frac{1}{N}K_{22} & \cdots & \frac{1}{N}K_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{N}K_{C1} & \frac{1}{N}K_{C2} & \cdots & \frac{1}{N}K_{CC} \end{bmatrix} 1_N \\ &= \frac{1}{N} 1_N^T B 1_N, \end{aligned} \tag{A.3}$$

$$\begin{aligned} \operatorname{tr} S_w &= \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{N_i} \left(y_i^j - m_i\right)^T \left(y_i^j - m_i\right) \\ &= \frac{1}{N}\sum_{i=1}^{C} \left(\sum_{j=1}^{N_i} \left(y_i^j\right)^T y_i^j - N_i {m_i}^T m_i \right) \\ &= \frac{1}{N} \left(\sum_{k=1}^{N} y_k^T y_k - \sum_{i=1}^{C} N_i {m_i}^T m_i \right) \\ &= \frac{1}{N}\sum_{k=1}^{N} y_k^T y_k - \frac{1}{N}\sum_{i=1}^{C} \frac{1}{N_i} 1_{N_i}^T K_{ii} 1_{N_i} \\ &= \frac{1}{N} 1_N^T \left\{ \operatorname{diag}\left(k_{11}, k_{22}, \ldots, k_{NN}\right) \right. \\ &\quad \left. - \begin{bmatrix} \frac{1}{N_1}K_{11} & 0 & \cdots & 0 \\ 0 & \frac{1}{N_2}K_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{1}{N_C}K_{CC} \end{bmatrix} \right\} 1_N \\ &= \frac{1}{N} 1_N^T W 1_N. \end{aligned} \tag{A.4}$$

Note formula (3), we easily get $B = QB_0Q$, $W = QW_0Q$, simultaneously, $1_N^T Q = \mathbf{q}^T$ and $Q1_N = \mathbf{q}$. Hence,

$$\begin{aligned} J &= \frac{(1/N)\, 1_N^T B 1_N}{(1/N)\, 1_N^T W 1_N} = \frac{1_N^T B 1_N}{1_N^T W 1_N} \\ &= \frac{1_N^T Q B_0 Q 1_N}{1_N^T Q W_0 Q 1_N} = \frac{\mathbf{q}^T B_0 \mathbf{q}}{\mathbf{q}^T W_0 \mathbf{q}}. \end{aligned} \tag{A.5}$$

□

Acknowledgments

The work is supported by Major Program of Natural Science Foundation of China, no. 61033012, Natural Science Foundation of China, no. 611003177, no. 61272371, Fundamental Research Funds for the Central Universities, no. DUT12JR07, and Specialized Research Fund for the Doctoral Program of Higher Education, no. 20120041120046.

References

[1] J. Huang, P. C. Yuen, W. S. Chen, and J. H. Lai, "Kernel subspace LDA with optimized kernel parameters on face recognition," in *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition (FGR '04)*, pp. 327–332, May 2004.

[2] L. Wang, K. L. Chan, and P. Xue, "A criterion for optimizing kernel parameters in KBDA for image retrieval," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 35, no. 3, pp. 556–562, 2005.

[3] B. Scholkopf, S. Mika, C. J. C. Burges et al., "Input space versus feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 1000–1017, 1999.

[4] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor, "On kernel target alignment," in *Proceedings of the Neural Information Processing Systems (NIPS '01)*, pp. 367–373, 2002.

[5] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[6] Y. W. Zhang, "Enhanced statistical analysis of nonlinear processes using KPCA, KICA and SVM," *Chemical Engineering Science*, vol. 64, no. 5, pp. 801–811, 2009.

[7] Y. W. Zhang and C. Ma, "Fault diagnosis of nonlinear processes using multiscale KPCA and multiscale KPLS," *Chemical Engineering Science*, vol. 66, no. 1, pp. 64–72, 2011.

[8] Y. W. Zhang and S. Joe Qin, "Improved nonlinear fault detection technique and statistical analysis," *AIChE Journal*, vol. 54, no. 12, pp. 3207–3220, 2008.

[9] Y. Zhang, H. Zhou, S. J. Qin, and T. Chai, "Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 1, pp. 3–10, 2010.

[10] Y. Zhang, "Modeling and monitoring of dynamic processes," *IEEE Transactions on Neural Networks and Learning System*, vol. 23, no. 2, pp. 277–284, 2012.

[11] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. SHuang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proceedings of IEEE*, vol. 98, no. 6, pp. 1031–1044, 2009.

[12] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[13] H. L. Xiong, M. N. S. Swamy, and M. O. Ahmad, "Optimizing the kernel in the empirical feature space," *IEEE Transactions on Neural Networks*, vol. 16, no. 2, pp. 460–474, 2005.

[14] S. Amari and S. Wu, "Improving support vector machine classifiers by modifying kernel functions," *Neural Network*, vol. 12, no. 6, pp. 783–789, 1999.

[15] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal l_1-norm solution is also the sparsest solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[16] E. J. Candès, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[17] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, 2001.

[18] D. L. Donoho and Y. Tsaig, "Fast solution of ℓ_1-norm minimization problems when the solution may be sparse," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4789–4812, 2008.

[19] D. Malioutov and M. Cetin, "Homotopy continuation for sparse signal representation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, 2005.

[20] E. Pekalska, P. Paclík, and R. P. W. Duin, "A generalized kernel approach to dissimilarity-based classification," *Journal of Machine Learning Research*, vol. 2, no. 2, pp. 175–211, 2002.

[21] P. L. Douville, "Measured and predicted synthetic aperture radar target comparison," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 1, pp. 25–37, 2002.

[22] I. Naseem, R. Togneri, and M. Bennamoun, "Linear regression for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 11, pp. 2106–2112, 2010.