Principal Composite Kernel Feature Analysis: Data-Dependent Kernel Approach

Yuichi Motai, Senior Member, IEEE, and Hiroyuki Yoshida, Member, IEEE

Abstract—Principal composite kernel feature analysis (PC-KFA) is presented to show kernel adaptations for nonlinear features of medical image data sets (MIDS) in computer-aided diagnosis (CAD). The proposed algorithm PC-KFA has extended the existing studies on kernel feature analysis (KFA), which extracts salient features from a sample of unclassified patterns by use of a kernel method. The principal composite process for PC-KFA herein has been applied to kernel principal component analysis [34] and to our previously developed accelerated kernel feature analysis [20]. Unlike other kernel-based feature selection algorithms, PC-KFA iteratively constructs a linear subspace of a high-dimensional feature space by maximizing a variance condition for the nonlinearly transformed samples, which we call data-dependent kernel approach. The resulting kernel subspace can be first chosen by principal component analysis, and then be processed for composite kernel subspace through the efficient combination representations used for further reconstruction and classification. Numerical experiments based on several MID feature spaces of cancer CAD data have shown that PC-KFA generates efficient and an effective feature representation, and has yielded a better classification performance for the proposed composite kernel subspace using a simple pattern classifier.

Index Terms—Principal component analysis, data-dependent kernel, nonlinear subspace, manifold structures

1 Introduction

Capitalizing on the recent success of kernel methods in pattern classification [62], [63], [64], [65], Schölkopf and Smola [34] developed and studied a feature selection algorithm, in which principal component analysis (PCA) was effectively applied to a sample of n, d-dimensional patterns that are first injected into a high-dimensional Hilbert space using a nonlinear embedding. Heuristically, embedding input patterns into a high-dimensional space may elucidate salient nonlinear features in the input distribution, in the same way that nonlinearly separable classification problems may become linearly separable in higher dimensional spaces as suggested by the Vapnik-Chervonenkis theory [14]. Both the PCA and the nonlinear embedding are facilitated by a Mercer kernel of two arguments $k : R^d \times R^d \to R$, which effectively computes the inner product of the transformed arguments. This algorithm, called kernel principal component analysis (KPCA), thus avoids the problem of representing transformed vectors in the Hilbert space, and enables the computation of the inner-product of two transformed vectors of an arbitrarily high dimension in constant time. Nevertheless, KPCA has two deficiencies: 1) The computation of the principal components involves the solution of an eigenvalue problem that requires $O(n^3)$ computations, and 2) each principal component in the Hilbert space depends on every one of the n input patterns, which defeats the goal of obtaining both an informative and concise representation.

Both of these deficiencies have been addressed in subsequent investigations that seek sets of salient features that only depend upon sparse subsets of transformed input patterns. Tipping [43] applied a maximum-likelihood technique to approximate the transformed covariance matrix in terms of such a sparse subset. Franc and Hlaváč [21] proposed a greedy method, which approximates the mapped space representation by selecting a subset of input data. It iteratively extracts the data in the mapped space until the reconstruction error in the mapped high-dimensional space falls below a threshold value. Its computational complexity is $O(nm^3)$, where n is the number of input patterns and m is the cardinality of the subset. Zheng et al. [56] split the input data into M groups of similar size, and then applied KPCA to each group. A set of eigenvectors was obtained for each group. KPCA was then applied to a subset of these eigenvectors to obtain a final set of features. Although these studies proposed useful approaches, none provided a method that is both computationally efficient and accurate.

To avoid the $O(n^3)$ eigenvalue problem, Mangasarian et al. [16] proposed sparse kernel feature analysis (SKFA), which extracts l features, one by one, using an l_1-constraint on the expansion coefficients. SKFA requires only $O(l^2n^2)$ operations, and is, thus, a significant improvement over KPCA if the number of dominant features is much less than the data size. However, if $l > \sqrt{n}$, then the computational cost of SKFA is likely to exceed that of KPCA.

In this paper, we propose an accelerated kernel feature analysis (AKFA) that generates l sparse features from a data set of n patterns using $O(ln^2)$ operations. Since AKFA is based on both KPCA and SKFA, we analyze the former

. Y. Motai is with the Sensory Intelligent Laboratory, Department of Electrical and Computer Engineering, School of Engineering, Virginia Commonwealth University, PO Box 843068, 601 West Main Street, Richmond, VA 23284-3068. E-mail: ymotai@vcu.edu.
. H. Yoshida is with 3D Imaging Research, Department of Radiology, Harvard Medical School and Massachusetts General Hospital, 25 New Chardon Street, Suite 400C, Boston, MA 02114. E-mail: yoshida.hiro@mgh.harvard.edu.

Manuscript received 24 June 2011; revised 29 Dec. 2011; accepted 10 May 2012; published online 22 May 2012.
Recommended for acceptance by X. Zhu.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2011-06-0373.
Digital Object Identifier no. 10.1109/TKDE.2012.110.

algorithms, that is, KPCA and SKFA, and then describe AKFA in Section 2.

We have evaluated other existing multiple kernel learning (MKL) approaches [66], [68], and found that those approaches do not rely on the data sets to combine and choose the kernel functions very much. The choice of an appropriate kernel function has reflected prior knowledge concerning the problem at hand. However, it is often difficult for us to exploit the prior knowledge on patterns for choosing a kernel function, and how to choose the best kernel function for a given data set is an open question. According to the no free lunch theorem [40] on machine learning, there is no superior kernel function in general, and the performance of a kernel function depends on applications, specifically the data sets. The five kernel functions, linear, polynomial, Gaussian, Laplace, and sigmoid, are chosen because they were known to have good performances [40], [41], [42], [43], [44], [45].

The main contribution of this paper is a principal composite kernel feature analysis (PC-KFA) described in Section 3. In this new approach, the kernel adaptation is employed in the kernel algorithms above KPCA and AKFA in the form of the best kernel selection, engineer a composite kernel which is a combination of data-dependent kernels, and the optimal number of kernel combination. Other MKL approaches combined basic kernels, but our proposed PC-KFA specifically chooses data-dependent kernels as linear composites.

In Section 4, we summarize numerical evaluation experiments based on medical image data sets (MIDs) in computer-aided diagnosis (CAD) using the proposed PC-KFA

1. to choose the kernel function,
2. to evaluate feature representation by calculating reconstruction errors,
3. to choose the number of kernel functions,
4. to composite the multiple kernel functions,
5. to evaluate feature classification using a simple classifier, and
6. to analyze the computation time.

Our conclusions appear in Section 5.

2 Kernel Feature Analysis

2.1 Kernel Basics

Using Mercer's theorem [15], a nonlinear, positive-definite kernel $k : R^d \times R^d \to R$ of an integral operator can be computed by the inner product of the transformed vectors $h(x), (y)$, where $R^d \to H$ denotes a nonlinear embedding (induced by k) into a possibly infinite dimensional Hilbert space H. Given n sample points in the domain $X_n = \{x_i \in R^d | i = 1, \ldots, n\}$, the image $Y_n = \{(x_i) | i = 1, \ldots, n\}$ of X_n spans a linear subspace of at most $(n+1)$ dimensions. By mapping the sample points into a higher dimensional space, H, the dominant linear correlations in the distribution of the image Y_n may elucidate important nonlinear dependences in the original data sample X_n. This is beneficial because it permits making PCA nonlinear without complicating the original PCA algorithm. Let us introduce kernel matrix K as a Hermitian and positive semi-definite matrix that computes the inner product between any finite sequences of inputs $x := \{x_j : j \in N_n\}$ and is defined as

$$K := (K(x_i, x_j) : i, j \in N_n) = ((x_i) \cdot (x_j)).$$

Commonly used kernel matrices are as follows [34]:

- The linear kernel:

$$K(x, x_i) = x^T x_i, \quad (1)$$

- The polynomial kernel:

$$K(x, x_i) = (x^T x_i + offset)^d, \quad (2)$$

- The Gaussian RBF kernel:

$$K(x, x_i) = \exp(-\|x - x_i\|^2 / 2\sigma^2), \quad (3)$$

- The Laplace RBF kernel:

$$K(x, x_i) = \exp(-\sigma \|x - x_i\|), \quad (4)$$

- The sigmoid kernel:

$$K(x, x_i) = \tanh(\alpha_0 x^T x_i + \alpha_1), \quad (5)$$

- The ANOVA RB kernel:

$$K(x, x_i) = \sum_{k=1}^{n} \exp(-\sigma (x^k - x_i^k)^2)^d. \quad (6)$$

- The linear spline kernel in one dimension:

$$K(x, x_i) = 1 + x x_i + \min(x, x_i) - \frac{x + x_i}{2} \min(x, x_i)^2 + \frac{(\min(x, x_i))^3}{3}. \quad (7)$$

Kernel selection is heavily dependent on the specific data set. Currently, the most commonly used kernel functions are the Gaussian and Laplace RBF for general purpose when prior knowledge of the data is not available. Gaussian kernel avoids the sparse distribution while the high degree polynomial kernel may cause the space distribution in large feature space. The polynomial kernel is widely used in image processing while ANOVA RB is often used for regression tasks. The spline kernels are useful for continuous signal processing algorithms that involve B-spline inner-products or the convolution of several spline basis functions. Thus, in this paper, we will adopt only the first five kernels in (1)-(5).

A choice of appropriate kernel functions as a generic learning element has been a major problem since classification accuracy itself heavily depends on the kernel selection. For example, Amari and Wu [66] modified the kernel function by extending the Riemannian geometry structure induced by the kernel. Souza and Carvalho [33] proposed

selecting the hyper planes parameters by using k-fold cross validation and leave-one-out criteria. Ding and Dubchak [57] proposed an ad hoc ensemble learning approach where multiclass k-nearest neighborhood classifiers were individually trained on each feature space and later combined. Damoulas and Girolami [31] proposed the use of four additional feature groups to replace the amino-acid composition. Pavlidis et al. [58] performed feature selection for SVMs by combining the feature scaling technique with the leave-one-out error bound. Chapelle et al. [59] tuned multiple parameters for two-norm SVMs by minimizing the radius margin bound or the span bound. Ong et al. [60] applied semidefinite programming to learn kernel function by hyperkernel. Lanckriet et al. [61] designed kernel matrix directly by semidefinite programming.

MKL has been considered as a solution to make the kernel choice in a feasible manner. Amari and Wu [66] proposed a method of modifying a kernel function to improve the performance of support vector machine classifier based on the Riemannian geometrical structure induced by the kernel function. This idea was to enlarge the spatial resolution around the separating boundary surface by a conformal mapping such that the separability between classes can be increased in the kernel space. The experiments results showed remarkable improvement for generalization errors. Rakotomamonjy et al. [68] adopted MKL method to learn a kernel and associate predictor in supervised learning settings at the same time. This study illustrated the usefulness of MKL for some regressions based on wavelet kernels and on some model selection problems related to multiclass classification problems.

In this paper, we propose a single multiclass kernel machine that is able to operate on all groups of features simultaneously and adaptively combine them. This new framework provides a new and efficient way of incorporating multiple feature characteristics without increasing the number of required classifiers. The proposed approach is based on the ability to embed each object description [47] via the kernel trick into a kernel Hilbert space. This process applies a similarity measure to every feature space. We show in this paper that these similarity measures can be combined in the form of the composite kernel space. We design a new single multiclass kernel machine that can operate composite spaces effectively by evaluating principal components of the number of kernel feature spaces. A hierarchical multiclass model enables us to learn the significance of each source/ feature space, and the predictive term computed by the corresponding kernel weights may provide the regressors and the kernel parameters without resorting to ad hoc ensemble learning, the combination of binary classifiers, or unnecessary parameter tuning.

2.2 Kernel Principal Component Analysis

KPCA uses a Mercer kernel [34] to perform a linear PCA. The gray level image of X_n of computed tomographic colonography (CTC) has been centered so that its scatter matrix of the data is given by $S = \sum_{i=1}^{n} \Phi(x_i)\Phi(x_i)^T$. Eigenvalues λ_j and eigenvectors e_j are obtained by solving

$$\lambda_j e_j = S e_j = \sum_{i=1}^{n} \Phi(x_i)\Phi(x_i)^T e_j = \sum_{i=1}^{n} \langle e_j, \Phi(x_i)\rangle \Phi(x_i), \quad (8)$$

for $j = 1, \ldots, n$. Since Φ is not known, (8) must be solved indirectly as proposed in the next Section. Let us introduce the inner product of the transformed vectors by

$$a_{ji} = \frac{1}{\lambda_j}\langle e_j, \Phi(x_i)\rangle,$$

where

$$e_j = \sum_{i=1}^{n} a_{ji}\Phi(x_i) \quad (9)$$

Multiplying by $\Phi(x_q)^T$ on the left, for $q = 1, \ldots, n$, and substituting yields

$$\lambda_j \langle \Phi(x_q), e_j\rangle = \sum_{i=1}^{n} \langle e_j, \Phi(x_i)\rangle \langle \Phi(x_q), \Phi(x_i)\rangle. \quad (10)$$

Substitution of (9) into (10) produces

$$\lambda_j \left\langle \Phi(x_q), \sum_{i=1}^{n} a_{ji}\Phi(x_i) \right\rangle = \sum_{i=1}^{n}\sum_{k=1}^{n} a_{jk}\langle \Phi(x_k), \Phi(x_i)\rangle \langle \Phi(x_q), \Phi(x_i)\rangle, \quad (11)$$

which can be rewritten as, $\lambda_j K a_j = K^2 a_j$, where K is an $n \times n$ Gram matrix, with the element $k_{ij} = \langle \Phi(x_i), \Phi(x_j)\rangle$ and $a_j = [a_{j1} a_{j2} \ldots a_{jn}]^T$. The latter is a dual eigenvalue problem equivalent to the problem

$$\lambda a_j = K a_j. \quad (12)$$

Note that $\|\lambda a_j\|^2 = 1/\lambda_j$.

For example, we may choose a Gaussian kernel such as

$$k_{ij} = \langle \Phi(x_i), \Phi(x_j)\rangle = \exp\left(\frac{1}{2\sigma^2}\|x_i - x_j\|^2\right). \quad (13)$$

Please note that if the image of X_n (finite sequences of inputs $x := \{x_j : j \in N_n\}$) is not centered in the Hilbert space, we need to use the centered Gram Matrix deduced by Mangasarian et al. [16] by applying the following $\hat{K}$:

$$\hat{K} = K - KT - TK + TKT, \quad (14)$$

where K is the Gram matrix of uncentered data, and

$$T = \begin{bmatrix} \frac{1}{n} & \cdots & \frac{1}{n} \\ \vdots & \ddots & \vdots \\ \frac{1}{n} & \cdots & \frac{1}{n} \end{bmatrix}_{n \times n}.$$

Let us keep the l eigenvectors associated with the l largest eigenvalues, we can reconstruct data in the mapped space:

$$\Phi = \sum_{j=1}^{l} \langle \Phi, e_j\rangle e_j = \sum_{j=1}^{l} \gamma_j e_j,$$

where $\gamma_j = \langle \Phi, e_j\rangle = \sum_{k=1}^{n} a_{jk}\langle \Phi, \Phi_k\rangle = \sum_{k=1}^{n} a_{jk}k_{ik}$. For the experimental evaluation, we introduce the reconstruction square error of each data $\Phi_i, i = 1, \ldots, n$, is

$$\mathrm{Err}_i = \|\Phi_i - \Phi_i'\|^2 = k_{ii} - \sum_{j=1}^{l} \gamma_{ji}^2.$$

The mean square error is $MErr = (1/n)\sum_{i=1}^{n} Err_i$. Using (12), $\lambda_{ji} = \lambda_j a_{ji}$. Therefore, the mean square reconstruction error is

$$MErr = (1/n)\sum_{i=1}^{n} k_{ii} - \sum_{j=1}^{l} \lambda_j a_{ji}^2 .$$

Since $\sum_{i=1}^{n} k_{ii} = \sum_{i=1}^{n} \lambda_i$ and $\sum_{i=1}^{n} a_{ji}^2 = \|a_j\|^2 = 1/\lambda_j$, $MErr = \frac{1}{n}\sum_{i=1}^{n-l} \lambda_i$.

The KPCA algorithm contains an eigenvalue problem of rank n, so the computational complexity of KPCA is $O(n^3)$. In addition, each resulting eigenvector is represented as a linear combination of n terms; the l features depend on n image vectors of X_n. Thus, all data contained in X_n must be retained, which is computationally cumbersome and unacceptable for our applications.

KPCA algorithm [34]
Step 1: Calculate the Gram matrix, which contains the inner products between pairs of image vectors.
Step 2: Use $\lambda_j a_j = K a_j$ to obtain the coefficient vectors a_j for $j=1,\ldots,n$.
Step 3: The projection of $x \in R^d$ along the j-th eigenvector is
$\langle e_j, \Phi(x)\rangle = \sum_{i=1}^{n} a_{ji} \langle \Phi(x_i), \Phi(x)\rangle = \sum_{i=1}^{n} a_{ji} k(x, x_i)$.

2.3 Accelerated Kernel Feature Analysis

AKFA [20] is the method that we have proposed to improve the efficiency and accuracy of SKFA [16]. SKFA improves the computational costs of KPCA, associated with both time complexity and data retention requirements. SKFA was introduced in [16] and is summarized in the following three steps:

SKFA algorithm [16]
Step 1: Compute the matrix $k_{ij} := k(x_i, x_j)$, it costs $O(d.n_2)$ operations, where d is the dimensionality of input space X.
Step 2: Initialize $a_{01},\ldots,a_{0m}=1$, and $idx(\cdot)$ as the empty list, these are the initial scaling for the directions of projection. It costs $O(m)$.
Step 3: For $i=1$ to I repeat(I represent the number of featues to be extracted)
1. Compute the Q values based on $\Phi(x_1),\ldots\Phi(x_m)$ for all directions $\Phi_{i,1},\ldots,\Phi_{i,m-i+1}$. it can be got by $\langle \Phi_j^i, \Phi(x_l)\rangle = a_{0j} k_{jl} + \sum_{t=1}^{i-1} a_{tl} k_{idx(t),l}$. It costs $O(i.m^2)$ steps since we need i operations per dot product. Compute the Q value for each direction Φ_j^i.
2. Perform a maximum search over all Q values ($O(m)$) and pick the corresponding $\Phi_{j_i}^i$ this is the i-th principal direction v_i, and store the corresponding coefficients $a_{1j},\ldots a_{ij}$, set $idx(i)=j$.
3. Compute the new search space to perform orthogonalization by $\Phi_j^{i+1} := \Phi_j^i - v_i \langle \Phi_j^i, v_i\rangle / \|v_i\|^2$. All coefficients have to be stored into a_{ij}. All entries $\Phi(x_j)$, concerning are sorted into a_{jl} with $1 \le l \le m$ respectively. The other coefficients are assigned to atl with $1 \le t \le i$ and $1 \le l \le m$.

AKFA [34] has been proposed by the author in an attempt to achieve further improvements: 1) saves computation time by iteratively updating the Gram matrix, 2) normalizes the images with the l_2 constraint before the l_1 constraint is applied, and 3) optionally discards data that falls below a magnitude threshold δ during updates.

To achieve the computation efficiency described in "1," instead of extracting features directly from the original mapped space, AKFA extracts the ith feature based on the ith updated Gram matrix K^i, where each element is $k_{jk}^i = \langle \Phi_j^i, \Phi_k^i \rangle$.

The second improvement described in "2" above is to revise the l_1 constraint. SKFA treats each individual sample data as a possible direction and computes the projection variances with all data. Since SKFA includes its length in its projection variance calculation, it is biased to select vectors with larger magnitude. We are ultimately looking for a direction with unit length, and when we choose an image vector as a possible direction, we ignore the length and only consider its direction for the improved accuracy of the features.

The third improvement in "3" is to discard negligible data and thereby eliminate unnecessary computations.

AKFA is described in the following three steps and showed the improvements 1-3 [20]. The vector $\hat{x}$ represents the reconstructed new data based on AKFA, and it can be calculated indirectly using the kernel trick:

$$\hat{x} = \sum_{j=1}^{l} \langle \Phi_j^i, \Phi(x)\rangle \Phi_j^i = C_l C_l^T K_i;$$

where $K_i = [k(x_1, idx(1)), k(x_2, idx(2)), \ldots, k(x_i, idx(i))]$. Then the reconstruction error of new data $\hat{x}_i$ $i = n+1,\ldots,n+m$, is represented as

$$Err_i = \|\Phi(x_i) - \hat{x}_i\|^2 = k_{ii} - K_i^T C_l C_l^T K_i .$$

AKFA algorithm [20]
Step 1: Compute the $n \times n$ Gram matrix $k_{ij}=k(x_i,x_j)$, where n is the number of input vectors. This part requires $O(n^2)$ operations.
Step 2: Let l denote the number of features to be extracted. Initialize the $l \times l$ coefficient matrix $\mathbf{C}$ to $\mathbf{0}$, and $idx(\cdot)$ as an empty list which will ultimately store the indices of the selected image vectors, and $\mathbf{C}_{(i-1)}$ is an upper-triangle coefficient matrix. Let us define $\Phi_{idx(i)}^i = \Phi_{idx(i)} - \sum_{t=1}^{i-1} \langle \Phi_{idx(i)}, v_t\rangle v_t$. Initialize the threshold value $\delta=0$ for the reconstruction error. The overall cost is $O(l^2)$.
Step 3: For $i=1$ to l repeat:
1. Using the i-th updated K^i matrix, extract the i-th feature. If $k_{jj}^i < \delta$, the predetermined $\delta > 0$. It is a threshold that determines the number of features we selected. Then discard j-th column and j-th row vector without calculating the projection variance. Use $idx(i)$ to store the index. This step requires $O(n^2)$ operations.
2. Update the coefficient matrix by using $\mathbf{C}_{i,i} = 1/\sqrt{k_{idx(i),idx(i)}^i}$ and $\mathbf{C}_{1:(i-1),i} = -\mathbf{C}_{i,i}\mathbf{C}_{(i-1)}\mathbf{C}_{(i-1)}^T \mathbf{K}_{idx(i)}$, which requires $O(i^2)$ operations.
3. Obtain $\mathbf{K}^{i+1}$, an updated Gram matrix. Neglect all rows and columns containing diagonal elements less than δ. This step requires $O(n^2)$ operations. The total computational complexity is increased to $O(ln^2)$ when no data is being truncated during updating in the AKFA.

The AKFA algorithm also contains an eigenvalue problem of rank n, so the computational complexity of AKFA is step 1 requires $O(n^2)$ operations, Step 2 is $O(l^2)$. Step 3 requires 1 for $O(n^2)$, 2 for $O(l^2)$, and 3 for $O(n^2)$. The total computational complexity is increased to $O(ln^2)$ when no data is being truncated during updating in the AKFA.

2.4 Comparison of the Relevant Kernel Methods

Multiple kernel adoption and combination methods are derived from the principle of empirical risk minimization,

TABLE 1
Overview of Method Comparison for Parameters Tunring

Method	Principle	(dis)Advantage
Empirical risk minimization[73]	averaging the loss function on the training set for unknown distribution	high variance, poor generalization, overfitting
Structural risk minimization[73]	incorporating a regularization penalty into the optimization	low bias, high variance, prevent overfitting
Approximation error[70]	featuring diameter of the smallest sphere containing the training points	expensive computation
Span bound [74]	applying a gradient descent method through learning the distribution of kernel functions	optimal approximation of an upper bound of the prediction risk
Jaakkola-Haussler bound [75]	computing the leave-one-out error and the inequality	loose approximations for bounds
Radius-margin bound[76]	introducing each feature, and calculating the gradient of bound value with the scaling factor	optimal parameters depending on the performance measure
Kernel linear discriminant analysis [77]	extending a noline LDA via a kernel trick	complicated characteristics of kernel discriminant analysis

which performs well in most applications. Actually, to access the expected risk, there is an increasing amount of the literature focusing on the theoretical approximation error bounds with respect to the kernel selection problem, for example, empirical risk minimization, structural risk minimization, approximation error, span bound, Jaakkola-Haussler bound, radius-margin bound, and kernel linear discriminant analysis. The following table lists some comparative methods among the multiple kernel methods.

Those kernel approaches listed in Table 1 have somehow overlapped the principles and (dis)advantages, depending on the nature of data. The proposed method described in Section 3 has a tradeoff in the computational time and accuracy, but outperformed those counterparts, even if bias of the data set exists due to cancer screening purposes. The proposed method works on the condition to measure the adaptability of a kernel to the target data. The introduced alignment measure provides a practical objective for kernel optimization as a method for measuring the fitness between a kernel and the learning task.

3 Principal Composite Kernel Feature Analysis

3.1 Kernel Selections

For kernel-based learning algorithms, the key challenge lies in the selection of kernel parameters and regularization parameters. Many researchers have identified this problem and, thus, have tried to solve it. However, the few existing solutions lack effectiveness, and thus this problem is still underdevelopment or regarded as an open problem. To this end, we are developing a new framework of kernel adaptation. Our method exploits the idea presented in [35], and [36], by exploring data-dependent kernel methodology as follows:

Let $\{x_i, y_i\}(i = 1, 2, \ldots, n)$ be n d-dimensional training samples of the given data, where $y_i = \{-1, 1\}$ represents the class labels of the samples. We develop a data-dependent kernel to capture the relationship among the data in this classification task by adopting the idea of "conformal mapping" [36]. To adopt this conformal transformation technique, this data-dependent composite kernel for $r = 1, 2, 3, 4, 5$ can be formulated as

$$k_r(x_i, x_j) = q_r(x_i)q_r(x_j)p_r(x_i, x_j), \quad (15)$$

where $p_r(x_i, x_j)$ is one kernel among five chosen kernels and $q(\cdot)$, the factor function, takes the following form for $r = 1, 2, 3, 4, 5$:

$$q_r(x_i) = \alpha_{r,0} + \sum_{m=1}^{n} \alpha_{r,m} k_0(x_i, x_m), \quad (16)$$

where $k_0(x_i, x_m) = \exp(-\|x_i - x_m\|^2/2\sigma^2)$ and $\alpha_{r,m}$ is the combination coefficient for the variable of x_m. Let us denote the vectors $\{q_r(x_1), q_r(x_2), \ldots, q_r(x_n)\}^T$ and $\{\alpha_{r,0}, \alpha_{r,1}, \alpha_{r,n}\}^T$ by q_r and $\alpha_r (r = 1, 2, 3, 4, 5)$, respectively, where we have $q_r = K_0\alpha_r$, where K_0 is a $n \times (n + 1)$ matrix given by

$$K_0 = \begin{bmatrix} 1 & k_0(x_1, x_1) & \cdots & k_0(x_1, x_n) \\ 1 & k_0(x_2, x_1) & \cdots & k_0(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & k_0(x_n, x_1) & \cdots & k_0(x_n, x_n) \end{bmatrix}. \quad (17)$$

Let the kernel matrices corresponding to $k(x_i, x_j)$, $p_1(x_i, x_j)$ and $p_2(x_i, x_j)$ be K, P_1, and P_2, respectively. We can express data-dependent kernel K as

$$K = (q_r(x_i)q_r(x_j)p_r(x_i, x_j))_{n \times n}. \quad (18)$$

Defining Q_i as the diagonal matrix of elements $\{q_1(x_1), q_1(x_2), \ldots, q_1(x_n)\}$, we can express (18) as the matrix form

$$K_r = Q_r P_r Q_r. \quad (19)$$

This kernel model was first introduced in [32] and called "conformal transformation of a kernel." We now perform kernel optimization based on the method to find the appropriate kernels for the data set.

The optimization of the data-dependent kernel in (19) is to set the value of combination coefficient vector α_r so that the class separability of the training data in mapped feature space is maximized. For this purpose, Fisher scalar is adopted as the objective function of our kernel optimization. Fisher scalar measures the class separability of the training data in the mapped feature space and is formulated as

$$J = \mathrm{tr}(S_{br})/\mathrm{tr}(S_{wr}), \quad (20)$$

where S_{b1} and S_{b2} represent the "between-class scatter matrices" and S_{w1} and S_{w2} are the "within-class scatter matrices." Suppose that the training data are grouped according to their class labels, i.e., the first n_1 data belong to

one class and the remaining n_2 data belong to the other class ($n_1 + n_2 = n$). Then, the basic kernel matrix P_i can be partitioned as

$$P_r = \begin{matrix} P_{11}^r & : & P_{12}^r \\ \cdots & \cdots & \cdots \\ P_{21}^r & : & P_{22}^r \end{matrix}; \quad (21)$$

where the sizes of the submatrices P_{11}^r, P_{12}^r, P_{21}^r, P_{22}^r, $r = 1; 2; 3; 4; 5$ are $n_1 \times n_1$, $n_1 \times n_2$, $n_2 \times n_1$, $n_2 \times n_2$, respectively.

A close relation between the class separability measure J and the kernel matrices has been established as

$$J(\alpha) = \frac{\alpha^T M_{0r} \alpha}{\alpha^T N_{0r} \alpha}, \quad (22)$$

where

$$M_{0r} = K_0^T B_{0r} K_0 \text{ and } N_{0r} = K_0^T W_{0r} K_0; \quad (23)$$

and for $r = 1; 2; 3; 4; 5$

$$B_{0r} = \begin{matrix} \frac{1}{n_1} P_{11}^r & : & 0 \\ \cdots & \cdots & \cdots \\ 0 & : & \frac{1}{n_2} P_{22}^r \end{matrix} - \frac{1}{n} P_r; \quad (24)$$

$$W_{0r} = \operatorname{diag}(p_{11}^r; p_{22}^r; \ldots; p_{nn}^r) - \begin{matrix} \frac{1}{n_1} P_{11}^r & : & 0 \\ \cdots & \cdots & \cdots \\ 0 & : & \frac{1}{n_2} P_{22}^r \end{matrix}.$$

To maximize $J(\alpha)$ in (22), the standard gradient approach is followed. If matrix N_{0i} is nonsingular, the optimal α that maximizes $J_i(\alpha)$ is the eigenvector corresponding to the maximum eigenvalue of the system, we will drive the following (22) as taking the derivatives

$$M_{0r}\alpha = \lambda N_{0r}\alpha. \quad (25)$$

The criterion for selecting the best kernel function is to find the kernel that produces the largest eigenvalue from (25), i.e.

$$\alpha = \arg\max_{\lambda} \lambda_r^* M_r. \quad (26)$$

The idea behind it is to choose the maximum eigenvector α corresponding to the maximum eigenvalue that can maximize the $J_i(\alpha)$ that will result in the optimum solution. We find the maximum Eigen values for all possible kernel functions and arrange them in descending order to choose the most optimum kernels, such as

$$\lambda_1 > \lambda_3 > \lambda_4 > \lambda_2 > \lambda_5 \quad (27)$$

We choose the kernels corresponding to the largest eigenvalues λ and forming composite kernels corresponding to $\{\lambda_1, \lambda_3 \ldots\}$ as follows.

Kernel Selection Algorithm

Step 1: Group the data according to their class labels. Calculate P_r, K_1 first and then B_{0r}, W_{0r} through which we can calculate M_{0r} and N_{0r} for $r = 1, 2, 3, 4, 5$;

Step 2: Calculate the eigenvalue α_r^* corresponding to maximum eigenvector $\lambda_r^* = \arg\max_{\lambda}(N^{-1}M)$;

Step 3: Arrange the eigenvalues in the descending order of magnitude;

Step 4: Choose the kernels corresponding to most dominant eigenvalues;

Step 5: Calculate $q_r = K_1 \alpha_r^*$;

Step 6: Calculate Q_r and then compute $Q_r P_r Q_r$ for the most dominant kernels.

3.2 Kernel Combinatory Optimization

In this section, we propose a principal composite kernel function that is defined as the weighted sum of the set of different optimized kernel functions [41], [42]. To obtain an optimum kernel process, we define the following composite kernel as

$$K_{comp}(\beta) = \sum_{i=1}^{p} \beta_i Q_i P_i Q_i; \quad (28)$$

where β is the constant scalar value of the composite coefficient, and p is the number of kernels we intend to combine. Through this approach, the relative contribution of both kernels to the model can be varied over the input space. We note that in (28), instead of using K_r as a kernel matrix, we use K_{comp} as a composite kernel matrix. According to [46], K_{comp} satisfies the Mercers condition. We use linear combination of individual kernels to yield an optimal composite kernel using the concept of kernel alignment: "conformal transformation of a kernel." The empirical alignment between kernel k_1 and kernel k_2 with respect to the training set S, is the following quantity metric:

$$A(k_1; k_2) = \frac{\langle K_1, K_2 \rangle_F}{\|K_1\|_F \|K_2\|_F}; \quad (29)$$

where K_i is the kernel matrix for the training set S using kernel function K_i, and $\|K_i\|_F = \sqrt{\langle K_i, K_i \rangle_F}$, $\langle K_i, K_j \rangle_F$ is the Frobenius inner product between K_i and K_j. $S = \{(x_i, y_i) | x_i \in X, y_i \in \{-1, 1\}\}_{i=1}^n$, X is the input space, y is the target vector. Let $K_2 = yy'$, then the empirical alignment between kernel K and target vector y is

$$A(k, yy') = \frac{\langle K, yy' \rangle_F}{\|K\|_F \|yy'\|_F} = \frac{y'Ky}{n\|K\|_F}. \quad (30)$$

It has been shown that if a kernel is well aligned with the target information, there exists a separation of the data with a low bound on the generalization error. Thus, we can optimize the kernel alignment based on training set information to improve the generalization performance of the test set. Let us consider the combination of kernel functions as follows:

$$k(\beta) = \sum_{i=1}^{p} \beta_i k_i; \quad (31)$$

where individual kernels k_i, $i = 1; 2; \ldots; p$ are known in advance. Our purpose is to tune β to maximize $A(k, yy')$ the empirical alignment between $k(\beta)$ and the target vector y. Hence, we have

$$\beta = \arg\max(A(k, yy')), \quad (32)$$

TABLE 2
Colon Cancer Data Set 1 (Low Resolution)

		Portion	Data Portion	Data Size
Training Set	TP	80.0%	31	148
	FP	78.3%	117	
Testing Set	TP	20.0%	8	40
	FP	21.7%	32	

TABLE 3
Colon Cancer Data Set 2

		Portion	Data Portion	Data Size
Training Set	TP	80.0%	16	766
	FP	70.0%	750	
Testing Set	TP	20.0%	16	316
	FP	30.0%	300	

TABLE 4
Colon Cancer Data Set 3

		Portion	Data Portion	Data Size
Training Set	TP	80.0%	22	1012
	FP	70.0%	990	
Testing Set	TP	20.0%	6	431
	FP	30.0%	425	

TABLE 5
Colon Cancer Data Set 4

		Portion	Data Portion	Data Size
Training Set	TP	80.0%	17	1817
	FP	60.0%	1800	
Testing Set	TP	20.0%	4	1204
	FP	40.0%	1200	

$$\rho = \arg\max \left(\frac{\sum_{i,j} \langle K_i, K_j \rangle}{n \sum_i \langle K_i, K_i \rangle^2} \right) = \arg\max \left(\frac{\sum_{i,j} \langle K_i, K_j \rangle}{\sum_{i,j} \langle K_i, K_j \rangle} \right) \quad (33)$$

$$= \arg\max \frac{\delta_P}{n^2} = \arg\max \frac{1}{n^2} \frac{U}{V},$$

where

$$u_i = \langle K_i, vv^T \rangle = v_i$$
$$= \langle K_i, K_j \rangle = U_{ij} = u_i u_j = V_{ij}$$
$$= v_i v_j, \quad = (1, 2, \ldots, p).$$

Let the generalized Raleigh coefficient be

$$J(\delta) = \frac{U}{V} \quad (34)$$

Therefore, we can obtain the value of δ by solving the generalized eigenvalue problem

$$U = \lambda V \quad (35)$$

where λ denotes the eigenvalues.

PC-KFA Algorithm

Step 1: Compute optimum parameter $\hat{\rho}$ in $U\rho = \delta V\rho$.

Step 2: Implement $K_{comp}(\rho)$ for optimum parameter $\hat{\rho}$.

Step 3: Build the model with $K_{comp}(\rho)$ using all training data.

Step 4: Test the completed model on the test set.

The PC-KFA algorithm contains an eigenvalue problem of rank n, so the computational complexity of PC-KFA is Step 1 requires $O(n^2)$ operations, Step 2 is n. Step 3 requires n operations. Step 4 requires n operations. The total computational complexity is increased to $O(n^2)$.

4 Experimental Analysis

4.1 Cancer Image Data Sets

4.1.1 Colon Cancer

This data set consisted of true-positive (TP) and false-positive (FP) detections obtained from our previously developed CAD scheme for the detection of polyps [5], when it was applied to a CTC image database. This database contained 146 patients who underwent a bowel preparation regimen with a standard precolonoscopy bowel-cleansing method. Each patient was scanned in both supine and prone positions, resulting in a total of 292 CT data sets. In the scanning, helical single-slice or multislice CT scanners were used, with collimations of 1.25-5.0 mm, reconstruction intervals of 1.0-5.0 mm, X-ray tube currents of 50-260 mA and voltages of 120- 140 kVp. In-plane voxel sizes were 0.51-0.94 mm, and the CT image matrix size was 512 × 512. Out of 146 patients, there were 108 normal cases and 38 abnormal cases with a total of 39 colonoscopy-confirmed polyps larger than 6 mm.

The CAD scheme was applied to the entire cases and it generated a segmented region for each of its detection (a candidate of polyp). A volume-of-interest (VOI) of size 64 × 64 × 64 voxels was placed at the center of mass of each candidate for encompassing its entire region; then, it was resampled to 12 × 12 × 12 voxels. Resulting VOIs of 39 TP and 149 FP detections from the CAD scheme made up the colon cancer data set 1.

Additional CTC image databases with a similar cohort of patients were collected from three different hospitals in the

TABLE 6
Colon Cancer Data Set 5

		Portion	Data Portion	Data Size
Training Set	TP	58. 8%	80	2080
	FP	55. 3%	2000	
Testing Set	TP	41.2%	56	1668
	FP	54.7%	1612	

TABLE 7
Breast Cancer Data Set

		Portion	Data Portion	Data Size
Training Set	TP	80.0%	51	126
	FP	60.0%	75	
Testing Set	TP	20.0%	13	63
	FP	40.0%	50	

TABLE 8
Lung Cancer Data Set

		Portion	Data Portion	Data Size
Training Set	TP	70.0%	15	126
	FP	80.0%	111	
Testing Set	TP	30.0%	6	34
	FP	20.0%	28	

US. The VOIs obtained from these databases were re-sampled to 16 × 16 × 16 voxels. We refer to the resulting data sets as colon cancer data sets 2, 3, 4, 5, and 6 for Tables 2, 3, 4, 5, and 6 with the distribution of the training and testing VOIs, respectively.

4.1.2 Breast Cancer

We extended our own colon cancer data sets into other cancer-relevant data sets. This data set is available at http:// www.ncbi.nlm.nih.gov/ geo/ query/ acc.cgi?acc = GSE2990. This data set contains data on 189 women, 64 of which were treated with tamoxifen, with primary operable invasive breast cancer, with each feature dimension of 22283. More information on this data set can be found in [50].

4.1.3 Lung Cancer

This data set is available at http:// www.broadinstitute. org/ cgi-in/ cancer/ data sets.cgi. It contains 160 tissue samples, 139 of which are of class '0' and the remaining are of class '2'. Each sample is represented by the expression levels of 1,000 genes for each feature dimension.

4.1.4 Lymphoma

This data set is available at http:// www.broad.mit.edu/ mpr/ lymphoma. It contains 77 tissue samples, 58 of which are diffuse large B-cell lymphomas and the remainder is follicular lymphomas, with each feature dimension of 7,129. Detailed information about this data set can be found in [48].

TABLE 9
Lymphoma Data Set

		Portion	Data Portion	Data Size
Training Set	TP	85.0%	17	62
	FP	78.0%	45	
Testing Set	TP	15.0%	3	14
	FP	22.0%	13	

TABLE 10
Prostate Cancer Data Set

		Portion	Data Portion	Data Size
Training Set	TP	85.0%	64	250
	FP	80.0%	186	
Testing Set	TP	15.0%	11	58
	FP	20.0%	47	

4.1.5 Prostate Cancer

This data set is collected from http:// www.ncbi.nlm.nih. gov/ geo/ query/ acc.cgi?acc = GSE6919. It contains prostate cancer data collected from 308 patients, 75 of which have metastatic prostate tumor and the rest of the cases were normal, with each feature dimension of 12,553. More information on this data set can be found in [51] and [52].

4.2 Kernel Selection

We first evaluate herein the performance on the kernel selection according to the method proposed in Section 3.1, regarding how to select the kernel function that will best fit the data. The larger the eigenvalue is, the greater the class separability measure J in (22) is to be expected. Table 11 shows the calculation of the algorithm for all the data sets mentioned to determine the eigenvalues of all the five kernels. Specifically, we have set the parameters such as d, offset, σ_0, σ_1, and σ of each kernel in (1)-(5), and computed their eigenvalues J for all the nine data sets Tables 2, 3, 4, 5, 6, 7, 8, 9, and 10. After arranging the eigenvalues for each data set in descending order we selected the kernel corresponding to the largest eigenvalue as the optimum kernel.

The largest eigenvalue for each data set is highlighted in Table 11. After evaluating the quantitative eigenvalues for all the nine data sets, we observed that the RBF kernel gives the maximum eigenvalue among all the five kernels. That means that RBF kernel produced the dominant results compared to all other four kernels. For five data sets, colon cancer data sets 2, 3, 4, 5, and 6: Lymphoma cancer data set, the polynomial kernel produced the second largest eigenvalue. Linear kernel gave the second largest eigenvalue for colon cancer data set 1 and lung cancer data set, where as the Laplace kernel produced the second largest eigenvalue for the breast cancer data set.

TABLE 11
Eigenvalues of Five Kernel Functions (1)-(5) and Their Parameters Selected

Cancer Datasets	Linear	Polynomial	Gaussian RBF	Laplace RBF	Sigmoid
Colon1	13.28	11.54	**16.82**	7.87	3.02
		d=1, *offset*=1	σ=4.00	σ=0.1	β_0=2, β_1=1.7
Colon2	75.43	84.07	**139.96**	40.37	64.5
		d=1, *offset*=4	σ=5.65	σ=1.5	β_0=1, β_1=2.5
Colon3	100.72	106.52	**137.74**	80.67	53.2
		d=1, *offset*=1	σ=4.47	σ=1.5	β_0=2, β_1=3
Colon4	148.69	166.44	**192.14**	34.99	142.3
		d=1, *offset*=1	σ=4.58	σ=1.5	β_0=2, β_1=2
Colon5	78. 4	91. 7	**141.72**	88. 1	102.4
		d=1.7, *offset*=0.8	σ =0.7	σ=2.3	β_0=0.01,β_1=0
Breast	22.85	20.43	**64.38**	56.85	23.2
		d=1.2, *offset*=1	σ=4.47	σ=3.0	β_0=0.75, β_1=1
Lung	36.72	47.49	**54.60**	38.74	29.2
		d=1.2, *offset*=4	σ=3.87	σ=2.4	β_0=4, β_1=2.5
Lymphoma	19.71	37.50	**42.13**	35.37	23.6
		d=1.5, *offset*=2	σ=2.82	σ=2.0	β_0=1.5, β_1=2
Prostate	50.93	48.82	**53.98**	40.33	43.1
		d=1, *offset*=1	σ=4.47	σ=1.5	β_0=0.5, β_1=0.5

TABLE 12
Mean-Square Reconstruction Error of KPCA, SKFA, and AKFA with the Selected Kernel Function

Cancer Datasets	Selected Kernel Function	Eigenspace Dimension	KPCA Error (%)	SKFA Error (%)	AKFA Error (%)
Colon1	RBF	75	6.86	11.56	10.74
Colon2	RBF	100	27.08	18.41	17.00
Colon3	RBF	100	14.30	22.29	20.59
Colon4	RBF	100	12.48	19.66	18.14
Colon5	RBF	90	11.89	15. 41	17.90
Breast	RBF	55	6.05	2.10	10.10
Lung	RBF	50	1.53	2.55	7.30
Lymphoma	RBF	20	3.27	7.2	3.87
Prostate	RBF	80	10.33	11.2	13.83

As shown in Table 11, the Gaussian GBF kernel showed largest eigenvalues for the all nine data sets. The performance of the selected Gaussian GBF was compared to the other single kernel function in the reconstruction error value. As a further experiment, the reconstruction error results have been evaluated for KPCA using MErr ¼ ðl=nÞ $\sum_{i=1}^{n}$ $\|\phi_i - \phi_i^{l}\|^2$ and for AKFA and SKFA using $Err_i = \|\phi_i - \phi_i^{l}\|^2 = k_{ii} - K_i^T C_l C_l^T K_i$ with the optimum kernel

TABLE 13
Mean-Square Reconstruction Error of KPCA with Other Four Kernel Functions

Cancer Datasets	Linear Kernel Function (%)	Polynomial Kernel Function (%)	Laplace RBF kernel (%)	Sigmoid kernel Function (%)
Colon1	1739	1739	46.43	238.6
Colon2	12133	33170	90.05	291.1
Colon3	4276	4276	38.41	294.6
Colon4	1972	1972	26.28	228.6
Colon5	1794	1801	29.71	198.6
Breast	477.6	2061	49.63	465.3
Lung	1009	5702	59.51	464.8
Lymphoma	362.5	362.5	63.04	228.5
Prostate	849.8	849.8	67.44	159.8

TABLE 14
Linear Combination for Selected Two Kernel Functions

Cancer Datasets	Two Selected Kernels	Linear Combination of Kernels
Colon1	RBF+Linear	$\hat{\rho}_1$=0.9852, $\hat{\rho}_2$=0.1527
Colon2	RBF+Polynomial	$\hat{\rho}_1$=0.6720, $\hat{\rho}_2$=0.1582
Colon3	RBF+Polynomial	$\hat{\rho}_1$=0.9920, $\hat{\rho}_2$=0.1204
Colon4	RBF+Polynomial	$\hat{\rho}_1$=0.9775, $\hat{\rho}_2$=0.1375
Colon5	RBF+Polynomial	$\hat{\rho}_1$=0.7300, $\hat{\rho}_2$=0.2700
Breast	RBF+Laplace	$\hat{\rho}_1$=0.8573, $\hat{\rho}_2$=0.1386
Lung	RBF+Linear	$\hat{\rho}_1$=0.9793, $\hat{\rho}_2$=0.1261
Lymphoma	RBF+Polynomial	$\hat{\rho}_1$=0.9903, $\hat{\rho}_2$=0.2082
Prostate	RBF+Linear	$\hat{\rho}_1$=0.9756, $\hat{\rho}_2$=0.1219

(RBF) selected from Table 11. We listed up the selected kernel, dimensions of the eigenspace (chosen empirically) and the reconstruction errors of both KPCA, SKFA, and AKFA for all the data sets shown in Table 12.

Table 12 shows that RBF, the single kernel selected, has a relatively small reconstruction error, from 3.27 percent to up to 14.30 percent in KPCA. The reconstruction error of KPCA is less than that of the reconstruction error of AKFA, from 0.6 percent to up to 6.29 percent. The difference in the reconstruction error between KPCA and AKFA increased as the size of the data sets increased. This could be due to the heterogeneous nature of the data sets. The Lymphoma data set produced the least mean square error, whereas the colon cancer data set 3 produced the largest mean-square error for both KPCA and AKFA.

Table 13 shows that the other four kernel functions have much more error than Gaissian RBF shown in Table 12. The difference between Tables 12 and 13 is more than four times larger reconstruction error, and sometimes 20 times when the other four kernel functions are applied.

TABLE 15
Mean-Square Reconstruction Error with Kernel Combinatory Optimization

Cancer Datasets	Eigenspace Dimension	KPCA Error (%)	SKFA Error (%)	AKFA Error (%)	**PC-KFA** (%)
Colon1	75	4.20	6.34	4.30	4.18
Colon2	100	5.53	7.23	5.20	5. 17
Colon3	100	5.23	7.70	7.29	5.21
Colon4	100	10.50	15.17	14.16	10.48
Colon5	90	5.61	5.81	5.76	4.98
Breast	55	2.88	3.47	6.56	2.78
Lung	50	2.43	3.71	3.67	2.44
Lymphoma	20	2.01	3.11	4.44	2.12
Prostate	80	1.34	2.23	1.06	1.28

4.3 Kernel Combination and Reconstruction

After selecting the number of kernels, we select the first p kernels that produced the p largest Eigen values in Table 11, and combine them according to the method proposed in Section 3.2 to yield lesser reconstruction error. The following Table 14 shows the coefficients calculated for the linear combination of kernels. After obtaining the linear coefficients according to (35), we combine the kernels according to (28) to generate the composite kernel matrix $K_{comp}(p)$. The following Table 15 shows the reconstruction error results for both KPCA and AKFA along with the composite kernel $K_{comp}(p)$.

The reconstruction error using two composite kernel functions shown in Table 15 is smaller than the reconstruction error in the single kernel function RBF in Table 12. This would lead us to claim that all nine data sets from the above table made evident that the reconstruction ability of kernel optimized KPCA and AKFA gives enhanced performance to that of single kernel KPCA and AKFA. The specific improvement in the reconstruction error performance is greater by up to 4.27 percent in the case of KPCA, and by up to 5.84 and 6.12 percent in the cases of AKFA and SKFA by mean, respectively. The best improvement of the error performance is observed in PC-KFA by 4.21 percent by mean. This improvement in reconstruction of all data sets is validated using PC-KFA. This successfully shows that the composite kernel produces only a small reconstruction error.

4.4 Kernel Combination and Classification

To analyze how feature extraction methods affect classification performance of polyp candidates, we used the k-nearest neighborhood classifier on the image vectors in the reduced eigenspace. We evaluated the performance of this simple classifier by applying to the kernel feature spaces obtained by KPCA and AKFA with both selected single kernel as well as composite kernel for all the nine data sets. Six nearest neighbors were used for the classification purpose. The classification accuracy was calculated as (TP þ TN)/(TP þ TN þ FN þ FP). The results of classification accuracy showed very high values as shown in Table 16.

TABLE 16
Classification Accuracy Using Six Nearest Neighborhoods for Single-Kernel and Two-Composite-Kernels with KPCA, SKFA, AKFA, and PC-KFA

Cancer Datasets	KPCA single	KPCA composite	SKFA single	SKFA composite	AKFA single	AKFA composite	**PC-KFA**
Colon1	97.50	97.50	92.50	97.50	95.00	95.00	97.61
Colon2	86.02	86.02	86.02	86.02	85.48	86.02	86.13
Colon3	98.61	98.61	98.61	98.61	98.61	98.61	98.82
Colon4	99.67	99.67	99.67	99.67	99.67	99.67	99.70
Colon5	98.12	98.12	98.12	98.12	96.47	95.31	96.47
Breast	87.50	98.41	96.81	98.41	95.21	96.83	98.55
Lung	91.18	97.06	94.12	94.12	91.18	94.12	97.14
Lymphoma	87.50	93.75	93.75	93.75	97.50	93.75	97.83
Prostate	87.96	94.83	91.38	98.28	89.66	98.28	98.56

TABLE 17
Overall Classification Comparison among Other Multiple Kernel Methods

Datasets	Regularized kernel discriminant analysis (RKDA)[32]	L2 Regularization[71]*	Generality Multiple Kernel Learning (GMKL)[69]	**Proposed PC-KFA**
heart	73.21	0.17	NA	81.21
cancer	95.64	NA	NA	95.84
breast	NA	0.03	NA	84.32
ionosphere	87.67	0.08	94.4	95.11
sonar	76.52	0.16	82.3	84.02
Parkinson's	NA	*NA*	92.7	93.17
Musk	NA	*NA*	93.6	93.87
Wpbc	NA	*NA*	80.0	80.56

* misclassification rate，NA：not available

The results from Table 16 indicate that the classification accuracy of the composite kernel is better than that of the single kernel for both KPCA and AKFA in colon cancer data det 1, Breast cancer, Lung Cancer, Lymphoma, and Prostate Cancer; whereas in the case of colon cancer data sets 2, 3, 4, 5, 6, because of the huge size of the data, the classification accuracy is very similar between single and composite kernels. From this quantitative characteristic among the entire nine data sets, we can evaluate that the composite kernel improved the classification performance, and with single and composite kernel cases the classification performance of AKFA is equally good as that of KPCA, from 85.48 percent up to 98.61 percent. The best classification performance has been shown in PC-KFA, up to 99.70 percent.

4.5 Comparisons of Other Composite Kernel Learning Studies

In this section, we make experimental comparisons of the proposed PC-KFA with other popular MKL technique. Such as regularized kernel discriminant analysis (RKDA) for MKL [32], L2 regulation learning [71], and generality MKL [69] in Table 17, as follows.

TABLE 18
PC-KFA Computation Time for Kernel Selection and Operation with KPCA, SKFA, AKFA, and PC-KFA

Cancer Data-sets	KPCA (sec)	SKFA (sec)	AKFA (sec)	PC-KFA (sec)
Colon1	0.266	3.39	0.218	6.12
Colon2	2.891	5.835	1.875	10.01
Colon3	6.83	16.44	3.30	21.25
Colon4	31.92	47.17	11.23	93.41
Colon5	50.78	61.81	19.76	160.4
Breast	0.266	0.717	0.219	1.37
Lung	0.125	0.709	0.0625	1.31
Lymphoma	0.0781	0.125	0.0469	0.27
Prostate	1.703	4.717	1.109	9.31

To evaluate algorithms [32], [71], [69], the eight data sets are used in the binary-class case from the UCI machine learning repository [61], [72]. L2 regularization learning [71] showed miss-classification ratio, which may not be equally comparative to the other three methods. The proposed PC-KFA overperformed these representative approaches. For example, PC-KFA for lung was 94.12 percent, not as good as the performance of SMKL, but better than RKDA for MKL and GMKL. The classification accuracy of RKDA for MKL in Data Set 4 and prostate is better than GMKL. This result indicates PC-KFA is very competitive to the well-known classifiers for multiple data sets.

4.6 Computation Time

We finally evaluate the computational efficiency of the proposed PC-KFA method by comparing its runtime with KPCA and AKFA for all nine data sets as shown in Table 18. The algorithms have been implemented in Matlab R2007b using the statistical pattern recognition toolbox for the Gram matrix calculation and kernel projection. The processor was a 3.2-GHz Intel Pentium 4 CPU with 3 GB of RAM. Runtime was determined using the CPU time command.

For each algorithm, computation time increases with increasing training data size (n), as expected. AKFA requires the computation of a Gram matrix whose size increases as the data size increases. The results from the table clearly indicate that AKFA is faster than KPCA. We also noticed that the decrease in computation time for AKFA compared to KPCA was relatively small, implying that the use of AKFA on a smaller training data set does not yield much advantage over KPCA. However, as the data size increases, the computational gain for AKFA is much larger than that of KPCA as shown in Fig. 1. PC-KFA shows more computational time since the composite data-dependent kernels needs calculations of a Gram matrix and optimization of coefficient parameters.

Fig. 1 illustrates the increase in the computational time of both KPCA as well as AKFA corresponding to increased data. Using Table 18, we listed the sizes of all the data sets in the ascending order from lymphoma (77) to colon cancer data set 4 (3021) on the X-axis versus the respective computational times on the Y-Axis. The red curve indicates the computational time for the KPCA, whereas the blue curve increases the computational time for AKFA for all nine data sets arranged in ascending order of their sizes. This curve clearly shows that as the size of the data

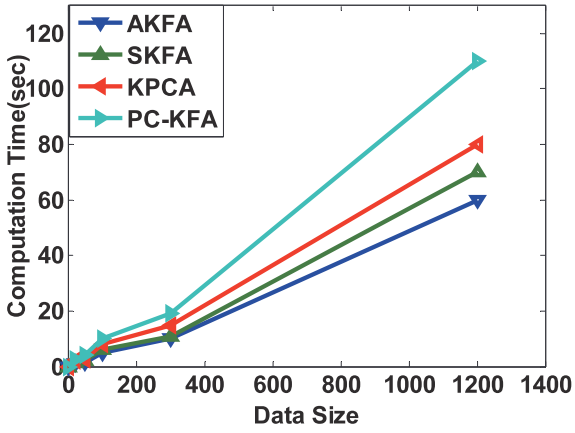

Fig. 1. The computation time comparison between KPCA, SKFA, AKFA, and PC-KFA as the data size increases.

increases, the computational gain of AKFA is greater. This indicates that AKFA is a powerful algorithm and it approaches the performance of KPCA by allowing for significant computational savings.

5 Conclusion

This paper describes first AKFA, a faster and more efficient feature extraction algorithm derived from the SKFA. The time complexity of AKFA is $O(dn^2)$, which has been shown to be more efficient than the $O(d^2n^2)$ time complexity of SKFA and the complexity $O(n^3)$ of a more systematic principal component analysis (KPCA). We have extended these methods into PC-KFA. By introducing a principal component metric in PC-KFA, the new criteria performed well in choosing the best kernel function adapted to the data set, as well as extending this process of best kernel selection into additional kernel functions by calculating linear composite kernel space. We conducted comprehensive experiments using nine cancer data sets for evaluating the reconstruction error, classification accuracy using a k-nearest neighbor classifier, and computational time. The PC-KFA with KPCA and AKFA had a lower reconstruction error compared to single kernel method, thus demonstrating that the features extracted by the composite kernel method are practically useful to represent the data sets. Composite kernel approach with KPCA and AKFA has the potential to yield high detection performance of polyps resulting in the accurate classification of cancers, compared to the single kernel method. The computation time was also evaluated across the variable data sizes, with a tradeoff in the computational time and accuracy, and showed a comparative advantage of composite kernel AKFA.

Acknowledgments

The authors would like to express gratitude for the support of this research by the American Cancer Society Institutional Research Grant through the Massey Cancer Center, Presidential Research Incentive Program at Virginia Commonwealth University, US National Science Foundation ECCS #1054333 (CAREER Award), and USPHS Grant R01CA095279 from National Cancer Institute. The work reported here would not be possible without the help of many of the past and present members of the laboratory, in particular, D. Ma, A. Docef, S. Myla, and L. Winter.

REFERENCES

[1] S. Winawer, R. Fletcher, D. Rex, J. Bond, R. Burt, J. Ferrucci, T. Ganiats, T. Levin, S. Woolf, D. Johnson, L. Kirk, S. Litin, and C. Simmang, "Colorectal Cancer Screening and Surveillance: Clinical Guidelines and Rationale-Update Based on New Evidence," Gastroenterology, vol. 124, pp. 544-560, Feb. 2003.

[2] K.D. Bodily, J.G. Fletcher, T. Engelby, M. Percival, J.A. Christensen, B. Young, A.J. Krych, D.C. Vander Kooi, D. Rodysill, J.L. Fidler, and C.D. Johnson, "Nonradiologists as Second Readers for Intraluminal Findings at CT Colonography," Academic Radiology, vol. 12, pp. 67-73, Jan. 2005.

[3] J.G. Fletcher, F. Booya, C.D. Johnson, and D. Ahlquist, "CT Colonography: Unraveling the Twists and Turns," Current Opinion in Gastroenterology, vol. 21, pp. 90-8, Jan. 2005.

[4] H. Yoshida and A.H. Dachman, "CAD Techniques, Challenges, and Controversies in Computed Tomographic Colonography," Abdominal Imaging, vol. 30, pp. 26-41, Jan./ Feb. 2005.

[5] H. Yoshida and J. Näppi, "Three-Dimensional Computer-Aided Diagnosis Scheme for Detection of Colonic Polyps," IEEE Trans. Medical Imaging, vol. 20, no. 12, pp. 1261-1274, Dec. 2001.

[6] R.M. Summers, C.F. Beaulieu, L.M. Pusanik, J.D. Malley, R.B. Jeffrey Jr., D.I. Glazer, and S. Napel, "Automated Polyp Detector for CT Colonography: Feasibility Study," Radiology, vol. 216, pp. 284-290, 2000.

[7] R.M. Summers, M. Franaszek, M.T. Miller, P.J. Pickhardt, J.R. Choi, and W.R. Schindler, "Computer-Aided Detection of Polyps on Oral Contrast-Enhanced CT Colonography," Am. J. Roentgenology, vol. 184, pp. 105-108, Jan. 2005.

[8] G. Kiss, J. Van Cleynenbreugel, M. Thomeer, P. Suetens, and G. Marchal, "Computer-Aided Diagnosis in Virtual Colonography via Combination of Surface Normal and Sphere Fitting Methods," European Radiology, vol. 12, pp. 77-81, Jan. 2002.

[9] D.S. Paik, C.F. Beaulieu, G.D. Rubin, B. Acar, R.B. Jeffrey Jr., J. Yee, J. Dey, and S. Napel, "Surface Normal Overlap: A Computer-Aided Detection Algorithm with Application to Colonic Polyps and Lung Nodules in Helical CT," IEEE Trans. Medical Imaging, vol. 23, no. 6, pp. 661-675, June 2004.

[10] A.K. Jerebko, R.M. Summers, J.D. Malley, M. Franaszek, and C.D. Johnson, "Computer-Assisted Detection of Colonic Polyps with CT Colonography Using Neural Networks and Binary Classification Trees," Medical Physics, vol. 30, pp. 52-60, Jan. 2003.

[11] J. Näppi and H. Yoshida, "Fully Automated Three-Dimensional Detection of Polyps in Fecal-Tagging CT Colonography," Academic Radiology, vol. 14, pp. 287-300, Mar. 2007.

[12] A.K. Jerebko, J.D. Malley, M. Franaszek, and R.M. Summers, "Multiple Neural Network Classification Scheme for Detection of Colonic Polyps in CT Colonography Data Sets," Academic Radiology, vol. 10, pp. 154-160, Feb. 2003.

[13] A.K. Jerebko, J.D. Malley, M. Franaszek, and R.M. Summers, "Support Vector Machines Committee Classification Method for Computer-Aided Polyp Detection in CT Colonography," Academic Radiology, vol. 12, pp. 479-486, Apr. 2005.

[14] V.N. Vapnik, The Nature of Statistical Learning Theory, second ed. Springer, 2000.

[15] R. Courant and D. Hilbert, Methods of Mathematical Physics, Vol. 1 edition, Wiley-VCH, 1989.

[16] O.L. Mangasarian, A.J. Smola, and B. Schölkopf, "Sparse Kernel Feature Analysis," Technical Report 99-04, Univ. of Wisconsin, 1999.

[17] J. Franc and V. Hlavac, "Statistical Pattern Recognition Toolbox for Matlab," http:// / cmp.felk.cvut.cz/ cmp/ software/ stprtool/ , 2004.

[18] "Partners Research Computing," http:// / www.partners.org/ rescomputing/ , 2006.

[19] A.J. Smola and B. Schölkopf, "Sparse Greedy Matrix Approximation for Machine Learning," Proc. 17th Int'l Conf. Machine Learning, 2000.

[20] X. Jiang, R.R. Snapp, Y. Motai, and X. Zhu, "Accelerated Kernel Feature Analysis," Proc. IEEE CS Conf. Computer Vision and Pattern Recognition, pp. 109-116, 2006.

[21] V. France and V. Hlavăc, "Greedy Algorithm for a Training set Reduction in the Kernel Methods," Computer Analysis of Image and Patterns, vol. 2756, pp. 426-433, 2003.

[22] K. Fukunaga and L. Hostetler, "Optimization of K-Nearest Neighbor Density Estimates," IEEE Trans. Information Theory, vol. IT-19, no. 3, pp. 320-326, May 1973.

[23] J.H. Friedman, "Flexible Metric Nearest Neighbor Classification," technical report, Dept. of Statistics, Stanford Univ., Nov. 1994.

[24] T. Hastie and R. Tibshirani, "Discriminant Adaptive Nearest Neighbor Classification," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 6, pp. 607-616, June 1996.

[25] D.G. Lowe, "Similarity Metric Learning for a Variable-Kernel Classifier," Neural Computation, vol. 7, no. 1, pp. 72-85, 1995.

[26] J. Peng, D.R. Heisterkamp, and H.K. Dai, "Adaptive Kernel Metric Nearest Neighbor Classification," Proc. 16th Int'l Conf. Pattern Recognition, vol. 3, pp. 33-36, 11-15 Aug. 2002.

[27] Q.B. Gao and Z.Z. Wang, "Center-Based Nearest Neighbor Classifier," Pattern Recognition, vol. 40, pp. 346-349, 2007.

[28] S. Li and J. Lu, "Face Recognition Using the Nearest Feature Line Method," IEEE Trans. Neural Networks, vol. 10, no. 2, pp. 439-443, Mar. 1999.

[29] P. Vincent and Y. Bengio, "K-local Hyperplane and Convex Distance Nearest Neighbor Algorithms," Proc. Advances in Neural Information Processing Systems Conf., vol. 14, pp. 985-992, 2002.

[30] W. Zheng, L. Zhao, and C. Zou, "Locally Nearest Neighbor Classifiers for Pattern Classification," Pattern Recognition, vol. 37, pp. 1307-1309, 2004.

[31] T. Damoulas and M.A. Girolami, "Probabilistic Multi-Class Multi-Kernel Learning: On Protein Fold Recognition and Remote Homology Detection," vol. 24, no. 10, pp. 1264-1270, 2008.

[32] J. Ye, S. Ji, and J. Chen, "Multi-Class Discriminant Kernel Learning Via Convex Programming," J. Machine Learning Research, vol. 9, pp. 719-758, 1999.

[33] B. Souza and A. de Carvalho, "Gene Selection Based on Multi-Class Support Vector Machines and Genetic Algorithms," Molecular Research, vol. 4, no. 3, pp. 599-607, 2005.

[34] B. Schölkopf and A.J. Smola, Learning with Kernels, pp. 211-214, MIT Press, 2002.

[35] H. Xiong, Y. Zhang, and X.-W. Chen, "Data-Dependent Kernel Machines for Microarray Data Classification," IEEE/ACM Trans. Computational Biology and Bioinformatics, vol. 4, no. 4, pp. 583-595, Oct.-Dec. 2007.

[36] H. Xiong, M.N.S. Swamy, and M.O. Ahmad, "Optimizing the Data-Dependent Kernel in the Empirical Feature Space," IEEE Trans. Neural Networks, vol. 16, no. 2, pp. 460-474, Mar. 2005.

[37] G.C. Cawley, "MATLAB Support Vector Machine Toolbox, School of Information Systems, Univ. of East Anglia," http:/ / theoval. sys.uea.ac.uk/ ~gcc/ svm/ toolbox, 2000.

[38] Y. Raviv and N. Intrator, "Bootstrapping with Noise: An Efficient Regularization Technique," Connection Science, vol. 8, pp. 355-372, 1996.

[39] H.A. Buchholdt, Structural Dynamics for Engineers. Thomas Telford, 1997.

[40] R.O. Duda, P.E. Hart, and D.G. Stork, Pattern Classification, second ed. John Wiley & Sons Inc., 2001.

[41] Pothin and Richard, "A Greedy Algorithm for Optimizing the Kernel Alignment and the Performance of Kernel Machines," Proc. 14th European Signal Processing Conf. (EUSIPCO '06), Sept. 2006.

[42] J. Kandola, J. Shawe-Taylor, and N. Cristianini, "Optimizing Kernel Alignment over Combinations of Kernels," technical report NeuroCOLT, 2002.

[43] M.E. Tipping, "Spare Kernel Principal Component Analysis," Proc. Neural Information Processing Systems Conf. (NIPS '00), pp. 633-639, 2000.

[44] H. Fröhlich, O. Chappelle, and B. Scholkopf, "Feature Selection for Support Vector Machines by Means of Genetic Algorithm," Proc. IEEE 15th Int'l Conf. Tools with Artificial Intelligence, pp. 142-148, 2003.

[45] S. Mika, G. Ratsch, and J. Weston, "Fisher Discriminant Analysis with Kernels," Proc. Neural Networks for Signal Processing Workshop, pp. 41-48, 1999.

[46] X.W. Chen, "Gene Selection for Cancer Classification Using Bootstrapped Genetic Algorithms and Support Vector Machines," Proc. IEEE Int'l Conf. Computational Systems, Bioinformatics, pp. 504-505, 2003.

[47] C. Park and S.B. Cho, "Genetic Search for Optimal Ensemble of Feature-Classifier Pairs in DNA Gene Expression Profiles," Proc. Int'l Joint Conf. Neural Networks, vol. 3, pp. 1702-1707, 2003.

[48] F.A. Sadjadi, "Polarimetric Radar Target Classification Using Support Vector Machines," Optical Eng., vol. 47, no. 4, p. 046201, Apr. 2008.

[49] T. Briggs and T. Oates, "Discovering Domain Specific Composite Kernels" Proc. 20th Nat'l Conf. Artificial Intelligence and 17th Ann. Conf. Innovative Applications Artificial Intelligence, pp. 732-739, 2005.

[50] N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor, "On Kernel Target Alignment," Proc. Neural Information Processing Systems Conf. (NIPS '01), pp. 367-373.

[51] M.A. Shipp, K.N. Ross, P. Tamayo, A.P. Weng, J.L. Kutok, R.C.T. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G.S. Pinkus, T.S. Ray, M.A. Koval, K.W. Last, A. Norton, T.A. Lister, J. Mesirov, D.S. Neuberg, E.S. Lander, J.C. Aster, and T.R. Golub, "Diffuse Large B-Cell Lymphoma Outcome Prediction by Gene Expression Profiling and Supervised Machine Learning," Nature Medicine, vol. 8, pp. 68-74, 2002.

[52] S. Dudoit, J. Fridlyand, and T.P. Speed, "Comparison of Discrimination Method for the Classification of Tumor Using Gene Expression Data," J. Am. Statistical Assoc., vol. 97, pp. 77-87, 2002.

[53] C. Sotiriou et al., "Gene Expression Profiling in Breast Cancer: Understanding the Molecular Basis of Histologic Grade to Improve Prognosis," J. Nat'l Cancer Inst., vol. 98, no. 4, pp. 262-272, Feb. 2006.

[54] U.R. Chandran et al., "Gene Expression Profiles of Prostate Cancer Reveal Involvement of Multiple Molecular Pathways in the Metastatic Process," BMC Cancer, pp. 7-64, Apr. 2007.

[55] Y.P. Yu et al., "Gene Expression Alterations in Prostate Cancer Predicting Tumor Aggression and Preceding Development of Malignancy," J. Clinical Oncology, vol. 22, no. 14, pp. 2790-2799, July 2004. PMID: 15254046.

[56] W. Zheng, C. Zou, and L. Zao, "An Improved Algorithm for Kernel Principal Component Analysis," Neural Processing Letters, vol. 22, no. 1, pp. 49-56, 2005.

[57] C. Ding and I. Dubchak, "Multi-Class Protein Fold Recognition Using Support Vector Machines and Neural Networks," Bioinformatics, vol. 17, no. 4, pp. 349-358, Apr. 2001.

[58] P. Pavlidis et al., "Learning Gene Functional Classifications from Multiple Data Types," J. Computational Biology, vol. 9, no. 2, pp. 401-411, 2002.

[59] O. Chapelle et al., "Choosing Multiple Parameters for Support Vector Machines," Machine Learning, vol. 46, nos. 1-3, pp. 131-159, 2002.

[60] C.S. Ong, A.J. Smola, and R.C. Williamson, "Learning the Kernel with Hyperkernels," J. Machine Learning Research, vol. 6, pp. 1043-1071, July 2005.

[61] G.R.G. Lanckriet, N. Cristianini, P. Bartlett, L.E. Ghaoui, and M.I. Jordan, "Learning the Kernel Matrix with Semidefinite Programming," J. Machine Learning Research, vol. 5, pp. 27-72, 2004.

[62] V. Atluri, S. Jajodia, and E. Bertino, "Transaction Processing in Multilevel Secure Databases with Kernelized Architecture: Challenges and Solutions," IEEE Trans. Knowledge and Data Eng., vol. 9, no. 5, pp. 697-708, Sept./ Oct. 1997.

[63] H. Cevikalp, M. Neamtu, and A. Barkana, "The Kernel Common Vector Method: A Novel Nonlinear Subspace Classifier for Pattern Recognition," IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 37, no. 4, pp. 937-951, Aug. 2007.

[64] G. Horvath and T. Szabo, "Kernel CMAC With Improved Capability," IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 37, no. 1, pp. 124-138, Feb. 2007.

[65] C. Heinz and B. Seeger, "Cluster Kernels: Resource-Aware Kernel Density Estimators over Streaming Data," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 7, pp. 880-893, July 2008.

[66] S. Amari and S. Wu, "Improving Support Vector Machine Classifiers by Modifying Kernel Functions," Neural Network, vol. 12, no. 6, pp. 783-789, 1999.

[67] Y. Tan and J. Wang, "A Support Vector Machine with a Hybrid Kernel and Minimal Vapnik-Chervonenkis Dimension," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 4, pp. 385-395, Apr. 2004.

[68] A. Rakotomamonjy, F.R. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," J. Machine Learning Research, vol. 9, pp. 2491-2521, Nov. 2008.

[69] M. Varma and B.R. Babu, "More Generality in Efficient Multiple Kernel Learning," Proc. Int'l Conf. Machine Learning, June 2009.

[70] K. Duan, S.S. Keerthi, and A.N. Poo, "Evaluation of Simple Performance Measures for Tuning SVM Hyperparameters," Neurocomputing, vol. 51, pp. 41-59, Apr. 2003.

[71] C. Cortes, M. Mohri, and A. Rostamizadeh, "L2 Regularization for Learning Kernels," Proc. 25th Conf. in Uncertainty in Artificial Intelligence, 2009.

[72] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz, UCI Repository of Machine Learning Databases, URL http://www.ics.uci.edu/mlearn/MLRepository.html. 1998.

[73] K. Chaudhuri, C. Monteleoni, and A.D. Sarwate, "Differentially Private Empirical Risk Minimization," J. Machine Learning Research, vol. 12, pp. 1069-1109, Mar. 2011.

[74] V. Vapnik and O. Chapelle, "Bounds on Error Expectation for Support Vector Machines," Neural Computation, vol. 12, no. 9, pp. 2013-2036, Sept. 2000.

[75] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing Multiple Parameters for Support Vector Machines," Machine Learning, vol. 46, pp. 131-159, 2002.

[76] K.M. Chung, W.C. Kao, C.-L. Sun, L.-L. Wang, and C.J. Lin, "Radius Margin Bounds for Support Vector Machines with the RBF Kernel," Neural Computation, vol. 15, no. 11, pp. 2643-2681, Nov. 2003.

[77] J. Yang, Z. Jin, J.Y. Yang, D. Zhang, and A.F. Frangi, "Essence of Kernel Fisher Discriminant: KPCA Plus LDA," Pattern Recognition, vol. 37, pp. 2097-2100, 2004.

Yuichi Motai (M'01) received the BEng degree in instrumentation engineering from Keio University, Tokyo, Japan, in 1991, the MEng degree in applied systems science from Kyoto University, Japan, in 1993, and the PhD degree in electrical and computer engineering from Purdue University, West Lafayette, Indiana, in 2002. He is currently an associate professor of electrical and computer engineering at Virginia Commonwealth University, Richmond. His research interests include the broad area of sensory intelligence, particularly in online classification and target tracking, applied to medical imaging, pattern recognition, computer vision, and robotics. He is a senior member of the IEEE.

Hiroyuki Yoshida (M'96) received the BS and MS degrees in physics and the PhD degree in information science from the University of Tokyo, Japan, in 1989. He previously held an assistant professorship in the Department of Radiology at the University of Chicago. He was a tenured associate professor when he left the university and joined the Massachusetts General Hospital (MGH) and Harvard Medical School (HMS) in 2005, where he is currently the director of 3D Imaging Research in the Department of Radiology, MGH, and an associate professor of radiology at HMS. His research interests include computer-aided diagnosis and quantitative medical imaging, in particular the diagnosis of colorectal cancers in CT colonography, for which he has been the principal investigator on eight national research projects funded by the National Institutes of Health and two cancer projects by the American Cancer Society, and received nine awards (Magna Cum Laude, two Cum Laude, four Certificate of Merit, and two Excellences in Design) from the Annual Meetings of Radiological Society of North America and Honorable Mention award from the SPIE Medical Imaging. He is the author and coauthor of more than 100 papers and 16 book chapters, author and editor of 10 books, and inventor and coinventor of eight patents. He was a guest editor of the IEEE Transaction on Medical Imaging in 2004, and currently serves on the editorial boards of the International Journal of Computer Assisted Radiology, the International Journal of Computers in Healthcare, and the Intelligent Decision Technology: An International Journal. He is a member of the IEEE.