

Learning Subspace Kernels for Classification

Jianhui Chen^{1,2}, Shuiwang Ji^{1,2}, Betul Ceran^{1,2}, Qi Li³, Mingrui Wu⁴, and Jieping Ye^{1,2}

¹Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287

²Center for Evolutionary Functional Genomics, Arizona State University, Tempe, AZ 85287

³Department of Computer Science, Western Kentucky University, Bowling Green, KY 42101

⁴Yahoo! Inc, Santa Clara, CA 95054

ABSTRACT

Kernel methods have been applied successfully in many data mining tasks. Subspace kernel learning was recently proposed to discover an effective low-dimensional subspace of a kernel feature space for improved classification. In this paper, we propose to construct a subspace kernel using the Hilbert-Schmidt Independence Criterion (HSIC). We show that the optimal subspace kernel can be obtained efficiently by solving an eigenvalue problem. One limitation of the existing subspace kernel learning formulations is that the kernel learning and classification are independent and the subspace kernel may not be optimally adapted for classification. To overcome this limitation, we propose a joint optimization framework, in which we learn the subspace kernel and subsequent classifiers simultaneously. In addition, we propose a novel learning formulation that extracts an uncorrelated subspace kernel to reduce the redundant information in a subspace kernel. Following the idea from multiple kernel learning, we extend the proposed formulations to the case when multiple kernels are available and need to be combined. We show that the integration of subspace kernels can be formulated as a semidefinite program (SDP) which is computationally expensive. To improve the efficiency of the SDP formulation, we propose an equivalent semi-infinite linear program (SILP) formulation which can be solved efficiently by the column generation technique. Experimental results on a collection of benchmark data sets demonstrate the effectiveness of the proposed algorithms.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—*Data Mining*

General Terms

Algorithms

Keywords

Classification, subspace kernel, Hilbert-Schmidt independence criterion, support vector machines

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.
Copyright 2008 ACM 978-1-60558-193-4/08/08 ...\$5.00.

1. INTRODUCTION

Kernel methods have been applied successfully in various data mining tasks such as clustering, regression, and classification [1, 4, 5, 18, 19, 20]. They work by mapping the data from the original input space to a high-dimensional (possibly infinite-dimensional) feature space. The key fact underlying the success of kernel methods is that the embedding into the feature space can be determined implicitly by specifying a symmetric kernel function that computes the dot product between pairs of data points in the feature space.

In many applications, the interesting patterns of the data may lie in a low-dimensional subspace of a certain kernel feature space. Subspace kernel learning that finds such a low-dimensional subspace for effective pattern discovery has received considerable attention recently [13, 16, 17, 26]. In [25], a discriminative subspace kernel learning algorithm was proposed to find a low-dimensional subspace of the kernel feature space. Kernel Target Alignment (KTA) [6] was employed as the learning criterion, resulting in a complex non-linear optimization problem, for which the conjugate gradient algorithm [15] was applied to compute a locally optimal solution.

In this paper, we propose to construct a subspace kernel using the Hilbert-Schmidt Independence Criterion (HSIC) recently proposed for measuring the statistical dependence of random variables [10]. Under HSIC, an optimal subspace kernel maximizes its dependence with the ideal kernel constructed from the class labels. We show that a globally optimal subspace kernel can be obtained efficiently by solving an eigenvalue problem. One limitation of the existing subspace kernel learning formulations is that the kernel learning is independent of the classifier employed subsequently, thus the subspace kernel may not be optimally adapted for classification. To overcome this limitation, we propose a joint optimization framework in which we perform subspace kernel learning and classification simultaneously. In addition, we propose a novel learning formulation that extracts an uncorrelated subspace kernel to reduce redundant information in a subspace kernel.

Following the idea from multiple kernel learning (MKL) [12], we extend the proposed formulations to the case when multiple kernels are available and need to be combined. For example, when we employ the Gaussian kernel for classification, we need to estimate the optimal value of the hyperparameter. Cross-validation is commonly applied for the hyperparameter estimation. In MKL, however, a set of Gaussian kernels (with different hyperparameters) can be integrated for improved classification performance. This is par-

ticularly useful when there exists complementary information among different kernels. In contrast, cross-validation selects only a single best kernel and fails to exploit such complementary information. We show that the integration of (uncorrelated) subspace kernels can be formulated as a semidefinite program (SDP) [2], which is computationally expensive to solve. To improve the efficiency of the SDP formulation, we propose an equivalent semi-infinite linear program (SILP) formulation which can be solved efficiently using the column generation technique [11, 23]. Experimental results on a collection of benchmark data sets demonstrate the effectiveness of the proposed algorithms.

The remainder of this paper is organized as follows: We review the basics of subspace kernel learning and the Hilbert-Schmidt Independence Criterion in Section 2. We present the subspace kernel learning via dependence maximization as well as the joint framework for simultaneous subspace kernel learning and classification in Section 3. We present the concept of uncorrelated subspace kernel and uncorrelated subspace kernel learning in Section 4, followed by subspace kernel integration in Section 5. We report the experimental results in Section 6 and the paper concludes in Section 7.

2. BACKGROUND

In this section, we review the basics of subspace kernel learning and the Hilbert-Schmidt independence criterion.

2.1 Subspace Kernel Learning

In classification tasks, we are given a set of training data $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^m$ and $y_i \in \{1, \dots, k\}$ are the input and output, respectively. In kernel methods, a symmetric function $K : \mathbb{R}^m \times \mathbb{R}^m \rightarrow \mathbb{R}$ is called a *kernel* function if it satisfies the finitely positive semidefinite property [18]. That is, for the input data $\{x_i\}_{i=1}^n \subset \mathbb{R}^m$, the *Gram* (kernel) matrix $G \in \mathbb{R}^{n \times n}$, defined as $G_{ij} = K(x_i, x_j)$, is positive semidefinite. Any valid kernel function K implicitly maps the input data from the input space \mathbb{R}^m to a *Reproducing Kernel Hilbert Space* (RKHS) \mathcal{F} equipped with the inner product $\langle \cdot, \cdot \rangle$ through a mapping function $\phi_K : \mathbb{R}^m \rightarrow \mathcal{F}$ as:

$$K(x, x') = \langle \phi(x), \phi(x') \rangle,$$

for all $x, x' \in \mathbb{R}^m$. Let $X = [x_1, \dots, x_n] \in \mathbb{R}^{m \times n}$ be the data matrix in \mathbb{R}^m , and let $\phi_K(X) = [\phi_K(x_1), \dots, \phi_K(x_n)]$ be the corresponding images of the data in \mathcal{F} .

In *subspace kernel learning* [25], a low-dimensional subspace of the feature space is computed to extract informative features. Let \mathcal{S} be an ℓ -dimensional subspace of \mathcal{F} , and let $Z = [z_1, \dots, z_\ell]$ be a basis of the subspace \mathcal{S} . It follows that each vector in \mathcal{S} can be expressed as a linear combination of $\{\phi_K(x_i)\}_{i=1}^n$, and hence Z can be expressed as:

$$Z = \phi_K(X)W, \quad (1)$$

for some transformation matrix $W \in \mathbb{R}^{n \times \ell}$.

Let $Z = U_z \Sigma_z V_z^T$ be the Singular Value Decomposition (SVD) [9] of Z , where U_z consists of orthonormal columns, $V_z \in \mathbb{R}^{\ell \times \ell}$ is orthogonal, and $\Sigma_z \in \mathbb{R}^{\ell \times \ell}$ is diagonal. Since the subspace \mathcal{S} can be spanned by $\{z_i\}_{i=1}^\ell$, the columns of U_z form an orthonormal basis of the subspace \mathcal{S} . Hence, the projection of $\phi_K(X)$ into \mathcal{S} , denoted as X_w , is given by

$$X_w = U_z^T \phi_K(X). \quad (2)$$

It follows that the kernel matrix in \mathcal{S} is given by [25]:

$$\begin{aligned} G_w &= \langle X_w, X_w \rangle = \phi_K(X)^T U_z U_z^T \phi_K(X) \\ &= \phi_K(X)^T Z (Z^T Z)^+ Z^T \phi_K(X) \\ &= G W (W^T G W)^+ W^T G, \end{aligned} \quad (3)$$

where the second equality above follows from

$$(Z^T Z)^+ = V_z (\Sigma_z^2)^+ V_z^T.$$

Note that $G = \langle \phi_K(X), \phi_K(X) \rangle$ is the (symmetric) kernel matrix computed from \mathcal{F} , and M^+ denotes the pseudo-inverse [9] of the matrix M .

In [25], the subspace kernel is optimized based on the *Kernel-Target Alignment* (KTA) criterion [6]. This leads to a complex nonlinear optimization problem. The conjugate gradient algorithm [15] is applied for computing the solution, which is only locally optimal.

2.2 Hilbert-Schmidt Independence Criterion

Hilbert-Schmidt Independence Criterion (HSIC) is proposed recently for measuring the statistical dependence of random variables [10]. Let \mathcal{F}_x be a RKHS defined on the domain \mathcal{X} associated with the kernel function $K_x : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ and the mapping function $\phi_{K_x} : \mathcal{X} \rightarrow \mathcal{F}_x$, and let \mathcal{F}_y be another RKHS defined on the domain \mathcal{Y} with the kernel function $K_y : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ and the mapping function $\phi_{K_y} : \mathcal{Y} \rightarrow \mathcal{F}_y$. Assume that $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ be drawn from some joint measure p_{xy} (probability distribution), then the cross-variance operator $C_{xy} : \mathcal{F}_y \rightarrow \mathcal{F}_x$ is defined as [7]:

$$C_{xy} = \mathbf{E}_{xy}[(\phi_{K_x}(x) - \mu_x) \otimes (\phi_{K_y}(y) - \mu_y)], \quad (4)$$

where \otimes is the tensor product operator, $\mu_x = \mathbf{E}[\phi_{K_x}(x)]$, and $\mu_y = \mathbf{E}[\phi_{K_y}(y)]$. Given that \mathcal{F}_x and \mathcal{F}_y are separable RKHSs, HSIC is then defined as the squared Hilber-Schmidt norm of the cross-covariance operator C_{xy} given by

$$\text{HSIC}(p_{xy}, \mathcal{F}_x, \mathcal{F}_y) := \|C_{xy}\|_{\text{HS}}^2.$$

HSIC can be expressed in terms of kernels as [10]:

$$\begin{aligned} \mathbf{E}_{xx'yy'}[K_x(x, x')K_y(y, y')] + \mathbf{E}_{xx'}[K_x(x, x')]\mathbf{E}_{yy'}[K_y(y, y')] \\ - 2 \mathbf{E}_{xy}[\mathbf{E}_{x'}[K_x(x, x')]\mathbf{E}_{y'}[K_y(y, y')]], \end{aligned}$$

where (x, y) and (x', y') are two independent pairs drawn independently from p_{xy} , and $\mathbf{E}_{xx'yy'}$ is the expectation over these two pairs. In practice, given a finite set of data pairs $Z = \{(x_i, y_i)\}_{i=1}^n \subset \mathbb{R}^m \times \mathbb{R}$ independently drawn from p_{xy} , the empirical HSIC is estimated by the trace of kernel matrices product as [10]:

$$\text{HSIC}(Z, \mathcal{F}_x, \mathcal{F}_y) := (n-1)^{-2} \text{tr}(G_x P G_y P), \quad (5)$$

where $\text{tr}(\cdot)$ denotes the trace of a matrix, $G_x, G_y \in \mathbb{R}^{n \times n}$ are the kernel matrices given by $(G_x)_{ij} = K_x(x_i, x_j)$, $(G_y)_{ij} = K_y(y_i, y_j)$, and $P = I - ee^T/n$ is the centering matrix, and e is the vector of all ones of length n . In essence, HSIC amounts to computing the trace of the product of two centered kernel matrices.

HSIC has been applied successfully in clustering [21] and supervised feature selection [22] tasks. In this paper, we propose to employ HSIC for subspace kernels learning. This is motivated by a number of appealing features of HSIC [10]: (1) HSIC is an independence measure; (2) HSIC is unbiased and concentrated; and (3) it can be computed efficiently.

3. LEARNING SUBSPACE KERNELS

In this section, we propose to learn a subspace kernel via dependence maximization based on HSIC measure. We also propose a joint framework in which the subspace kernel and Support Vector Machines (SVM) [18] are learned (trained) simultaneously.

3.1 Learning via Dependence Maximization

We propose to learn the subspace kernel in Eq. (3) via dependence maximization. That is, the dependency between the optimal subspace kernel and the ideal kernel derived from the labels is maximized.

Assume that we are given a centered kernel matrix $G \in \mathbb{R}^{n \times n}$, i.e., $Ge = e^T G = 0$, where e is the vector of all ones of length n , and let $y \in \mathbb{R}^n$ be the associated label vector with the i th entry $y_i \in \{1, \dots, k\}$. Following Eq. (3), we propose to construct a regularized subspace kernel \tilde{G}_w as follows:

$$\tilde{G}_w = GW \left(W^T (G + \lambda I) W \right)^{-1} W^T G, \quad (6)$$

where a regularization term controlled by the regularization parameter $\lambda > 0$ is included to avoid the singularity problem. Mathematically, the subspace kernel learning problem based on HSIC can be formulated as the following trace maximization problem:

$$\max_{W \in \mathbb{R}^{n \times \ell}} \text{tr} \left(\tilde{G}_w H(y) \right), \quad (7)$$

where $H(y) \in \mathbb{R}^{n \times n}$ is the *ideal kernel* derived from the label vector $y \in \mathbb{R}^n$. A similar formulation has been proposed for clustering in [21]. It follows from the properties of matrix trace that the optimization problem in Eq. (7) can be reformulated equivalently as follows:

$$\max_{W \in \mathbb{R}^{n \times \ell}} \text{tr} \left(\left(W^T (G + \lambda I) W \right)^{-1} \left(W^T G H(y) G W \right) \right). \quad (8)$$

The optimal transformation W^* to Eq. (8) can be obtained by solving an eigenvalue problem [8], as summarized below.

THEOREM 3.1. *Let G be a centered kernel matrix, and let columns of $V = [v_1, \dots, v_\ell]$ be the first ℓ eigenvectors of $(G + \lambda I)^{-1} G H(y) G$ corresponding to the largest ℓ eigenvalues. Then the optimal W^* to Eq. (8) is given by V .*

The optimal W^* of Eq. (7) is determined by the two kernel matrices G and $H(y)$ derived from the data and the corresponding labels, respectively. With different choices of kernel functions for G and $H(y)$, the optimal W^* forms a family of transformations for subspace kernel learning. The commonly used data space kernels include the linear kernel, polynomial kernel, and RBF kernel for vectorial data, and the string kernel for structured data [18]. Similarly, various choices for the label space kernels can be employed, and we present some representative kernels below:

DEFINITION 1. *Let $y \in \mathbb{R}^n$ be the label vector with the i th entry $y_i \in \{1, \dots, k\}$, and let $Y \in \mathbb{R}^{n \times k}$ be the class indicator matrices defined as: $Y(ij) = 1$ if $y_i = j$ and $Y(ij) = 0$ otherwise. Two representative label kernel matrices are:*

- (1). $H_1(y) = Y Y^T$,
- (2). $H_2(y) = L L^T$, $L = Y (Y^T Y)^{-\frac{1}{2}}$.

Note that both label kernels defined above capture the correlation among labels, but in different ways. Based on the

optimal transformation matrix W^* to Eq. (7), we can construct the optimal subspace kernel \tilde{G}_w as in Eq. (6), which can then be used in kernel machines such as SVM.

3.2 A Joint Learning Framework

Existing approaches for subspace kernel learning learn the subspace kernel without taking into account the subsequent kernel classifiers such as SVM [25]. In the following, we propose a joint framework in which we learn the subspace kernel and SVM simultaneously.

Given a centered kernel matrix $G \in \mathbb{R}^{n \times n}$ and the associated label vector $y \in \mathbb{R}^n$, we define an indicator matrix $\hat{Y} = [\hat{y}_1, \dots, \hat{y}_k] \in \mathbb{R}^{n \times k}$, where $\hat{y}_{ij} = 1$ if $y_i = j$ and $\hat{y}_{ij} = -1$ otherwise. By substituting the subspace kernel in Eq. (6) into the dual formulation of the SVM optimization problem [5], we obtain the following min-max problem:

$$\begin{aligned} \min_{W \in \mathbb{R}^{n \times \ell}} \max_{\alpha_i \in \mathbb{R}^n, \forall i} \sum_{i=1}^k \left(\alpha_i^T e - \frac{1}{2} \alpha_i^T \text{diag}(\hat{y}_i) \tilde{G}_w \text{diag}(\hat{y}_i) \alpha_i \right) \\ \text{subject to } \alpha_i^T \hat{y}_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, k, \end{aligned} \quad (9)$$

where $\{\alpha_i\}_{i=1}^k$ are the vectors of Lagrange dual variables [2], $\text{diag}(\hat{y}_i)$ is a diagonal matrix with the diagonal entries as $\hat{y}_i \in \mathbb{R}^n$, C is the pre-specified tradeoff parameter, and e is the vector of all ones of length n .

The min-max problem in Eq. (9) is difficult to solve directly. However, if one of the two optimization variables (W or $\{\alpha_i\}_{i=1}^k$) is fixed, the other one can be optimized in terms of the fixed one. We thus propose an iterative procedure to solve Eq. (9), in which W and $\{\alpha_i\}_{i=1}^k$ are updated iteratively.

In particular, for a fixed \tilde{G}_w , the optimal $\{\alpha_i^*\}_{i=1}^k$ can be computed by solving the following optimization problem:

$$\begin{aligned} \max_{\alpha_i \in \mathbb{R}^n, \forall i} \sum_{i=1}^k \left(\alpha_i^T e - \frac{1}{2} \alpha_i^T \text{diag}(\hat{y}_i) \tilde{G}_w \text{diag}(\hat{y}_i) \alpha_i \right) \\ \text{subject to } \alpha_i^T \hat{y}_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, k. \end{aligned} \quad (10)$$

Note that the maximization problem in Eq. (10) decouples with a fixed \tilde{G}_w , and thus the optimal $\{\alpha_i^*\}_{i=1}^k$ can be computed independently. In this case, the optimization problem in Eq. (10) is equivalent to k independent SVMs [18].

For a fixed $\{\alpha_i^*\}_{i=1}^k$, the optimal W^* can be computed by solving the following optimization problem:

$$\max_{W \in \mathbb{R}^{n \times \ell}} \sum_{i=1}^k \left(\alpha_i^{*T} \text{diag}(\hat{y}_i) \tilde{G}_w \text{diag}(\hat{y}_i) \alpha_i \right). \quad (11)$$

Let $A_w = [(\text{diag}(\hat{y}_i) \alpha_i), \dots, (\text{diag}(\hat{y}_k) \alpha_k)]$, the optimization problem in Eq. (11) can be reformulated in a compact form given by

$$\max_{W \in \mathbb{R}^{n \times \ell}} \text{tr} \left(\left(W^T (G + \lambda I) W \right)^{-1} \left(W^T G A_w A_w^T G W \right) \right). \quad (12)$$

Similar to Theorem 3.1, the optimal W^* to Eq. (12) is given by the first ℓ eigenvectors of $(G + \lambda I)^{-1} (G A_w A_w^T G)$ corresponding to the largest ℓ eigenvalues.

Based on the discussion above, we propose an iterative optimization procedure for solving Eq. (9). Note that we determine the convergence of the algorithm by computing the relative change of the objective value in Eq. (9), and the iterative procedure stops if the relative change in objective

value is smaller than a pre-specified parameter ϵ or if the number of iterations exceeds a pre-specified number.

4. UNCORRELATED SUBSPACE KERNEL LEARNING

In this section, we introduce the concept of uncorrelated subspace kernel. Similarly, we propose to learn uncorrelated subspace kernel based on HSIC and in a joint framework, respectively.

4.1 Uncorrelated Subspace Kernels

Recall that in Eq. (3), the subspace kernel G_w is generated by projecting the data image $\phi_K(X)$ into the subspace \mathcal{S} using the projection matrix U_z . The projection matrix U_z is required to have orthonormal columns, i.e., $U_z^T U_z = I$. However, redundant information may still exist in \mathcal{S} . We propose *uncorrelated subspace kernels*, i.e., subspace kernels with uncorrelated features, thus reducing the redundant information in the subspace.

Formally, let U_q be the transformation matrix that projects the data image $\phi_K(X)$ from \mathcal{F} into \mathcal{S} as:

$$X_q = U_q^T \phi_K(X), \quad (13)$$

where X_q is the projection of $\phi_K(X)$ in \mathcal{S} . It follows from the *Representer Theorem* [18] that U_q can be expressed as:

$$U_q = \phi_K(X)Q, \quad (14)$$

for some matrix $Q \in \mathbb{R}^{n \times \ell}$ where ℓ is the dimensionality of the subspace. We compute a projection such that the resulting features are orthonormal, that is,

$$\left(U_q^T \phi_K(X) \right) \left(U_q^T \phi_K(X) \right)^T = Q^T G G Q = I. \quad (15)$$

Let $Q = [q_1, \dots, q_\ell]$, and denote $R = Q^T G$ as the data matrix after the projection. It follows that the i th feature component (row vector) of R is $R_i = q_i^T G$, and the covariance between R_i and R_j is

$$\text{Cov}(R_i, R_j) = E(R_i - ER_i)(R_j - ER_j) = q_i^T G G q_j, \quad (16)$$

where the last equality follows since G is centered. It follows that their correlation coefficient is given by

$$\text{Cor}(R_i, R_j) = \frac{q_i^T G G q_j}{\sqrt{q_i^T G G q_i} \sqrt{q_j^T G G q_j}}. \quad (17)$$

Since the features (after projection) are required to be orthonormal as in Eq. (15), we have $\text{Cor}(R_i, R_j) = 0$ if $i \neq j$, and $\text{Cor}(R_i, R_j) = 1$ otherwise. Therefore, the feature vectors in R (the data matrix after projection) are mutually uncorrelated and hence retain minimum redundancy.

The resulting uncorrelated subspace kernel is given by

$$G_q = \left\langle U_q^T \phi_K(X), U_q^T \phi_K(X) \right\rangle = G Q Q^T G. \quad (18)$$

subject to the constraint in Eq. (15). Note that the key difference between the subspace kernel G_w in Eq. (3) and the uncorrelated subspace kernel G_q in Eq. (18) is that the former employs an orthogonal projection, while the latter leads to orthonormal features.

4.2 Learning via Dependence Maximization

We propose to optimize the uncorrelated subspace kernel via dependence maximization as follows:

$$\begin{aligned} & \max_{Q \in \mathbb{R}^{n \times \ell}} \quad \text{tr}(G_q H(y)) \\ & \text{subject to} \quad Q^T (G G + \xi G) Q = I, \end{aligned} \quad (19)$$

where $G_q \in \mathbb{R}^{n \times n}$ is the uncorrelated subspace kernel defined in Eq. (18), $H(y)$ is the ideal kernel derived from the labels as in Definition 1, and $\xi > 0$ is a pre-specified regularization parameter.

The optimal Q^* to Eq. (19) can be obtained by solving a generalized eigenvalue problem, as summarized below.

THEOREM 4.1. *Given a centered kernel matrix G . Let G_q be the subspace kernel matrix defined in Eq. (18), and let $H(y)$ be defined as in Definition 1. Let $V = [v_1, \dots, v_\ell]$ consists of the first ℓ eigenvectors of $(G G + \xi G)^+ G H(y) G$ corresponding to the largest ℓ eigenvalues. Then the optimal Q^* to Eq. (19) is given by $Q^* = V$.*

4.3 A Joint Learning Framework

We propose to learn the uncorrelated subspace kernel and the subsequent SVM classifier simultaneously in a joint framework as follows:

$$\begin{aligned} & \min_{Q \in \mathbb{R}^{n \times \ell}} \max_{\alpha_i \in \mathbb{R}^n, \forall i} \sum_{i=1}^k \left(\alpha_i^T e - \frac{1}{2} \alpha_i^T \text{diag}(\hat{y}_i) G_q \text{diag}(\hat{y}_i) \alpha_i \right) \\ & \text{subject to} \quad \alpha_i^T \hat{y}_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, k, \\ & \quad \quad \quad Q^T (G G + \xi G) Q = I, \end{aligned} \quad (20)$$

where G_q is defined in Eq. (18). The min-max problem in Eq. (20) has optimization variables Q and $\{\alpha_i\}_{i=1}^k$. Similar to the case in Section 3.2, we apply an iterative procedure to solve this optimization problem, in which Q and $\{\alpha_i\}_{i=1}^k$ are updated iteratively.

For a fixed G_q , the optimal $\{\alpha_i^*\}_{i=1}^k$ can be obtained by solving k independent SVMs. For a fixed $\{\alpha_i\}_{i=1}^k$, the optimal Q^* can be obtained by solving the following maximization problem:

$$\begin{aligned} & \max_{Q \in \mathbb{R}^{n \times \ell}} \sum_{i=1}^k \left(\alpha_i^T \text{diag}(\hat{y}_i) G_q \text{diag}(\hat{y}_i) \alpha_i \right) \\ & \text{subject to} \quad Q^T (G G + \xi G) Q = I. \end{aligned} \quad (21)$$

Denoting $A_q = [(\text{diag}(\hat{y}_i) \alpha_i), \dots, (\text{diag}(\hat{y}_k) \alpha_k)]$, then the optimization problem in Eq. (21) can be reformulated as:

$$\begin{aligned} & \max_{Q \in \mathbb{R}^{n \times \ell}} \quad \text{tr} \left(Q^T G A_q A_q^T G Q \right) \\ & \text{subject to} \quad Q^T (G G + \xi G) Q = I. \end{aligned} \quad (22)$$

Similar as in Theorem 4.1, it can be shown that the optimal Q^* to Eq. (22) is given by the first ℓ eigenvectors of $(G G + \xi G)^+ (G A_q A_q^T G)$ corresponding to the largest ℓ eigenvalues.

5. SUBSPACE KERNEL INTEGRATION

In this section, we propose to learn the optimal uncorrelated subspace kernel from a set of pre-specified kernel matrices. We start with the formulation based on the dependence maximization.

Following Eq. (19), the uncorrelated subspace kernel learning problem can be formulated as:

$$\begin{aligned} \max_{\hat{G}_K, Q} \quad & \text{tr} \left(Q^T \hat{G}_K H(y) \hat{G}_K Q \right) \\ \text{subject to} \quad & Q^T (\hat{G}_K \hat{G}_K + \xi \hat{G}_K) Q = I, \end{aligned} \quad (23)$$

where \hat{G}_K is restricted to be a convex combination of p pre-specified kernel matrices $\{G_i\}_{i=1}^p$ as:

$$\hat{G}_K = \sum_{i=1}^p \theta_i G_i, \quad (24)$$

subject to the constraints that $\theta_i \geq 0, i = 1, \dots, p$, and $\sum_{i=1}^p \theta_i \text{tr}(G_i) = 1$. The optimization problem in Eq. (23) performs multiple kernel learning (computation of \hat{G}_K) and subspace kernel learning (computation of Q) simultaneously. However, this optimization problem is nonlinear and hence difficult to solve directly. In the following, we derive an equivalent formulation for this problem which leads to an efficient algorithm.

5.1 Equivalent Formulation

The key observation that leads to the equivalent formulation is that the optimal Q^* to Eq. (23) is given by a closed-form function on \hat{G}_K , and thus it can be factored out from the objective function in Eq. (23).

For notational simplicity, we denote the objective function in Eq. (23) as

$$F(\hat{G}_K, Q) = \text{tr} \left(Q^T \hat{G}_K H(y) \hat{G}_K Q \right). \quad (25)$$

It follows from Theorem 4.1 that, for any fixed (positive semidefinite) \hat{G}_K , the optimal Q^* maximizing $F(\hat{G}_K, Q)$ subject to the constraint in Eq. (23) is given by the first ℓ eigenvectors of $(\hat{G}_K \hat{G}_K + \xi \hat{G}_K)^+ \hat{G}_K H(y) \hat{G}_K$. Moreover, it can be verified that

$$F(\hat{G}_K, Q^*) = \text{tr} \left((\hat{G}_K \hat{G}_K + \xi \hat{G}_K)^+ \hat{G}_K H(y) \hat{G}_K \right). \quad (26)$$

Note that, if $\ell \geq \text{rank}(\hat{G}_K H(y))$, all the nonzero eigenvalues of $(\hat{G}_K \hat{G}_K + \xi \hat{G}_K)^+ \hat{G}_K H(y) \hat{G}_K$ can be captured by $F(\hat{G}_K, Q^*)$ as in Eq. (26). It follows that

$$\begin{aligned} F(\hat{G}_K, Q^*) &= \text{tr} \left(\hat{G}_K (\hat{G}_K \hat{G}_K + \xi \hat{G}_K)^+ \hat{G}_K H(y) \right) \\ &= \text{tr} \left(\left(I - \left(I + \frac{1}{\xi} \hat{G}_K \right)^{-1} \right) H(y) \right). \end{aligned}$$

Thus the maximization problem in Eq. (23) can be reformulated equivalently as:

$$\min_{\hat{G}_K} \text{tr} \left(\left(I + \frac{1}{\xi} \hat{G}_K \right)^{-1} H(y) \right), \quad (27)$$

where \hat{G}_K is constrained as in Eq. (24).

5.2 SDP Formulation

We show that the minimization problem in Eq. (27) can be formulated as a semidefinite program (SDP) [2]. Following Definition 1, let $H(y)$ be decomposed as $H(y) =$

$L_h L_h^T$, where $L_h = [L_{h1}, \dots, L_{hk}] \in \mathbb{R}^{n \times k}$. By introducing variables $\{t_i\}_{i=1}^k$ and following the Schur Complement Lemma [9], we can rewrite the inequalities:

$$L_{hi}^T \left(I + \frac{1}{\xi} \hat{G}_K \right)^{-1} L_{hi} \leq t_i, \quad i = 1, \dots, k, \quad (28)$$

as the following generalized inequalities [2]:

$$\begin{pmatrix} I + \frac{1}{\xi} \hat{G}_K & L_{hi} \\ L_{hi}^T & t_i \end{pmatrix} \succeq 0, \quad i = 1, \dots, k. \quad (29)$$

Let $\theta = [\theta_1, \dots, \theta_p]$ and $r = [r_1, \dots, r_p]$, where $r_i = \text{tr}(G_i)$. The optimization problem in Eq. (23) can be reformulated as the SDP problem given below:

$$\begin{aligned} \min_{\theta, t_i, \forall i} \quad & \sum_{j=1}^k t_j \\ \text{subject to} \quad & \begin{pmatrix} I + \frac{1}{\xi} \sum_{i=1}^p \theta_i G_i & L_{hj} \\ L_{hj}^T & t_j \end{pmatrix} \succeq 0, \forall j, \\ & \theta \geq 0, \quad \theta^T r = 1. \end{aligned} \quad (30)$$

The SDP problem in Eq. (30) can be solved by standard optimization solvers such as SeDuMi [24]. However, it may not be scalable to large data set (a large value of n) due to its positive semidefinite constraints.

5.3 SILP Formulation

We propose to reformulate the maximization problem in Eq. (23) as a semi-infinite program (SIP) [11], which can then be solved more efficiently. The SIP problem refers to optimization problems that maximize a functional $S(a)$ subject to a system of constraints on a , i.e., $s(a, b) \leq 0$ for all b in some set B . When both the objective function and the constraints are linear, the optimization problem is known as semi-infinite linear program (SILP) [23].

It can be shown (using Lagrangian methods) that the optimization problem in Eq. (27) is equivalent to the following min-max problem:

$$\min_{\hat{G}_K} \max_{\beta_i \in \mathbb{R}^n, \forall i} \sum_{j=1}^k \left(-\frac{1}{4} \beta_j^T \beta_j - \frac{1}{4\xi} \beta_j^T \hat{G}_K \beta_j + \beta_j^T L_{hj} \right), \quad (31)$$

where \hat{G}_K is constrained as in Eq. (24), $\{\beta_j\}_{j=1}^k \subset \mathbb{R}^n$ are the dual variables of the optimization problem in Eq. (27), and L_{hj} is defined as in Eq. (30). Let $\hat{\beta} = [\beta_1, \dots, \beta_k]$. We denote $S_i(\hat{\beta})$ (for $i = 1, \dots, p$) as

$$S_i(\hat{\beta}) = \sum_{j=1}^k \left(\frac{1}{4} r_i \beta_j^T \beta_j + \frac{1}{4\xi} \beta_j^T G_i \beta_j - r_i \beta_j^T L_{hj} \right), \quad (32)$$

where $r = [r_1, \dots, r_p]$ is defined as in Eq. (30). The optimization problem in Eq. (31) can be expressed as:

$$\begin{aligned} \max_{\theta} \min_{\hat{\beta}} \quad & \sum_{i=1}^p \theta_i S_i(\hat{\beta}) \\ \text{subject to} \quad & \theta^T r = 1, \quad \theta \geq 0. \end{aligned} \quad (33)$$

By assuming that $\hat{\beta}^*$ is the optimal solution of Eq. (33), we then have

$$\sum_{i=1}^p \theta_i S_i(\hat{\beta}) \geq \sum_{i=1}^p \theta_i S_i(\hat{\beta}^*), \quad \forall \hat{\beta}. \quad (34)$$

Denote $\gamma = \sum_{i=1}^p \theta_i S_i(\hat{\beta})$. The optimization problem in Eq. (33) can be reformulated as:

$$\begin{aligned} \max_{\theta, \gamma} \quad & \gamma \\ \text{subject to} \quad & \theta^T r = 1, \theta \geq 0, \\ & \sum_{i=1}^p \theta_i S_i(\hat{\beta}) \geq \gamma, \quad \forall \hat{\beta}. \end{aligned} \quad (35)$$

The optimization problem in Eq. (35) has two optimization variables (γ and θ) with an infinite number of linear constraints, i.e., one linear constraint for each $\hat{\beta}$. When there is only one fixed $\hat{\beta}$, the optimization problem is simplified as the standard linear program (LP).

5.4 Solving the SILP Formulation

We propose to use the *column generation* technique to solve this SILP problem as in [23]. In this technique, the variables θ and γ are optimized from a restricted subset of constraints in Eq. (35) and this problem is called the *restricted master problem*. Constraints that are not satisfied by current θ and γ are added successively to the restricted master problem until all constraints are satisfied. For fast convergence of the algorithm, it is desirable to add constraint that maximizes the violation for current θ and γ . That is, the $\hat{\beta}$ value that solves

$$\hat{\beta}_\theta = \operatorname{argmin}_{\hat{\beta}} \sum_{i=1}^p \theta_i S_i(\hat{\beta}), \quad (36)$$

is desired. If $\sum_{i=1}^p \theta_i S_i(\hat{\beta}_\theta) \geq \gamma$, then all the constraints are satisfied and θ and γ reach their optimal values. Otherwise, this constraint is added to the restricted master problem and the iteration continues.

It follows from the definition of $S_i(\hat{\beta})$ in Eq. (32) that the problem in Eq. (36) can be written as

$$\min_{\hat{\beta}} \sum_{j=1}^k \left\{ \frac{1}{4} \beta_j^T \beta_j + \frac{1}{4\xi} \beta_j^T \hat{G}_K \beta_j - \beta_j^T L_{hj} \right\}. \quad (37)$$

For a fixed \hat{G}_K (determined by the optimal θ computed from the restricted master problem), the problem in Eq. (37) is an unconstrained convex quadratic program whose optimal solution can be obtained via solving k linear systems of equations as follows:

$$\frac{1}{2} \beta_j + \frac{1}{2\xi} \hat{G}_K \beta_j = L_{hj}, \quad j = 1, 2, \dots, k. \quad (38)$$

After $\hat{\beta} = [\beta_1, \dots, \beta_k]$ is obtained, the corresponding constraint is added to the restricted master problem to update the intermediate θ and γ . Note that the restricted master problem is a linear program. Thus the proposed algorithm for solving the SILP problem alternates between solving k linear systems and a linear program.

5.5 Extension to the Joint Learning Framework

We can further extend the subspace kernel integration formulation to the joint learning framework in Section 4.3.

Following Eq. (20), the joint learning framework with sub-

space kernel integration can be formulated as follows:

$$\begin{aligned} \min_{\hat{G}_K, Q} \max_{\alpha_i \in \mathbb{R}^n, \forall i} \quad & \sum_{i=1}^k \left(\alpha_i^T e - \frac{1}{2} \alpha_i^T \operatorname{diag}(\hat{y}_i) \hat{G}_q \operatorname{diag}(\hat{y}_i) \alpha_i \right) \\ \text{subject to} \quad & Q^T (\hat{G}_K \hat{G}_K + \xi \hat{G}_K) Q = I, \\ & \hat{G}_q = \hat{G}_K Q Q^T \hat{G}_K, \\ & \alpha_i^T \hat{y}_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, \dots, k, \end{aligned} \quad (39)$$

where \hat{G}_K is constrained as in Eq. (24).

The min-max problem in Eq. (39) has optimization variables \hat{G}_K, Q , and $\{\alpha_i\}_{i=1}^k$. For a fixed \hat{G}_K and Q , the optimal $\{\alpha_i^*\}_{i=1}^k$ can be obtained by solving k independent SVMs. For a fixed $\{\alpha_i\}_{i=1}^k$, following a similar derivation as in Section 4.3, the optimal \hat{G}_K^* and Q^* can be obtained by solving the following optimization problem:

$$\begin{aligned} \max_Q \quad & \operatorname{tr} \left(Q^T \hat{G}_K \hat{A}_q \hat{A}_q^T \hat{G}_K Q \right) \\ \text{subject to} \quad & Q^T (\hat{G}_K \hat{G}_K + \xi \hat{G}_K) Q = I, \end{aligned} \quad (40)$$

where $\hat{A}_q = [(\operatorname{diag}(\hat{y}_i) \alpha_i), \dots, (\operatorname{diag}(\hat{y}_k) \alpha_k)]$. The optimization problem in Eq. (40) has the same form as the one in Eq. (23). Hence it can be reformulated as a SDP and a SILP problem following a similar derivation as in Section 5.

6. EXPERIMENTAL STUDY

In this section, we empirically evaluate the proposed subspace kernel learning algorithms and conduct sensitivity studies on various parameters of the algorithms. The algorithms are implemented in MATLAB, and the codes are available at the supplemental website¹.

Seven benchmark data sets are employed in our experiments. Five of them are from UCI Machine Learning Repository²: satimage, waveform, segment, wine, and USPS. Two of them are gene expression data sets³: B. Tumor1 and B. Tumor2. For wine and two gene data sets, we use the entire data sets. For others, we randomly sample 300 data points from each class. All of the data sets are normalized. The statistics of the data set are summarized in Table 1.

Table 1: Statistics of the benchmark data sets.

Data Set	Sample Size	Dimension	Class	Type
Satimage	1800	36	6	UCI
Waveform	900	40	3	UCI
Segment	2100	19	7	UCI
Wine	178	13	3	UCI
USPS	3000	256	10	UCI
B. Tumor1	90	5920	5	Gene
B. Tumor2	50	10367	4	Gene

6.1 Classification Performance

We evaluate the proposed algorithms in terms of classification error rate (in percentage). The reported error rates are averaged over 20 random partitions of the data sets into training and test sets using the ratio 1 : 1.

¹<http://www.public.asu.edu/~jchen74/SKL>

²<http://archive.ics.uci.edu/ml>

³<http://www.gems-system.org>

Table 2: Average classification error rates (with standard derivation) of different subspace kernel learning algorithms over 20 random partitions of the seven data sets. The subspace dimension ℓ is set as the number of classes in each data set. The best performance on each data set is highlighted.

Data Set	SVM _{org}	SKFE	HSIC	uHSIC	SVM _{joint}	uSVM _{joint}
Satimage	8.556 ± 0.402	5.458 ± 0.240	5.326 ± 0.240	5.248 ± 0.263	5.170 ± 0.478	4.502 ± 0.342
Waveform	23.749 ± 0.578	20.938 ± 1.120	17.783 ± 1.112	17.039 ± 1.254	17.032 ± 0.959	16.089 ± 0.487
Segment	10.254 ± 0.376	4.386 ± 0.238	3.691 ± 0.305	3.205 ± 0.011	3.692 ± 0.265	3.301 ± 0.194
Wine	7.917 ± 1.045	3.334 ± 1.549	3.539 ± 1.363	3.224 ± 1.318	3.141 ± 1.874	3.016 ± 1.806
USPS	5.936 ± 0.172	2.263 ± 0.070	2.136 ± 0.060	2.116 ± 0.149	2.021 ± 0.103	2.001 ± 0.056
B. Tumor1	15.600 ± 1.550	6.533 ± 1.390	7.822 ± 1.626	7.134 ± 0.600	7.035 ± 1.364	6.067 ± 1.192
B. Tumor2	25.523 ± 1.144	20.272 ± 1.177	18.647 ± 1.164	18.124 ± 1.291	17.543 ± 1.146	16.179 ± 1.156

Table 3: Average classification error rates of uncorrelated subspace kernel learning formulations with multiple kernel learning schemes (HSIC_{mkl} and SVM_{mkl}) incorporated. Classification performance of individual kernels with uHSIC and uSVM_{joint} applied are used as baseline measures (HSIC_{keri} and SVM_{keri}).

Data Set	HSIC _{ker1}	HSIC _{ker2}	HSIC _{ker3}	HSIC _{ker4}	HSIC _{mkl}	SVM _{ker1}	SVM _{ker2}	SVM _{ker3}	SVM _{ker4}	SVM _{mkl}
Satimage	5.831	8.527	7.082	4.416	4.063	5.927	7.782	6.981	6.845	3.456
Waveform	19.331	17.432	17.579	18.441	16.021	15.681	14.672	16.39	17.781	11.216
Segment	5.489	5.286	4.878	5.674	3.571	4.796	5.694	4.204	6.265	3.033
Wine	7.471	3.948	3.776	3.865	3.965	8.511	3.031	4.200	2.822	2.918
USPS	5.721	5.179	2.222	2.128	2.105	6.345	5.026	2.319	2.622	2.205
B. Tumor1	7.112	6.667	6.212	6.140	5.781	6.781	6.216	6.132	6.127	5.139
B. Tumor2	20.833	18.750	17.708	18.125	15.625	20.917	17.625	16.625	16.667	12.458

6.1.1 Subspace Kernel Learning

We perform a comparative study on the proposed subspace kernel learning algorithms: **HSIC** (learning via dependence maximization), **SVM_{joint}** (joint learning with SVM), and **uHSIC** and **uSVM_{joint}** for learning the uncorrelated subspace kernels accordingly. Our comparative study also includes two baseline algorithms: **SVM_{org}** (classification with SVM on the original kernels) and **SKFE** algorithm proposed in [25]. Following [25], we set the subspace dimension as the number of classes in the corresponding data sets. LIBSVM toolbox [3] is used for solving SVM optimization problems in the following experiments.

In data space, we employ the Gaussian kernel: $K(x, x') = \exp(-\|x - x'\|^2 / \sigma)$. For the UCI data, we apply 5-fold cross-validation to select the best value for the hyperparameter σ from the set $\{10^{-3}, 5 \times 10^{-3}, 10^{-2}, 5 \times 10^{-2}\} \cup \{10^{-1} \times i\}_{i=1}^{10}$. For the gene data, we choose the best σ from $\{10 \times i\}_{i=1}^{20}$. In label space, we use $H_2(y)$ in Definition 1 as the kernel function. The obtained subspace kernels are evaluated using SVM. The parameter C for SVM is tuned from the set $\{1 + 2 \times i\}_{i=1}^9 \cup \{20 + 5 \times i\}_{i=1}^{16} \cup \{100 + 50 \times i\}_{i=1}^{18}$ via cross-validation, and the regularization parameters λ and ξ are tuned from the set $\{10^i\}_{i=-5}^5$.

We present the average (classification) error rates and the standard deviations of the algorithms in Table 2. From Table 2, we have the following major observations: (1) the proposed algorithms achieve smaller error rates than SVM_{org}, and meanwhile they outperform or perform competitively compared to SKFE; (2) the uncorrelated subspace kernel learning algorithms: uHSIC and uSVM_{joint} perform better than HSIC and SVM_{joint}, respectively, which demonstrates the significance of learning uncorrelated subspace kernels; (3) the joint learning formulations: SVM_{joint} and uSVM_{joint} perform favorably among all the compared algorithms, which gives strong support for our rationale of improving classification

performance by learning subspace kernel and SVM classifiers simultaneously; and (4) uSVM_{joint} achieves the best performance among the six compared algorithms on all data sets except Segment.

6.1.2 Multiple Kernel Learning

We evaluate the proposed multiple kernel learning formulations (SILP) in terms of classification error rates. The formulations based on uHSIC and uSVM_{joint} are denoted as **HSIC_{mkl}** and **SVM_{mkl}**, respectively. For all of the data sets, we construct four candidate Gaussian kernels with $\sigma \in \{10^{-3}, 10^{-2}, 5 \times 10^{-1}, 10^{-1}\}$. **HSIC_{mkl}** and **SVM_{mkl}** formulations are applied to compute optimal linear combinations of the candidate kernels, and the obtained kernel combinations are then evaluated using uHSIC and uSVM_{joint}, respectively. Each of the candidate kernels is evaluated by uHSIC and uSVM_{joint}, and their performance are used as the baseline measures, denoted as **HSIC_{keri}** and **SVM_{keri}** ($\forall i \in \{1, 2, 3, 4\}$), respectively.

The experimental results are presented in Table 3. We can observe that HSIC_{mkl} and SVM_{mkl} achieve favorable performance on all of the benchmark data sets. Specifically, on the data sets: satimage, waveform, segment, and USPS, HSIC_{mkl} and SVM_{mkl} achieves better performance (lower error rates) than the corresponding baseline measures. The improved performance resulting from multiple kernel learning may be due to the existence of some complementary information among different kernels. This demonstrates the effectiveness of incorporating multiple kernel learning scheme into uncorrelated subspace kernel learning. We can also observe that, given the same set of kernel matrices, SVM_{mkl} can achieve smaller error rate than HSIC_{mkl}, showing its enhanced ability to explore the informative domain knowledge underlying the data by jointly learning subspace kernel and SVM classifier.

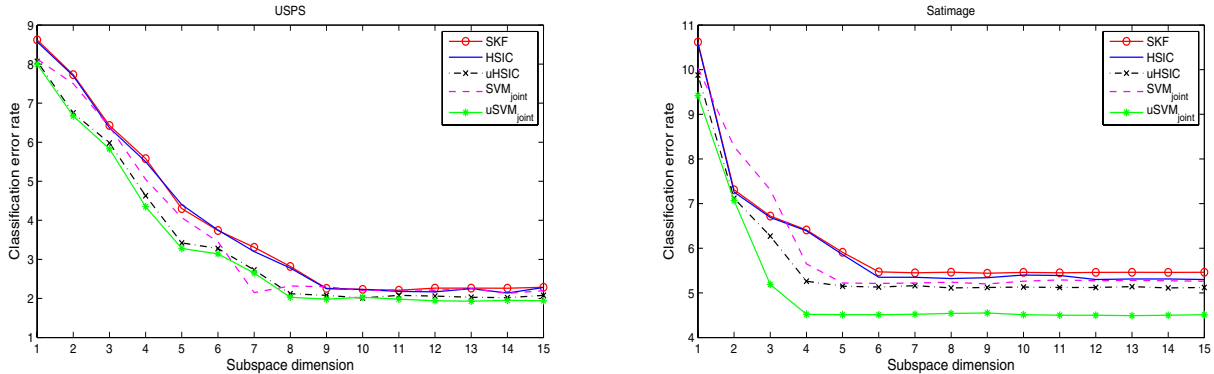


Figure 1: Classification error rates of subspace kernel learning algorithms with different subspace dimensionalities on USPS (right figure) and satimage (left figure) data sets.

6.2 Sensitivity Studies

We perform sensitivity studies on the various parameters of the proposed subspace kernel learning algorithms.

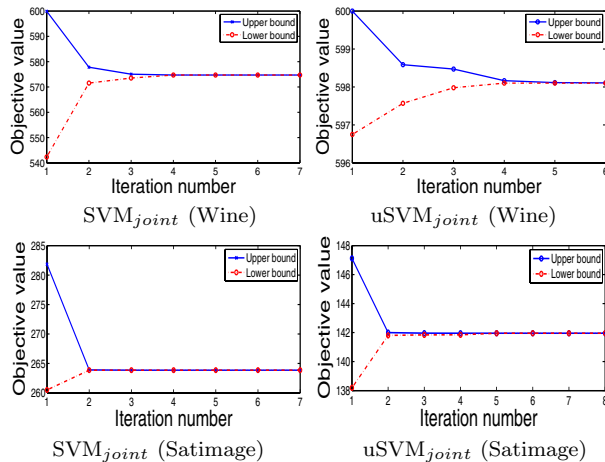


Figure 2: Convergence plots of the iterative procedure for solving the min-max optimization problems in SVM_{joint} (left column) and $uSVM_{joint}$ (right column) on the wine and satimage data sets. In each iteration, the objective value of the maximization and minimization problems are denoted as Upper bound and Lower bound, respectively.

Effect of Subspace Dimension We vary the subspace dimensionality (ℓ) from 1 to 15 for the subspace kernel learning algorithms (the proposed formulations and SKFE), and study the corresponding change of classification performance. We use USPS and satimage data sets for this experiments, and the experimental results (error rates) are depicted in Figure 1. We can observe that, for all of the compared algorithms, the error rates decrease with the increase of the subspace dimensionality (ℓ) when ℓ is smaller than the number of classes (k) in the data. We also observe that the algorithms generally achieve the smallest error rates when ℓ is close to $k - 1$.

Algorithm Convergence We study the convergence property of the iterative procedure for solving the min-max optimization problems in SVM_{joint} and $uSVM_{joint}$ on wine and

satimage data sets. We plot the change of objective values with respect to iterations (for the minimization and maximization problems separately) in Figure 2. We can observe that the upper bound and lower bound are approaching to each other within a small number of iterations. It follows from the theory for min-max problems in [14] that the iterative procedure converges to the saddle point of the optimization problems.

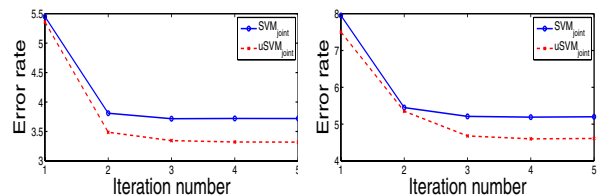


Figure 3: The change of performance for SVM_{joint} and $uSVM_{joint}$ with training iterations on two data sets: Segment (left plot) and satimage (right plot).

Subspace Kernel Optimization We study the successive optimization effect on the subspace kernel in the iterative procedure of solving SVM_{joint} and $uSVM_{joint}$. After each iteration, we evaluate the obtained intermediate subspace kernel in terms of classification error rate. In this experiment, we use the data sets: segment and satimage, and the results are presented in Figure 3. We can observe that the algorithms: SVM_{joint} and $uSVM_{joint}$ improve the resulting classification performance, and generally converge to the best performance within 3 or 4 iterations.

6.3 Efficiency Comparison

We investigate the computation time of the proposed subspace kernel learning algorithms, and compare them with SKFE. The average computation time over each data set is presented in Table 4. We can observe that HSIC and uHSIC have comparable computation time, while SKFE requires relatively larger amount of computation time. Since SVM_{joint} and $uSVM_{joint}$ involve quadratic programs, they have relatively higher computation cost. It is worth noting that SVM_{joint} and $uSVM_{joint}$ generally achieve the best classification performance among the compared algorithms, while they have higher computational costs.

Table 4: Average computation time (in seconds) for the subspace kernel learning algorithms.

Data Set	HSIC	uHSIC	SVM _{joint}	uSVM _{joint}	SKFE
Satimage	3.594	3.359	15.015	21.360	4.547
Waveform	0.359	0.375	1.422	2.297	1.251
Segment	3.656	2.891	17.812	31.188	7.125
Wine	0.016	0.016	0.109	0.141	0.203
USPS	14.869	15.000	61.453	142.359	19.422
B. Tumor1	0.016	0.016	0.063	0.078	0.234
B. Tumor2	0.016	0.016	0.047	0.047	0.234

7. CONCLUSION

We study the problem of learning subspace kernels for classification. We propose to construct a subspace kernel using the Hilbert-Schmidt Independence Criterion. We show that an optimal subspace kernel can be computed effectively by solving an eigenvalue problem. We further propose a joint framework in which we learn the subspace kernel and the subsequent kernel classifier simultaneously. In addition, we propose to learn uncorrelated subspace kernels to reduce redundant information in the subspace kernel. We extend the proposed formulations to the case when multiple kernels are available and need to be combined, following the idea in multiple kernel learning. We show that the integration of subspace kernels can be formulated as a semidefinite program (SDP). To improve the efficiency of the SDP formulation, we propose an equivalent semi-infinite linear program (SILP) formulation which can be solved efficiently. We have conducted experiments on a collection of benchmark data sets. Experimental results demonstrate the effectiveness of the proposed algorithms.

Our subspace kernel integration is based on the uncorrelated subspace kernel learning formulations. The derivation presented in this paper can not be directly extended to the original subspace kernel learning formulations. We plan to explore this further in the future. We plan to apply the proposed kernel integration formulation to real-world applications involving multiple data sources as in [12, 27].

Acknowledgment

This research is supported in part by funds from the Arizona State University and the National Science Foundation (NSF) under Grant No. IIS-0612069.

8. REFERENCES

- [1] G. H. Bakir, T. Hofmann, B. Schölkopf, A. J. Smola, B. Taskar, and S. V. N. Vishwanathan. *Predicting Structured Data*. MIT Press, 2007.
- [2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- [3] C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines. 2001.
- [4] N. Cristianini and M. Hahn. *Introduction to Computational Genomics*. Cambridge University Press, 2006.
- [5] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.
- [6] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. S.

- Kandola. On kernel-target alignment. In *NIPS*, pages 367–373, 2001.
- [7] K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.
- [8] K. Fukunaga. *Introduction to Statistical Pattern Classification*. Academic Press, 1990.
- [9] G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, 1996.
- [10] A. Gretton, O. Bousquet, A. J. Smola, and B. Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *ALT*, pages 63–77, 2005.
- [11] R. Hettich and K. O. Kortanek. Semi-infinite programming: theory, methods, and applications. *SIAM Review*, 35(3):380–429, 1993.
- [12] G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.
- [13] S. Mika, G. Rätsch, and K.-R. Müller. A mathematical programming approach to the kernel fisher algorithm. In *NIPS*, pages 591–597, 2000.
- [14] A. Nemirovski. *Efficient methods in convex programming*, 1994. Lecture Notes.
- [15] J. Nocedal and S. J. Wright. *Numerical Optimization Springer series in operations research*. Springer, 1999.
- [16] C. H. Park and H. Park. Nonlinear feature extraction based on centroids and kernel functions. *Pattern Recognition*, 37(4):801–810, 2004.
- [17] R. Rosipal and L. J. Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research*, 2:97–123, 2001.
- [18] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2001.
- [19] B. Schölkopf, K. Tsuda, and J.-P. Vert. *Kernel Methods in Computational Biology*. MIT Press, 2004.
- [20] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.
- [21] L. Song, A. J. Smola, A. Gretton, and K. M. Borgwardt. A dependence maximization view of clustering. In *ICML*, pages 815–822, 2007.
- [22] L. Song, A. J. Smola, A. Gretton, K. M. Borgwardt, and J. Bedo. Supervised feature selection via dependence estimation. In *ICML*, pages 823–830, 2007.
- [23] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006.
- [24] J. F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12:625–653, 1999.
- [25] M. Wu and J. D. R. Farquhar. A subspace kernel for nonlinear feature extraction. In *IJCAI*, pages 1125–1130, 2007.
- [26] T. Xiong, J. Ye, Q. Li, R. Janardan, and V. Cherkassky. Efficient kernel discriminant analysis via QR decomposition. In *NIPS*, 2004.
- [27] J. Ye and et al. Heterogeneous data fusion and analysis for alzheimer’s disease study. In *KDD*, 2008.