METRIC-CONSTRAINED KERNEL UNION OF SUBSPACES

Tong Wu and Waheed U. Bajwa

Department of Electrical and Computer Engineering, Rutgers, The State University of New Jersey
Emails: {tong.wu.ee, waheed.bajwa}@rutgers.edu

ABSTRACT

This paper addresses the problem of learning a collection of nonlinear manifolds. Inspired by kernel methods, it puts forth a generalization of the kernel subspace model, termed the Metric-Constrained Kernel Union-of-Subspaces (MC-KUoS) model. It then develops an iterative method for learning of an MC-KUoS whose solution is based on the data representation capability of the manifolds and distances between subspaces in the kernel (feature) space. The proposed method (when using Gaussian and polynomial kernels) outperforms existing competitive state-of-the-art methods for real-world image denoising, which shows the benefits of the MC-KUoS model and the proposed denoising approach.

Index Terms— Data-driven learning, image denoising, kernel trick, manifold learning, union of subspaces.

1. INTRODUCTION

Many information processing methods are based on the maxim that high-dimensional data often lie on or near some low-dimensional geometric structures. Recovery of such low-dimensional geometric structures embedded in a high-dimensional ambient space and transforming data into low-dimensional representations not only help us exhibit relevant information within them, but also facilitate processing and computations significantly. Various techniques have been proposed in the literature to learn the geometry underlying data using different manifold models [1–8]. Some works in hybrid linear modeling and clustering are aimed at approximating the data using a collection of subspaces [1, 6]. On the other hand, works like [2–4] attempt to preserve global/local geometric properties of the data in their low-dimensional representations.

Kernel methods [9] have proven to be very useful in extracting the nonlinear characteristics of data. The fundamental theme of kernel methods is to map the data from a nonlinear manifold $\mathcal{M} \subseteq \mathbb{R}^m$ to a very high-dimensional feature space $\mathcal{H}$ via a nonlinear mapping $\phi : \mathcal{M} \to \mathcal{H}$. For a given kernel function $k : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$, any point $y \in \mathcal{M}$ is mapped to a feature vector $\phi(y)$ in a *Reproducing Kernel Hilbert Space* (RKHS) $\mathcal{H}$ such that for all $y, y' \in \mathcal{M}$, we have $k(y, y') = \langle \phi(y), \phi(y') \rangle$. The problem of learning the manifold $\mathcal{M}$ can be ultimately formulated in terms of the kernel matrix. Some useful kernels include Gaussian kernel and polynomial kernel. Interestingly, many nonlinear manifold models [2–4] can be viewed as the *kernel subspace* model, which states that the nonlinear mapping of data to $\mathcal{H}$ lie near a low-dimensional subspace. We refer the reader to [10] for a discussion of the connection between manifold learning algorithms and the kernel PCA [9].

Our Contributions: Kernel subspace model has been shown to be successful in many applications [11, 12]. But the use of a single subspace in the feature space can sometimes require a large dimension of the subspace to capture salient information of the entire data. In order to address this problem, we put forth a natural generalization of the kernel subspace model, termed the *metric-constrained kernel union-of-subspaces* (MC-KUoS) model. The MC-KUoS model asserts that there exists a nonlinear map $\phi : \mathcal{M} \to \mathcal{H}$ such that the ϕ-mapped "images" of signals describing *similar phenomena* (i) belong to a union of low-dimensional subspaces in the feature space $\mathcal{H}$, and (ii) the individual subspaces are also close to each other with respect to a metric defined on the Grassmann manifold in $\mathcal{H}$. The MC-KUoS model can also be regarded as an extension of our recently proposed *metric-constrained union-of-subspaces* (MC-UoS) model [13] for highly nonlinear data (e.g., handwritten digits). In this paper, we propose an iterative algorithm for learning of an MC-KUoS using the kernel trick [14]. In order to demonstrate the validity of MC-KUoS model and the effectiveness of our learning algorithm, we carry out numerical experiments involving Gaussian and polynomial kernels for the denoising task. Results of these experiments show that our approach outperforms other kernel subspace methods.

Notation: Throughout the paper, we use lower-case and upper-case letters for vectors and matrices, respectively. The i-th element of a vector v is denoted by $v(i)$ and the (i, j)-th element of a matrix A is denoted by $a_{i,j}$. The $m \times m$ identity matrix is denoted by I_m. Given a set Ω, $[A]_{\Omega,:}$ (resp., $[v]_\Omega$) denotes the submatrix of A (resp., subvector of v) corresponding to the rows of A (resp., entries of v) indexed by Ω. Given two sets Ω_1 and Ω_2, $[A]_{\Omega_1,\Omega_2}$ denotes the submatrix of A corresponding to rows and columns indexed by Ω_1 and Ω_2, respectively. Finally, $(\cdot)^T$ and $\mathrm{tr}(\cdot)$ denote transpose and trace operations, respectively, while $\|\cdot\|_F$ and $\|\cdot\|_p$ denote Frobenius norm and ℓ_p norm of matrices and vectors, respectively.

2. PROBLEM FORMULATION

In this section, we rigorously formulate the problem studied in this paper. Let $\mathcal{Y} \subseteq \mathbb{R}^m$ be an m-dimensional input space and $\mathcal{H} \subseteq \mathbb{R}^{\widetilde{m}}$ denote the feature space. In practice, $\widetilde{m}$ is usually much larger than m. The nonlinear map $\phi : \mathcal{Y} \to \mathcal{H}$ is implicitly induced by a *positive definite kernel* function $k : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ that describes the similarity between two points in the Hilbert space $\mathcal{H}$. The basic premise in this paper is that the data mapped to $\mathcal{H}$ lie near a union of L subspaces in the feature space; that is, $\mathcal{U} \equiv \phi(\mathcal{Y}) = \bigcup_{\ell=1}^{L} \mathcal{S}_\ell \subset \mathcal{H}$. We make a simplified assumption that all the subspaces have the same dimension s, i.e., $\forall \ell, \dim(\mathcal{S}_\ell) = s$. Then each subspace $\mathcal{S}_\ell$ corresponds to a point on the Grassmann manifold $\mathcal{G}_{\widetilde{m},s}$, which denotes the set of s-dimensional subspaces in $\mathbb{R}^{\widetilde{m}}$. This means the data can be considered as lying in a union of s-dimensional nonlinear manifolds in $\mathbb{R}^m$, i.e., $\mathcal{Y} = \bigcup_{\ell=1}^{L} \mathcal{M}_\ell$. Here, we assume L and s are known a priori. Data-driven estimation of L and s will be considered in future work. Now, if data in the input space describe similar phenomena then we expect the individual subspaces $\mathcal{S}_\ell$'s in

This work is supported in part by the Army Research Office under grant W911NF-14-1-0295 and by an Army Research Lab Robotics CTA subaward.

the feature space to be close to each other with respect to a metric d_u defined on $\mathcal{G}_{\widetilde{m},s}$ in $\mathcal{H}$. This heuristic leads to the following definition of a *metric-constrained kernel union-of-subspaces* (MC-KUoS).

Definition 1. ***(Metric-Constrained Kernel Union-of-Subspaces.)*** *A union of manifolds $\mathcal{Y} = \bigcup_{\ell=1}^{L} \mathcal{M}_\ell$ is said to be a metric-constrained kernel union-of-subspaces with respect to a metric $d_u : \mathcal{G}_{\widetilde{m},s} \times \mathcal{G}_{\widetilde{m},s} \to [0,\infty)$ under the mapping ϕ if $\max_{\ell,p:\ell\neq p} d_u(\mathcal{S}_\ell, \mathcal{S}_p) \leq \epsilon$ for some positive constant ϵ.*

The MC-KUoS model can be considered a nonlinear generalization of the MC-UoS model proposed in [13]. In order to quantify closeness between subspaces on $\mathcal{G}_{\widetilde{m},s}$, we again use the metric defined in [15]. Specifically, if $D_\ell, D_p \in \mathbb{R}^{\widetilde{m}\times s}$ are orthonormal bases for $\mathcal{S}_\ell$ and $\mathcal{S}_p$, then

$$d_u(\mathcal{S}_\ell, \mathcal{S}_p) = \sqrt{s - \mathrm{tr}(D_p^T D_\ell D_\ell^T D_p)} = \|D_p - P_{\mathcal{S}_\ell} D_p\|_F, \quad (1)$$

where $P_{\mathcal{S}_\ell}$ denotes the projection operator $P_{\mathcal{S}_\ell} = D_\ell D_\ell^T$. In order to learn an MC-KUoS, assume we are given a set of N signals $Y = \{y_i\}_{i=1}^N \in \mathbb{R}^{m\times N}$ that correspond to samples drawn from an MC-KUoS $\mathcal{Y}$. The samples in Y can be transformed to a matrix $\phi(Y) = [\phi(y_1), \ldots, \phi(y_N)]$ via the nonlinear mapping ϕ. We make an assumption that all the $\phi(y_i)$'s are linearly independent, i.e., $\mathrm{rank}(\phi(Y)) = N$, which is valid since we usually have $\widetilde{m} \gg N$. This implies the kernel matrix $G = \phi(Y)^T\phi(Y) \in \mathbb{R}^{N\times N}$, with individual entries defined as $g_{i,j} = k(y_i, y_j)$, is positive definite. As proposed in [13], we use Y to learn the MC-KUoS such that (i) each $\phi(y_i)$ can be well represented by one of the $\mathcal{S}_\ell$'s and (ii) all the $\mathcal{S}_\ell$'s are close to each other. Toward this end, we pose the problem of learning $\mathcal{U} = \bigcup_{\ell=1}^{L} \mathcal{S}_\ell$ in $\mathcal{H}$ as the following optimization problem:

$$\{\mathcal{S}_\ell\} = \underset{\{\mathcal{S}_\ell\}\subset\mathcal{G}_{\widetilde{m},s}}{\arg\min} \sum_{\substack{\ell,p=1\\ \ell\neq p}}^{L} d_u^2(\mathcal{S}_\ell, \mathcal{S}_p) + \lambda \sum_{i=1}^{N} \|\phi(y_i) - P_{\mathcal{S}_{l_i}}\phi(y_i)\|_2^2, \quad (2)$$

where $l_i = \arg\min_\ell \|\phi(y_i) - P_{\mathcal{S}_\ell}\phi(y_i)\|_2^2$ with $P_{\mathcal{S}_\ell}\phi(y_i)$ denoting the projection of $\phi(y_i)$ onto $\mathcal{S}_\ell$. In (2), the first term encourages the learned subspaces to be close to each other, while the second term ensures the learned subspaces will yield good approximations of ϕ-mapped features of training samples. The tuning parameter $\lambda > 0$ provides a compromise between the two terms. Our goal is to develop an efficient algorithm for solving (2) using the kernel trick [14], which avoids explicitly mapping Y to the feature space. Given a noisy signal y, we will also discuss an approach to computing its projection $\widehat{y}$ onto the learned MC-KUoS for signal denoising.

3. PROPOSED ALGORITHM

In this section, we present our approach for learning an MC-KUoS from the training data Y. In analogy with kernel PCA [9], we first calculate the kernel matrix $G = \phi(Y)^T\phi(Y)$. Then the centered kernel matrix $\widetilde{G}$, defined as $\widetilde{g}_{i,j} = \langle\widetilde{\phi}(y_i), \widetilde{\phi}(y_j)\rangle$ with $\widetilde{\phi}(y_i) = \phi(y_i) - \bar{\phi}$ and $\bar{\phi} = \frac{1}{N}\sum_{i=1}^N \phi(y_i)$, can be obtained from G by $\widetilde{G} = G - H_N G - G H_N + H_N G H_N$, where H_N is an $N \times N$ matrix with all elements $\frac{1}{N}$. Then for any $y, y' \in \mathbb{R}^m$, we have [16]

$$\begin{aligned}\widetilde{k}(y, y') &= \widetilde{\phi}(y)^T\widetilde{\phi}(y') \\ &= k(y, y') - \frac{1}{N}\mathbf{1}_N^T\boldsymbol{k}_y - \frac{1}{N}\mathbf{1}_N^T\boldsymbol{k}_{y'} + \frac{1}{N^2}\mathbf{1}_N^T G\mathbf{1}_N,\end{aligned}$$

where $\mathbf{1}_N = [1, 1, \ldots, 1]^T$ is an N-dimensional vector and $\boldsymbol{k}_y = [k(y, y_1), \ldots, k(y, y_N)]^T$. Next, to simplify the expression in (2), we define an $L \times N$ membership matrix W as

$$W \overset{def}{=} [w_{\ell,i} \in \{0,1\} : \sum_{\ell=1}^{L} w_{\ell,i} = 1, i = 1, 2, \ldots, N]. \quad (3)$$

Here, $w_{\ell,i} = 1$ if and only if $\widetilde{\phi}(y_i)$ is assigned to subspace $\mathcal{S}_\ell$. Let $D_\ell \in \mathbb{R}^{\widetilde{m}\times s}$ denote an orthonormal basis of $\mathcal{S}_\ell$ and $D = [D_1, \ldots, D_L]$, then for any $i = 1, \ldots, N$, we have the following

$$\begin{aligned}\|\phi(y_i) - P_{\mathcal{S}_\ell}\phi(y_i)\|_2^2 &= \|\widetilde{\phi}(y_i) - P_{\mathcal{S}_\ell}\widetilde{\phi}(y_i)\|_2^2 \\ &= \|\widetilde{\phi}(y_i)\|_2^2 - \|D_\ell^T\widetilde{\phi}(y_i)\|_2^2. \quad (4)\end{aligned}$$

Therefore, the optimization problem (2) can be written as $(D, W) = \arg\min_{D,W} F(D, W)$ where

$$\begin{aligned}F(D, W) = &\sum_{\substack{\ell,p=1\\ \ell\neq p}}^{L} \|D_\ell - P_{\mathcal{S}_p} D_\ell\|_F^2 \\ &+ \lambda \sum_{i=1}^{N}\sum_{\ell=1}^{L} w_{\ell,i}(\|\widetilde{\phi}(y_i)\|_2^2 - \|D_\ell^T\widetilde{\phi}(y_i)\|_2^2). \quad (5)\end{aligned}$$

Let $c_\ell = \{i \in \{1, \ldots, N\} : w_{\ell,i} = 1\}$ denote the indices of all $\widetilde{\phi}(y_i)$'s that are assigned to subspace $\mathcal{S}_\ell$ and define $Y_\ell = \big[y_i : i \in c_\ell\big]$ to be the corresponding $m \times N_\ell$ matrix with $N_\ell = |c_\ell|$. The centered data which are assigned to $\mathcal{S}_\ell$ are denoted by $\widetilde{\phi}(Y_\ell) = \big[\widetilde{\phi}(y_i) : i \in c_\ell\big]$. Note that since $\mathcal{S}_\ell$ is spanned by the columns of $\widetilde{\phi}(Y_\ell)$, we can write $D_\ell = \widetilde{\phi}(Y_\ell)U_\ell$, where $U_\ell \in \mathbb{R}^{N_\ell\times s}$ is some basis representation matrix to make D_ℓ orthonormal. We then have $U_\ell^T[\widetilde{G}]_{c_\ell,c_\ell}U_\ell = I_s$, where $[\widetilde{G}]_{c_\ell,c_\ell} = \widetilde{\phi}(Y_\ell)^T\widetilde{\phi}(Y_\ell)$ denotes the centered kernel matrix for subspace $\mathcal{S}_\ell$. In the following, all the computations involving D_ℓ's for MC-KUoS learning can be carried out by using c_ℓ's, U_ℓ's and the kernel trick, which greatly simplifies the computation. Now for any $i = 1, 2, \ldots, N$,

$$\|\widetilde{\phi}(y_i)\|_2^2 - \|D_\ell^T\widetilde{\phi}(y_i)\|_2^2 = \widetilde{k}(y_i, y_i) - \|U_\ell^T\widetilde{\phi}(Y_\ell)^T\widetilde{\phi}(y_i)\|_2^2 \quad (6)$$

where $\widetilde{k}(y_i, y_i) = k(y_i, y_i) - \frac{2}{N}\mathbf{1}_N^T\boldsymbol{k}_{y_i} + \frac{1}{N^2}\mathbf{1}_N^T G\mathbf{1}_N$. Let $\boldsymbol{\psi}_\ell(y_i) = [k(y_{c_{\ell(1)}}, y_i), k(y_{c_{\ell(2)}}, y_i), \ldots, k(y_{c_{\ell(N_\ell)}}, y_i)]^T$ denote an N_ℓ-dimensional vector with elements being inner products between $\phi(y_i)$ and the columns of $\phi(Y_\ell)$, where $\phi(Y_\ell) = \big[\phi(y_i) : i \in c_\ell\big]$. Then $\widetilde{\boldsymbol{\psi}}_\ell(y_i) \overset{def}{=} \widetilde{\phi}(Y_\ell)^T\widetilde{\phi}(y_i) = \boldsymbol{\psi}_\ell(y_i) - \frac{1}{N}\mathbf{1}_{N_\ell}\mathbf{1}_N^T\boldsymbol{k}_{y_i} - \frac{1}{N}[G]_{c_\ell,:}\mathbf{1}_N + \frac{1}{N^2}\mathbf{1}_{N_\ell}\mathbf{1}_N^T G\mathbf{1}_N$. Hence (6) can be written as $\|\widetilde{\phi}(y_i)\|_2^2 - \|D_\ell^T\widetilde{\phi}(y_i)\|_2^2 = \widetilde{k}(y_i, y_i) - \|U_\ell^T\widetilde{\boldsymbol{\psi}}_\ell(y_i)\|_2^2$. Next, after some algebraic manipulations, we obtain the following

$$\begin{aligned}&\|D_\ell - P_{\mathcal{S}_p} D_\ell\|_F^2 \\ &= s - \mathrm{tr}\big[(\widetilde{\phi}(Y_\ell)U_\ell)^T\widetilde{\phi}(Y_p)U_p(\widetilde{\phi}(Y_p)U_p)^T\widetilde{\phi}(Y_\ell)U_\ell\big] \\ &= s - \mathrm{tr}\big[U_\ell^T[\widetilde{G}]_{c_\ell,c_p}U_pU_p^T[\widetilde{G}]_{c_p,c_\ell}U_\ell\big], \quad (7)\end{aligned}$$

where $[\widetilde{G}]_{c_\ell,c_p} = \widetilde{\phi}(Y_\ell)^T\widetilde{\phi}(Y_p)$ denotes the centered inter-subspace kernel matrix between $\mathcal{S}_\ell$ and $\mathcal{S}_p$.

Instead of optimizing (5) simultaneously over (D, W), which will be computationally cumbersome, we will resort to minimizing F by alternating between minimizing $F(D, W)$ over W for a fixed D (the *kernel subspace assignment* step) and minimizing $F(D, W)$

Algorithm 1 Initialization for $\mathcal{S}_\ell$'s (KIOP)

Input: Centered kernel matrix $\widetilde{G}$, parameters L and s.
Initialize: $\mathcal{I}_N = \{1, \ldots, N\}$.

for $\ell = 1$ to L **do**
  $c_\ell \leftarrow$ randomly choose s elements in $\mathcal{I}_N$, $\mathcal{I}_N \leftarrow \mathcal{I}_N \setminus c_\ell$.
  Eigen decomposition of $[\widetilde{G}]_{c_\ell,c_\ell} = V_\ell \Sigma_\ell V_\ell^T$.
  $U_\ell \leftarrow V_\ell \Sigma_\ell^{-\frac{1}{2}}$.
end for

Output: Initial $\{c_\ell\}_{\ell=1}^L$ and $\{U_\ell \in \mathbb{R}^{s\times s}\}_{\ell=1}^L$.

over D for a fixed W (the *kernel subspace update* step). We start by initialization of the D_ℓ's. Since a subspace basis can be represented in the form of $D_\ell = \widetilde{\phi}(Y_\ell)U_\ell$ and we can compute U_ℓ explicitly by using $[\widetilde{G}]_{c_\ell,c_\ell}$, this step can be treated as the initialization of c_ℓ. Note that any s linearly independent vectors describe an s-dimensional subspace. In this regard, to initialize c_ℓ, or equivalently, Y_ℓ, we only need to choose s signals in the training set such that the ϕ-mapped "images" of these training samples are linearly independent. Based on the assumption that all $\phi(y_i)$'s are linearly independent, the initialization of c_ℓ can be done by randomly picking s indexes from $\{1, \ldots, N\}$ without replacement. We propose an initialization method in Algorithm 1, referred to as *kernel initial-orthogonalization procedure* (KIOP). Note that since $\bigcap_{\ell=1}^L c_\ell = \emptyset$ and we compute U_ℓ by $U_\ell = V_\ell \Sigma_\ell^{-\frac{1}{2}}$, it is trivial to verify $D_\ell^T D_\ell = I_s$ in this setting.

We now move onto the kernel subspace assignment stage. When D is fixed, kernel subspace assignment corresponds to solving $\forall i = 1, \ldots, N$, $w_{l_i,i} = 1$ if

$$l_i = \underset{\ell=1,\ldots,L}{\arg\min}\ \widetilde{k}(y_i, y_i) - \|U_\ell^T \widetilde{\boldsymbol{\psi}}_\ell(y_i)\|_2^2. \tag{8}$$

Then for the subspace update stage, since W is fixed, all the c_ℓ's and Y_ℓ's are fixed. By fixing those variables, we can write the reduced problem of (5) as a function of U_ℓ's as follows:

$$\begin{aligned}\min_{U_1,\ldots,U_L} \quad & f(U_1, \ldots, U_L) = \sum_{\substack{\ell,p=1\\ \ell\neq p}}^{L} \|\widetilde{\phi}(Y_\ell)U_\ell - P_{\mathcal{S}_p}(\widetilde{\phi}(Y_\ell)U_\ell)\|_F^2 \\ & + \lambda \sum_{\ell=1}^{L} \left(\|\widetilde{\phi}(Y_\ell)\|_F^2 - \|U_\ell^T \widetilde{\phi}(Y_\ell)^T \widetilde{\phi}(Y_\ell)\|_F^2\right) \\ \text{s.t.} \quad & U_\ell^T [\widetilde{G}]_{c_\ell,c_\ell} U_\ell = I_s, \ell = 1, 2, \ldots, L. \end{aligned} \tag{9}$$

Instead of updating all the U_ℓ's simultaneously, which is again a difficult problem, we use block coordinate descent method [17] to minimize f and update U_ℓ's sequentially. Before that, we first need to initialize all the U_ℓ's such that $U_\ell \in \mathbb{R}^{N_\ell \times s}$ and $U_\ell^T [\widetilde{G}]_{c_\ell,c_\ell} U_\ell = I_s$. To do so, we again apply eigen decomposition of $[\widetilde{G}]_{c_\ell,c_\ell} = V_\ell \Sigma_\ell V_\ell^T$ and define $\mathcal{I}_s = \{1, \ldots, s\}$, which then results in $U_\ell = [V_\ell]_{:,\mathcal{I}_s} [\Sigma_\ell]_{\mathcal{I}_s,\mathcal{I}_s}^{-\frac{1}{2}}$. After the bases initialization step, we update U_ℓ sequentially and each subproblem of (9) reduces to

$$\begin{aligned} U_\ell = & \underset{Q^T[\widetilde{G}]_{c_\ell,c_\ell}Q = I_s}{\arg\min} \sum_{p\neq\ell} \|\widetilde{\phi}(Y_\ell)Q - P_{\mathcal{S}_p}(\widetilde{\phi}(Y_\ell)Q)\|_F^2 \\ & + \frac{\lambda}{2}(\|\widetilde{\phi}(Y_\ell)\|_F^2 - \|Q^T \widetilde{\phi}(Y_\ell)^T \widetilde{\phi}(Y_\ell)\|_F^2) \\ = & \underset{Q^T[\widetilde{G}]_{c_\ell,c_\ell}Q = I_s}{\arg\max} \operatorname{tr}(Q^T A_\ell Q), \end{aligned}$$

Algorithm 2 Metric-Constrained Kernel UoS Learning

Input: Training data Y, parameters L, s and λ, kernel function k.

Compute kernel matrix G such that $g_{i,j} = k(y_i, y_j)$.
$\widetilde{G} \leftarrow G - H_N G - G H_N + H_N G H_N$.
Initialize $\{U_\ell\}_{\ell=1}^L$ and $\{c_\ell\}_{\ell=1}^L$ by KIOP (Algorithm 1).
while stopping rule **do**
  for $i = 1$ to N (*Kernel Subspace Assignment*) **do**
    $l_i \leftarrow \arg\min_{\ell=1,\ldots,L} \widetilde{k}(y_i, y_i) - \|U_\ell^T \widetilde{\boldsymbol{\psi}}_\ell(y_i)\|_2^2$.
    $w_{l_i,i} \leftarrow 1$, and $\forall \ell \neq l_i, w_{\ell,i} \leftarrow 0$.
  end for
  for $\ell = 1$ to L (*Kernel Bases Initialization*) **do**
    $c_\ell \leftarrow \{1 \leq i \leq N : w_{\ell,i} = 1\}$, $N_\ell \leftarrow |c_\ell|$.
    Eigen decomposition of $[\widetilde{G}]_{c_\ell,c_\ell} = V_\ell \Sigma_\ell V_\ell^T$, with the diagonal elements of Σ_ℓ in nonincreasing order.
    $U_\ell \leftarrow [V_\ell]_{:,\mathcal{I}_s} [\Sigma_\ell]_{\mathcal{I}_s,\mathcal{I}_s}^{-\frac{1}{2}}$.
  end for
  while stopping rule **do**
    for $\ell = 1$ to L (*Kernel Subspace Update*) **do**
      $A_\ell \leftarrow \sum_{p\neq\ell} [\widetilde{G}]_{c_\ell,c_p} U_p U_p^T [\widetilde{G}]_{c_p,c_\ell} + \frac{\lambda}{2}[\widetilde{G}]_{c_\ell,c_\ell}^2$.
      $U_\ell \leftarrow$ eigenvectors corresponding to the largest s eigenvalues for the generalized problem $A_\ell b = \alpha[\widetilde{G}]_{c_\ell,c_\ell} b$ such that $U_\ell^T [\widetilde{G}]_{c_\ell,c_\ell} U_\ell = I_s$.
    end for
  end while
end while

Output: $\{N_\ell \in \mathbb{N}\}_{\ell=1}^L$, $\{c_\ell\}_{\ell=1}^L$ and $\{U_\ell \in \mathbb{R}^{N_\ell \times s}\}_{\ell=1}^L$.

where $A_\ell = \sum_{p\neq\ell} [\widetilde{G}]_{c_\ell,c_p} U_p U_p^T [\widetilde{G}]_{c_p,c_\ell} + \frac{\lambda}{2}[\widetilde{G}]_{c_\ell,c_\ell}^2$ is a symmetric matrix of dimension $N_\ell \times N_\ell$. Since $[\widetilde{G}]_{c_\ell,c_\ell}$ is a positive definite matrix, it follows from [18] that the trace of $U_\ell^T A_\ell U_\ell$ is maximized when U_ℓ is a set of eigenvectors associated with the largest s eigenvalues for the generalized problem $A_\ell b = \alpha[\widetilde{G}]_{c_\ell,c_\ell} b$ with $U_\ell^T [\widetilde{G}]_{c_\ell,c_\ell} U_\ell = I_s$. The whole algorithm can be detailed in Algorithm 2, termed as *Metric-Constrained Kernel Union-of-Subspaces Learning* (MC-KUSaL).

4. PRE-IMAGE RECONSTRUCTION

So far, we have only discussed an algorithm for learning an MC-KUoS using the kernel trick. Now suppose we have a noisy test sample $y = x + \text{noise} \in \mathbb{R}^m$ where $\widetilde{\phi}(x)$ is assumed to belong to one of the subspaces $\mathcal{S}_\tau$ in $\mathcal{U}$ with $\widetilde{\phi}(x) = \phi(x) - \bar{\phi}$. In order to denoise this test sample and interpret/visualize the denoised signal, we need to find a pre-image of y, denoted by $\widehat{y} \in \mathbb{R}^m$, such that $\phi(\widehat{y}) = P_{\mathcal{S}_\tau}\phi(y)$ for some $\tau \in \{1, \ldots, L\}$. First of all, the solution for τ is trivial since $\tau = \arg\min_\ell \|\phi(y) - P_{\mathcal{S}_\ell}\phi(y)\|_2^2$, which can be done by the subspace assignment described in (8). Then $P_{\mathcal{S}_\tau}\phi(y)$ is given by $P_{\mathcal{S}_\tau}\phi(y) = D_\tau D_\tau^T \widetilde{\phi}(y) + \bar{\phi}$ with $\widetilde{\phi}(y) = \phi(y) - \bar{\phi}$. However, as noted in [11], the pre-image does not always exist and the authors in [11] reformulated this problem by minimizing the squared distance between the feature point $\phi(\widehat{y})$ and $P_{\mathcal{S}_\tau}\phi(y)$, i.e.,

$$\min_{\widehat{y}\in\mathbb{R}^m} \|\phi(\widehat{y}) - P_{\mathcal{S}_\tau}\phi(y)\|_2^2 = \|\phi(\widehat{y})\|_2^2 - 2(P_{\mathcal{S}_\tau}\phi(y))^T \phi(\widehat{y}) + \Upsilon$$

where Υ includes terms independent of $\widehat{y}$. We carry out this pre-image computation by leveraging the idea in [16, 19] and only using the feature-space distances to find an appropriate pre-image. To this

end, we first introduce the notion of the squared feature distance between $P_{\mathcal{S}_\tau}\phi(y)$ and any $\phi(y_i)$ under the MC-KUoS model, defined as follows [16]

$$d^2_{\mathcal{H}}(\phi(y_i), P_{\mathcal{S}_\tau}\phi(y)) = \|P_{\mathcal{S}_\tau}\phi(y)\|_2^2 + \|\phi(y_i)\|_2^2 - 2(P_{\mathcal{S}_\tau}\phi(y))^T\phi(y_i), \quad (10)$$

where $\|P_{\mathcal{S}_\tau}\phi(y)\|_2^2$ and $(P_{\mathcal{S}_\tau}\phi(y))^T\phi(y_i)$ can be calculated in terms of kernel representation by $\|P_{\mathcal{S}_\tau}\phi(y)\|_2^2 = \widetilde{\boldsymbol{\psi}}_\tau(y)^T U_\tau U_\tau^T \left(\widetilde{\boldsymbol{\psi}}_\tau(y) + \frac{2}{N}[G]_{c_\tau,:}\mathbf{1}_N - \frac{2}{N^2}\mathbf{1}_{N_\tau}\mathbf{1}_N^T G\mathbf{1}_N\right) + \frac{1}{N^2}\mathbf{1}_N^T G\mathbf{1}_N$ and $(P_{\mathcal{S}_\tau}\phi(y))^T\phi(y_i) = \widetilde{\boldsymbol{\psi}}_\tau(y)^T U_\tau U_\tau^T\left(\boldsymbol{\psi}_\tau(y_i) - \frac{1}{N}\mathbf{1}_{N_\tau}\mathbf{1}_N^T\boldsymbol{k}_{y_i}\right) + \frac{1}{N}\mathbf{1}_N^T\boldsymbol{k}_{y_i}$. Therefore, (10) becomes

$$\begin{aligned} &d^2_{\mathcal{H}}(\phi(y_i), P_{\mathcal{S}_\tau}\phi(y)) \\ &= \widetilde{\boldsymbol{\psi}}_\tau(y)^T U_\tau U_\tau^T\Big(\widetilde{\boldsymbol{\psi}}_\tau(y) - \frac{2}{N}\mathbf{1}_{N_\tau}\mathbf{1}_N^T(\frac{1}{N}G\mathbf{1}_N - \boldsymbol{k}_{y_i}) - 2\boldsymbol{\psi}_\tau(y_i) \\ &\quad + \frac{2}{N}[G]_{c_\tau,:}\mathbf{1}_N\Big) + g_{i,i} + \frac{1}{N^2}\mathbf{1}_N^T G\mathbf{1}_N - \frac{2}{N}\mathbf{1}_N^T\boldsymbol{k}_{y_i} \end{aligned} \quad (11)$$

with $g_{i,i} = k(y_i, y_i)$.

Let us now first consider the solution of $\widehat{y}$ for the Gaussian kernel $k(y, y') = \exp(-\|y - y'\|_2^2/c)$ with $c > 0$. In this case the problem is equivalent to maximizing $\rho(\widehat{y}) = (P_{\mathcal{S}_\tau}\phi(y))^T\phi(\widehat{y})$ [11]. To do so, we express $\rho(\widehat{y})$ by

$$\begin{aligned} \rho(\widehat{y}) &= (D_\tau D_\tau^T\widetilde{\phi}(y) + \bar{\phi})^T\phi(\widehat{y}) \\ &= \widetilde{\boldsymbol{\psi}}_\tau(y)^T U_\tau U_\tau^T(\boldsymbol{\psi}_\tau(\widehat{y}) - \frac{1}{N}\mathbf{1}_{N_\tau}\mathbf{1}_N^T\boldsymbol{k}_{\widehat{y}}) + \frac{1}{N}\mathbf{1}_N^T\boldsymbol{k}_{\widehat{y}}. \end{aligned}$$

Next, we define $\gamma = \frac{1}{N}(1 - \widetilde{\boldsymbol{\psi}}_\tau(y)^T U_\tau U_\tau^T\mathbf{1}_{N_\tau})\mathbf{1}_N \in \mathbb{R}^N$ and let $\widehat{\gamma}$ be an N-dimensional vector such that $[\widehat{\gamma}]_{c_\tau} = [\gamma]_{c_\tau} + U_\tau U_\tau^T\widetilde{\boldsymbol{\psi}}_\tau(y)$ and $[\widehat{\gamma}]_{\mathcal{I}_N\setminus c_\tau} = [\gamma]_{\mathcal{I}_N\setminus c_\tau}$, then $\rho(\widehat{y}) = \widehat{\gamma}^T\boldsymbol{k}_{\widehat{y}} = \sum_{i=1}^N\widehat{\gamma}(i)k(\widehat{y}, y_i)$. The extremum can be obtained by setting $\nabla_{\widehat{y}}\rho(\widehat{y}) = 0$ and it follows that

$$\widehat{y} = \frac{\sum_{i=1}^N\widehat{\gamma}(i)\exp(-\|\widehat{y} - y_i\|_2^2/c)y_i}{\sum_{i=1}^N\widehat{\gamma}(i)\exp(-\|\widehat{y} - y_i\|_2^2/c)}. \quad (12)$$

By using the approximation $P_{\mathcal{S}_\tau}\phi(y) \approx \phi(\widehat{y})$ and the relation $\|\widehat{y} - y_i\|_2^2 = -c\log(\frac{1}{2}(2 - d^2_{\mathcal{H}}(\phi(\widehat{y}), \phi(y_i))))$ [16], we can finally reconstruct the pre-image as follows:

$$\widehat{y} = \frac{\sum_{i=1}^N\widehat{\gamma}(i)\Big(\frac{1}{2}\big(2 - d^2_{\mathcal{H}}(P_{\mathcal{S}_\tau}\phi(y), \phi(y_i))\big)\Big)y_i}{\sum_{i=1}^N\widehat{\gamma}(i)\Big(\frac{1}{2}\big(2 - d^2_{\mathcal{H}}(P_{\mathcal{S}_\tau}\phi(y), \phi(y_i))\big)\Big)}. \quad (13)$$

Next, for the polynomial kernel $k(y, y') = (\langle y, y'\rangle + c)^d$ with $c \geq 0$ and an odd degree d, we can follow a similar procedure and have the following expression to provide an approximate solution for the problem of pre-image computation:

$$\widehat{y} = \sum_{i=1}^N\widehat{\gamma}(i)\Big(\frac{(P_{\mathcal{S}_\tau}\phi(y))^T\phi(y_i)}{\|P_{\mathcal{S}_\tau}\phi(y)\|_2^2}\Big)^{\frac{d-1}{d}}y_i. \quad (14)$$

5. NUMERICAL RESULTS

In this section, we present some preliminary denoising results on the USPS dataset, which consists of a collection of $m = 256$-dimensional handwritten digits. In our experiments, we learn a union of L subspaces in the kernel space from the noiseless training data, followed by denoising of noisy test samples using learned subspaces. We assume that every noisy *test* sample $y^{te} = x^{te} + \xi$ where $\phi(x^{te})$ belongs to one of the $\mathcal{S}_\ell$'s in $\mathcal{H}$ (with $\|x^{te}\|_2^2 = 1$) and ξ is of $\mathcal{N}(0, (\sigma_{te}^2/m)\boldsymbol{I}_m)$ distribution. We add white Gaussian noise with different expected noise power ($\mathbb{E}[\|\xi\|_2^2] = \sigma_{te}^2$) ranging from 0.2 to 0.5 to the noiseless test set. We use X^{te} and Y^{te} to denote the set of "clean" and noisy test signals respectively. The results of our proposed approach are compared with three other methods: (i) kernel k-means clustering followed by kernel PCA on each cluster (kernel k-means) [14], (ii) kernel PCA [9] with the same number of eigenvectors as in MC-KUSaL (KPCA-FIX) and (iii) kernel PCA with the number of eigenvectors chosen in an oracle fashion by $s = \arg\min_s ||P_{\mathcal{S}}\phi(y^{te}) - \phi(x^{te})||_2^2$ (KPCA-Oracle), where x^{te} and y^{te} are clean and noisy test samples, respectively. The relative reconstruction error of $x_i^{te} \in X^{te}$ is then calculated by $\|x_i^{te} - \widehat{y}_i^{te}\|_2^2/\|x_i^{te}\|_2^2$, where $\widehat{y}_i^{te}$ denotes the pre-image of y_i^{te}. We use $\lambda = 1$ for all experiments and report the mean of relative reconstruction errors of X^{te}.

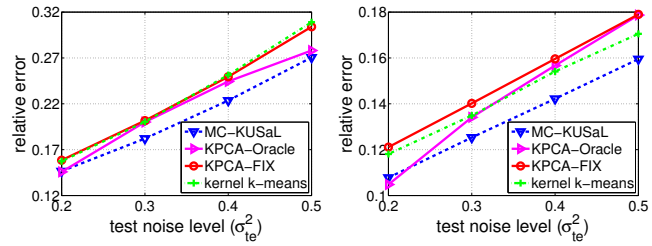

Fig. 1. Denoising result on USPS dataset for (a) $k(y, y') = \exp(-\|y - y'\|_2^2/4)$ and (b) $k(y, y') = (\langle y, y'\rangle + 2)^3$. Note that the KPCA-Oracle algorithm is the ideal setting of the kernel PCA.

We first experiment with a Gaussian kernel with parameters $c = 4$, $L = 2$ and $s = 70$. In this experiment, we select the first 200 samples from digits "0" and "8" in the dataset (400 images in total). All these 400 samples are then vectorized and normalized to unit ℓ_2 norms. Within these samples, we randomly choose 150 samples (without replacement) from each class for training and the remaining 50 samples for testing, forming $Y \in \mathbb{R}^{256\times 300}$ and $X^{te} \in \mathbb{R}^{256\times 100}$. Fig. 1(a) shows the relative error of test data for different methods. It can be inferred from this figure that our method produces better results than other methods for almost all σ_{te}'s (the only exception is when $\sigma_{te}^2 = 0.2$, in which case MC-KUSaL's results are comparable to those of KPCA-Oracle).

Finally, we choose the last 200 samples of digits "0" and "1" in the polynomial kernel experiments and generate data with the same procedure as in the previous experiments. The parameters are $c = 2$, $d = 3$, $L = 2$ and $s = 40$. The relative error of test data are shown in Fig. 1(b) and we can see the proposed method again yields better denoising performance than all the other approaches when the signal-to-noise ratio (SNR) of test data is relatively low.

6. CONCLUSION AND FUTURE WORK

In this paper, we introduced a framework for learning of a collection of nonlinear manifolds based on the MC-KUoS model. Experimental results validate the effectiveness of both the MC-KUoS model and our iterative method for learning an MC-KUoS in the application of denoising task. One of the interesting avenues of future work is the detection of the number and dimensions of the subspaces in the kernel space from the training data.

7. REFERENCES

[1] Z. Ghahramani and G. E. Hinton, "The EM algorithm for mixtures of factor analyzers," Tech. Rep., CRG-TR-96-1, University of Toronto, 1997.

[2] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[3] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[4] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[5] N. D. Lawrence, "Spectral dimensionality reduction via maximum entropy," in *Proc. Intl. Conf. Artificial Intelligence and Statistics (AISTATS)*, 2011, pp. 51–59.

[6] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *Int. J. Comput. Vis.*, vol. 100, no. 3, pp. 217–240, 2012.

[7] T. Ahmed and W. U. Bajwa, "A greedy, adaptive approach to learning geometry of nonlinear manifolds," in *Proc. IEEE Workshop Statistical Signal Processing (SSP)*, 2014, pp. 133–136.

[8] T. Ahmed and W. U. Bajwa, "Geometric manifold approximation using union of tangent patches," in *Proc. 2nd IEEE Global Conf. Signal and Information Processing (GlobalSIP)*, 2014.

[9] B. Schölkopf, A. J. Smola, and K.-R. Müller, "Kernel principal component analysis," *Advances in kernel methods: support vector learning*, pp. 327–352, 1999.

[10] J. Ham, D. D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds," in *Proc. Intl. Conf. Machine Learning (ICML)*, 2004, pp. 47–54.

[11] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces," in *Advances in Neural Information Processing Systems (NIPS)*, 1999, pp. 536–542.

[12] E. Kokiopoulou and Y. Saad, "Orthogonal neighborhood preserving projections: A projection-based dimensionality reduction technique," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2143–2156, 2007.

[13] T. Wu and W. U. Bajwa, "Revisiting robustness of the union-of-subspaces model for data-adaptive learning of nonlinear signal models," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, 2014, pp. 3390–3394.

[14] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10, no. 5, pp. 1299–1319, 1998.

[15] L. Wang, X. Wang, and J. Feng, "Subspace distance analysis with application to adaptive Bayesian algorithm for face recognition," *Pattern Recognition*, vol. 39, no. 3, pp. 456–464, 2006.

[16] J. T.-Y. Kwok and I. W.-H. Tsang, "The pre-image problem in kernel methods," *IEEE Trans. Neural Netw.*, vol. 15, no. 6, pp. 1517–1525, 2004.

[17] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 2nd edition, 1999.

[18] E. Kokiopoulou, J. Chen, and Y. Saad, "Trace optimization and eigenproblems in dimension reduction methods," *Numer. Linear Algebra Appl.*, vol. 18, no. 3, pp. 565–602, 2011.

[19] Y. Rathi, S. Dambreville, and A. Tannenbaum, "Statistical shape analysis using kernel PCA," in *Proc. SPIE*, 2006, vol. 6064, pp. 425–432.