Dimensionality Reduction using Covariance Operator Inverse Regression

Minyoung Kim and Vladimir Pavlovic
Department of Computer Science
Rutgers University, NJ 08854
{mikim,vladimir}@cs.rutgers.edu
http://seqam.rutgers.edu

Abstract

We consider the task of dimensionality reduction for regression (DRR) whose goal is to find a low dimensional representation of input covariates, while preserving the statistical correlation with output targets. DRR is particularly suited for visualization of high dimensional data as well as the efficient regressor design with a reduced input dimension. In this paper we propose a novel nonlinear method for DRR that exploits the kernel Gram matrices of input and output. While most existing DRR techniques rely on the inverse regression, our approach removes the need for explicit slicing of the output space using covariance operators in RKHS. This unique property make DRR applicable to problem domains with high dimensional output data with potentially significant amounts of noise. Although recent kernel dimensionality reduction algorithms make use of RKHS covariance operators to quantify conditional dependency between the input and the targets via the dimension-reduced input, they are either limited to a transduction setting or linear input subspaces and restricted to non-closed-form solutions. In contrast, our approach provides a closed-form solution to the nonlinear basis functions on which any new input point can be easily projected. We demonstrate the benefits of the proposed method in a comprehensive set of evaluations on several important regression problems that arise in computer vision.

1. Introduction

The task of *dimensionality reduction for regression* (DRR) is to find a low dimensional representation, $\mathbf{z} \in \mathbb{R}^q$, of the input covariates, $\mathbf{x} \in \mathbb{R}^p$, with $q \ll p$, for regressing the output, $\mathbf{y} \in \mathbb{R}^d$, given n i.i.d. data $\{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^n$. DRR has found many applications in visualization of high dimensional data, efficient regressor design with a reduced input dimension, and elimination of noise in data $\mathbf{x}$ by uncovering the essential information $\mathbf{z}$ for predicting $\mathbf{y}$. In all these tasks DRR is not tied to a particular regression estimation method, but can be rather seen as a prior task to the regressor design for a better understanding of data.

DRR differs from other well-known dimensionality reduction algorithms in several ways. One can view DRR as a *supervised* learning technique with *real multivariate* labels $\mathbf{y}$. Most other supervised techniques focus on the classification setting (*i.e.*, discrete $\mathbf{y}$), including Linear Discriminant Analysis (LDA), kernel LDA, general graph embedding [21], and metric learning [5, 17, 20]. Unsupervised dimension reduction methods, on the other hand, assume that $\mathbf{y}$ is unknown. Principal subspace methods (PCA and kernel PCA [14]), nonlinear locality-preserving manifold learning (LLE [11], ISOMAP [15], and Laplacian Eigenmap [2]), and probabilistic methods like GPLVM [6] belong to this class of approaches that do not leverage known target values. DRR has been a focus of several important lines of research in the statistical machine learning community ([3, 4, 9, 10]). However, it has received significantly less attention in the domain of computer vision.

The crucial notion related to DRR is that of *sufficiency in dimension reduction* (SDR, [3, 4, 9]). SDR states that one has to find the linear subspace bases $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_q]$ with $\mathbf{b}_l \in \mathbb{R}^p$, (or basis functions in the nonlinear case, $\mathbf{B} = \{\mathbf{b}_1(\cdot), \ldots, \mathbf{b}_q(\cdot)\}$) such that $\mathbf{y}$ and $\mathbf{x}$ are conditionally independent given $\mathbf{B}^\top \mathbf{x}$. As this condition implies that the conditional distribution of $\mathbf{y}$ given $\mathbf{x}$ equals that of $\mathbf{y}$ given $\mathbf{z} = \mathbf{B}^\top \mathbf{x}$, the dimension reduction entails no loss of information for the purpose of regression. It is known that such $\mathbf{B}$ always exists (at least the identity $\mathbf{B} = \mathbf{I}$ for $q = p$) with non-unique solutions[1]. Hence, one is naturally interested in the minimal subspace or the intersection of all such subspaces, often called the *central subspace* [2].

Two schools of approaches have been suggested to find the central subspace: the inverse regression (IR) [9, 19] and

[1] Any set of bases that spans the subspace of $\mathbf{B}$ will be a solution.

[2] Although the term *subspace* is usually meant for a linear case, however, we abuse the term for both linear and nonlinear cases throughout the paper.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

the kernel dimension reduction (KDR) [4, 10]. KDR [4] directly reduces the task of imposing conditional independence to the optimization problem that minimizes the conditional covariance operator[3] in a RKHS (reproducing kernel Hilbert space). This is achieved by quantifying the notion of conditional dependency (between $\mathbf{y}$ and $\mathbf{x}$ given $\mathbf{B}^\top\mathbf{x}$) using a positive definite ordering of the expected covariance operators in what is called the probability-determining RKHS (*e.g.*, the RBF kernel-induced Hilbert space).

Although KDR formulates the problem in RKHS, the final projection is linear in the original space. For a nonlinear extension, [10] proposed the manifold KDR which first maps the original input space to a nonlinear manifold (*e.g.*, by Laplacian Eigenmap learned from $\mathbf{x}$ only), and applies the KDR to find a linear subspace in the manifold. However, this introduces a tight coupling between the central subspace and the separately learned input manifold, restricting the approach to a transduction setting. That is, for a new input point, one has to rebuild the manifold entirely with data including the new point[4]. Moreover, neither of the methods has a closed-form solution and resorts to a gradient-based optimization.

The inverse regression (IR) is another interesting framework for DRR. IR is based on the fact that the inverse regression, $\mathbb{E}[\mathbf{x}|\mathbf{y}]$, lies on the subspace spanned by $\mathbf{B}$ (the bases of the central subspace), provided that the marginal distribution of $\mathbf{x}$ is ellipse-symmetric (*e.g.*, a Gaussian). Thus $\mathbf{B}$ coincides with the principal directions in the variance of the inverse regression, namely, $\mathbb{V}(\mathbb{E}[\mathbf{x}|\mathbf{y}])$. In [9], this variance was estimated by slicing the output space (*i.e.*, clustering $\mathbf{y}$), lending the name sliced IR (or SIR).

Despite its simplicity and a closed-form solution, SIR assumes a *linear* central subspace, with a strong restriction on the marginal distribution of $\mathbf{x}$. To cope with the limitation, a natural kernel extension (KSIR) was proposed in [19]. KSIR discovers a nonlinear central subspace and allows few restrictions on the class of distribution on $\mathbf{x}$, for example, admitting a nonparametric kernel density. However, KSIR still resorts to slicing of $\mathbf{y}$, which can result in unreliable variance estimates for high dimensional $\mathbf{y}$.

In this paper we propose a novel nonlinear method for DRR that exploits the covariance functions of input as well as the output. We estimate the variance of the inverse regression under the IR framework but avoid explicit slicing by an effective use of covariance operators in RKHS. This leads to a general solution with KSIR as its special case. Our approach can be reliably applied to the cases of high dimensional output, while suppressing potential noise in the output data.

The main contributions of this work address important limitations of existing DRR techniques. In particular, our approach provides the following benefits: (1) a closed-form solution, (2) a nonlinear central subspace, (3) mild assumption on the input distribution, (4) reliable estimation for high dimensional output, (5) robustness to noise, and (6) ease of generalization to new input points.

The paper is organized as follows: We briefly review the inverse regression framework and slicing-based techniques (SIR and KSIR) in Sec. 2. Sec. 3 introduces our approach. In Sec. 4 the benefits of the proposed method are demonstrated in a comprehensive set of evaluations on several regression problems. We conclude the paper with Sec. 5.

2. Background

Throughout the paper, we assume that the data pair ($\mathbf{x} \in \mathbb{R}^p, \mathbf{y} \in \mathbb{R}^d$) is drawn from an unknown joint distribution $P(\mathbf{x}, \mathbf{y})$, where all the expectations and (co)variances that appear in the paper are taken w.r.t. $P(\mathbf{x}, \mathbf{y})$.

2.1. Sliced Inverse Regression (SIR)

The following theorem plays a crucial role in the IR framework. See [9] for the proof. Without loss of generality, we assume that $\mathbf{x}$ is centered, *i.e.*, $\mathbb{E}[\mathbf{x}] = 0$.

Theorem 1. *If (a) there exists a q-dimensional central subspace with bases $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_q]$,* i.e., *$\mathbf{y} \perp \mathbf{x} | \mathbf{B}^\top\mathbf{x}$, and (b) for any $\mathbf{a} \in \mathbb{R}^p$, $\mathbb{E}[\mathbf{a}^\top\mathbf{x}|\mathbf{B}^\top\mathbf{x}]$ is linear in $\{\mathbf{b}_l^\top\mathbf{x}\}_{l=1}^q$ [5], then $\mathbb{E}[\mathbf{x}|\mathbf{y}] \in \mathbb{R}^p$ (traced by $\mathbf{y}$) lies on the subspace spanned by $\{\mathbf{\Sigma_{xx}}\mathbf{b}_l\}_{l=1}^q$, where $\mathbf{\Sigma_{xx}}$ is the covariance of $\mathbf{x}$.*

According to Thm. 1, $\mathbf{B}$ can be obtained from the q principal directions of $\mathbb{E}[\mathbf{x}|\mathbf{y}]$; the column vectors of $\mathbf{B}$ are the q largest eigenvectors of $\mathbb{V}(\mathbb{E}[\mathbf{x}|\mathbf{y}])$, pre-multiplied by $\mathbf{\Sigma}_{\mathbf{xx}}^{-1}$. Given the data $\{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^n$, SIR of [9] suggests to slice (or cluster) $\mathbf{y}$ so as to compute the sample estimate of $\mathbb{V}(\mathbb{E}[\mathbf{x}|\mathbf{y}])$. More specifically, after clustering $\{\mathbf{y}^i\}_{i=1}^n$ into J slices, $S_1, \ldots, S_J$, and computing the slice means, $\mathbf{m}_j = \frac{1}{|S_j|}\sum_{i \in S_j}\mathbf{x}^i$, to approximate $\mathbb{E}[\mathbf{x}|\mathbf{y} \in S_j]$, the sample estimate is, $\mathbf{V} = \sum_j p_j \mathbf{m}_j \mathbf{m}_j^\top$, where $p_j = |S_j|/n$ is the j-th slice proportion.

SIR finds the directions of maximum variance, with n data points collapsed into J slice means using the affinity in $\mathbf{y}$. Not surprisingly, for the extreme case of $J = n$, when each slice identifies with a single data point, $\mathbf{V}$ becomes the sample covariance of $\mathbf{x}$, gives rise to the PCA. However, for $J < n$, the $\mathbf{y}$ labels have an effect on *suppressing the variance* of directions within the same slice, which is a desirable strategy for the purpose of regression.

It is known that the condition (b) in Thm. 1 equivalently imposes an elliptically-symmetric distribution (*e.g.*, a Gaussian) of $\mathbf{x}$. Hence, SIR relies on two assumptions: the

[3] Refer to Sec. 3.1 for the definition.

[4] One may estimate the manifold image of the new point by extra/interpolation. However, this requires additional estimation effort.

[5] $\exists\{\alpha_l\}_{l=0}^q$ s.t. $\mathbb{E}[\mathbf{a}^\top\mathbf{x}|\mathbf{b}_1^\top\mathbf{x}, \ldots, \mathbf{b}_q^\top\mathbf{x}] = \alpha_0 + \sum_{l=1}^q \alpha_l \cdot \mathbf{b}_l^\top\mathbf{x}$.

linearity of the central subspace and the elliptical symmetry of the marginal distribution of $\mathbf{x}$. These assumptions can be strong in certain situations, leading to failure of the SIR if the conditions are not met. In what follows, one can consider a rather natural nonlinear extension via the RKHS mapping $\mathbf{x} \rightarrow \Phi(\mathbf{x})$, which helps relax the strong constraints of SIR.

2.2. Kernel extension of Inverse Regression

In the kernel extension of SIR, $\mathbf{x}$ is mapped to $\Phi(\mathbf{x}) \in \mathcal{H}_k$, where $\mathcal{H}_k$ is the Hilbert space induced from the kernel function $k(\cdot,\cdot)$ defined on the $\mathbf{x}$ space. We assume that $\Phi(\mathbf{x})$ is centered[6] in $\mathcal{H}_k$. The kernel extension consequently results in: (1) $\mathbf{B}$ has nonlinear basis functions $\mathbf{b}_l(\cdot) \in \mathcal{H}_k$, $l = 1,\ldots,q$, (2) $\mathbb{E}[\Phi(\mathbf{x})|\mathbf{y}]$ lies on a nonlinear function space spanned by $\{\mathbf{\Sigma_{xx}}\mathbf{b}_l\}_{l=1}^q$ [7], and (3) we estimate the operator, $\mathbb{V}(\mathbb{E}[\Phi(\mathbf{x})|\mathbf{y}])$, and its major eigenfunctions.

Similarly to SIR, KSIR of [19] estimates $\mathbb{V}(\mathbb{E}[\Phi(\mathbf{x})|\mathbf{y}])$ by slicing the output, leading to the following algorithm:

1. Cluster $\{\mathbf{y}^i\}_{i=1}^n$ into J slices: $S_1,\ldots,S_J$. Compute cluster means, $\mathbf{m}_j = \frac{1}{|S_j|}\sum_{i\in S_j}\Phi(\mathbf{x}^i)$ for $j = 1,\ldots,J$. p_j $(= |S_j|/n)$ is the j-th cluster proportion.
2. Estimate the sample covariance of the slice-wise inverse regression, *i.e.*, $\mathbf{V} = \sum_{j=1}^J p_j \mathbf{m}_j \mathbf{m}_j^\top$. Its q major eigenfunctions are denoted as $\{\mathbf{v}_l\}_{l=1}^q$.
3. The central subspace directions are obtained as $\mathbf{b}_l = \mathbf{\Sigma_{xx}^{-1}}\mathbf{v}_l$ for $l = 1,\ldots,q$.

Even though the above computations can be accomplished in the original input space (*i.e.*, SIR), they cannot be represented explicitly in the RKHS. For instance, $\mathbf{m}_j$ is a *function* and $\mathbf{V}$ is an *operator*. However, using the representer theorem [12], the eigenfunctions $\mathbf{v}$ and the central subspace directions $\mathbf{b}$ can be obtained in dual forms. The trick is similar to that of kernel PCA [14].

In Step-2, to solve the eigensystem $\mathbf{V}\cdot\mathbf{v} = \lambda\cdot\mathbf{v}$, we represent $\mathbf{v}$ as a linear combination of $\{\Phi(\mathbf{x}^i)\}_{i=1}^n$, *i.e.*, $\mathbf{v} = \sum_{i=1}^n \alpha_i \Phi(\mathbf{x}^i)$. Pre-multiplying by $\Phi(\mathbf{x}^r)^\top$ (for $r = 1,\ldots,n$) yields the LHS of the eigensystem as:

$$\sum_{j=1}^J p_j \big(\mathbf{m}_j^\top \Phi(\mathbf{x}^r)\big)\cdot\Big(\sum_{i=1}^n \alpha_i \big(\mathbf{m}_j^\top\Phi(\mathbf{x}^i)\big)\Big). \qquad (1)$$

Since $\mathbf{m}_j = \frac{1}{|S_j|}\sum_{i\in S_j}\Phi(\mathbf{x}^i)$ and $\Phi(\mathbf{x})^\top\cdot\Phi(\mathbf{x}') = k(\mathbf{x},\mathbf{x}')$, stacking up n equations ($r = 1,\ldots,n$) results in the following dual version of the eigensystem:

$$\mathbf{GPG}^\top\boldsymbol{\alpha} = \lambda\mathbf{K_x}\boldsymbol{\alpha}, \qquad (2)$$

where $\mathbf{G}$ is the $(n\times J)$ matrix with $\mathbf{G}(r,j) = \mathbf{m}_j^\top\Phi(\mathbf{x}^r) = \frac{1}{|S_j|}\sum_{i\in S_j} k(\mathbf{x}^i,\mathbf{x}^r)$, $\mathbf{P}$ is the $(J\times J)$ diagonal matrix with $\mathbf{P}(j,j) = p_j$, $\mathbf{K_x}$ is the kernel Gram matrix for $\mathbf{x}$, *i.e.*, $\mathbf{K_x}(i,r) = k(\mathbf{x}^i,\mathbf{x}^r)$, and $\boldsymbol{\alpha} = [\alpha_1,\ldots,\alpha_n]^\top$. It is often the case that the eigenfunctions need to be normalized to unit-norm, which introduces extra constraints, $\boldsymbol{\alpha}^\top\mathbf{K_x}\boldsymbol{\alpha} = 1$. Eq.(2) is the generalized eigenvalue problem, where we find the q major eigenvectors $\boldsymbol{\alpha}$.

We note several interesting aspects of KSIR: First, the (linear) SIR can be simply derived from the KSIR algorithm with a linear kernel, $k(\mathbf{x},\mathbf{x}') = \mathbf{x}^\top\mathbf{x}'$. Secondly, similarly to the relationship between SIR and PCA, when $J = n$, KSIR is equivalent to the kernel PCA on $\{\mathbf{x}^i\}_{i=1}^n$. To see this, as $J\rightarrow n$, note that $\mathbf{P}\rightarrow\frac{1}{n}\mathbf{I}_n$, ($\mathbf{I}_n$ is the $(n\times n)$ identity) and $\mathbf{G}\rightarrow\mathbf{K_x}$. Hence Eq.(2) reduces to $\mathbf{K_x}\boldsymbol{\alpha} = n\lambda\boldsymbol{\alpha}$, which is the exact derivation for the kernel PCA [14].

Once we have $\mathbf{v}$ (from the dual solution $\boldsymbol{\alpha}$), the corresponding central subspace direction $\mathbf{b}$ of Step-3 can be obtained using a similar trick. To solve $\mathbf{\Sigma_{xx}}\cdot\mathbf{b} = \mathbf{v}$, we replace the covariance operator $\mathbf{\Sigma_{xx}}$ by the sample estimate $\frac{1}{n}\mathbf{W_x}\mathbf{W_x}^\top$, where $\mathbf{W_x} = [\Phi(\mathbf{x}^1),\ldots,\Phi(\mathbf{x}^n)]$. Letting $\mathbf{b} = \sum_{i=1}^n\beta_i\Phi(\mathbf{x}^i)$ and pre-multiplying by $\Phi(\mathbf{x}^r)^\top$ (for $r = 1,\ldots,n$), leads to the closed-form solution:

$$\boldsymbol{\beta} = n\mathbf{K_x^{-1}}\boldsymbol{\alpha}, \qquad (3)$$

where $\boldsymbol{\beta} = [\beta_1,\ldots,\beta_n]^\top$.

The nonlinear central subspace is then represented by q basis functions, $\{\mathbf{b}_l\}_{l=1}^q$ from the dual solutions $\{\boldsymbol{\beta}^l\}_{l=1}^q$. For a new test input point $\mathbf{x}^*$, its low dimensional representation $\mathbf{z}^*\in\mathbb{R}^q$ can be obtained by projecting $\Phi(\mathbf{x}^*)$ onto the central subspace. That is, the l-th element of $\mathbf{z}^*$ is, $\mathbf{z}_l^* = \mathbf{b}_l^\top\Phi(\mathbf{x}^*) = \mathbf{k}_*^\top\boldsymbol{\beta}^l$, for $l = 1,\ldots,q$, where $\mathbf{k}_* = [k(\mathbf{x}^1,\mathbf{x}^*),\ldots,k(\mathbf{x}^n,\mathbf{x}^*)]^\top$.

The kernel extension of inverse regression resolves certain limitation of the (linear) SIR. Not restricted to a linear central subspace, it allows the distribution of $\mathbf{x}$ to be within a rich family of nonparametric kernel densities. However, KSIR's slicing-based estimation of $\mathbb{V}(\mathbb{E}[\Phi(\mathbf{x})|\mathbf{y}])$ may be unreliable for high dimensional $\mathbf{y}$. This makes KSIR restricted to single-output regression or classification settings [19].

In the following section, we propose a novel estimation method that avoids slicing by exploiting the kernel matrices of the input and the output. As we will see in Sec. 4, our approach is successfully applied to regression problems with a large number of output variables that often arise in computer vision (*e.g.*, 3D body pose estimation, image reconstruction with noise removal).

[6] The centralization of the kernel matrix is fairly straightforward and can be found in Appendix A of [14].

[7] Here, we abuse the notation $\mathbf{\Sigma_{xx}}$ to indicate the covariance *operator*, and the multiplication means applying the operator to the function.

3. Proposed Approach

3.1. IR using Covariance Operators in RKHS

Our estimation of $\mathbb{V}(\mathbb{E}[\Phi(\mathbf{x})|\mathbf{y}])$ is based on the (cross) covariance operator theorems [1, 4]. First, we introduce the *covariance operator* as a natural RKHS extension of the covariance matrix in the original space. For two random vectors $\mathbf{y}$ and $\mathbf{x}$ endowed with Hilbert spaces $\mathcal{H}_\mathbf{y}$ with $k_\mathbf{y}(\cdot,\cdot)$ and $\mathcal{H}_\mathbf{x}$ with $k_\mathbf{x}(\cdot,\cdot)$, respectively, we define $\mathbf{\Sigma_{yx}} \triangleq \text{Cov}(\Phi(\mathbf{y}), \Phi(\mathbf{x}))$, namely, the (cross) covariance in the feature spaces[8]. Note that $\mathbf{\Sigma_{yx}}$ is an operator that maps from $\mathcal{H}_\mathbf{x}$ to $\mathcal{H}_\mathbf{y}$, thus having dimension $(dim(\mathcal{H}_\mathbf{y}) \times dim(\mathcal{H}_\mathbf{x}))$. One can similarly define other covariance operators, $\mathbf{\Sigma_{xy}}$, $\mathbf{\Sigma_{yy}}$, or $\mathbf{\Sigma_{xx}}$.

For notational convenience, we treat the covariance operators as if they were matrices. For instance, for $\mathbf{g} \in \mathcal{H}_\mathbf{y}$ and $\mathbf{f} \in \mathcal{H}_\mathbf{x}$, $\mathbf{g}^\top \mathbf{\Sigma_{yx}} \mathbf{f}$ means the inner product $\langle \mathbf{g}, \mathbf{\Sigma_{yx}} \mathbf{f} \rangle$ in $\mathcal{H}_\mathbf{y}$ space. Besides, $\mathbf{f}^\top \Phi(\mathbf{x})$ and $\mathbf{f}(\mathbf{x})$ are used interchangeably, as they are equivalent from the Riesz representation theorem [16]. We then define the *conditional covariance operator* of $\mathbf{y}$ given $\mathbf{x}$, denoted by $\mathbf{\Sigma_{yy|x}}$, as:

$$\mathbf{\Sigma_{yy|x}} \triangleq \mathbf{\Sigma_{yy}} - \mathbf{\Sigma_{yx}} \mathbf{\Sigma_{xx}^{-1}} \mathbf{\Sigma_{xy}}. \tag{4}$$

The following theorem [4] states that under fairly mild condition, $\mathbf{\Sigma_{yy|x}}$ equals to the expected conditional variance of $\Phi(\mathbf{y})$ given $\mathbf{x}$ (*i.e.*, $\mathbb{E}[\mathbb{V}(\Phi(\mathbf{y})|\mathbf{x})]$). See Appendix for the proof.

Theorem 2. *For any $\mathbf{g} \in \mathcal{H}_\mathbf{y}$, if there exists $\mathbf{f} \in \mathcal{H}_\mathbf{x}$ such that $\mathbb{E}[\mathbf{g}(\mathbf{y})|\mathbf{x}] = \mathbf{f}(\mathbf{x})$, then $\mathbf{\Sigma_{yy|x}} = \mathbb{E}[\mathbb{V}(\Phi(\mathbf{y})|\mathbf{x})]$.*

The condition in Thm. 2 implies that the regression function from $\mathbf{x}$ to $\mathbf{g}(\mathbf{y})$ for any given $\mathbf{g} \in \mathcal{H}_\mathbf{y}$ has to be *linear* in RKHS, namely, of the form $\mathbf{f}^\top \Phi(\mathbf{x})$ for some $\mathbf{f} \in \mathcal{H}_\mathbf{x}$. This is a reasonable condition as it corresponds to a rich family of smooth functions in the original space (*i.e.*, $\mathbf{f}^\top \Phi(\mathbf{x}) = \mathbf{f}(\mathbf{x})$) [13].

We next propose to represent $\mathbb{V}(\mathbb{E}[\Phi(\mathbf{x})|\mathbf{y}])$ of IR in terms of the conditional covariance operators. More specifically, using the well-known *E-V-V-E* identity[9], it can be written as:

$$\mathbb{V}(\mathbb{E}[\Phi(\mathbf{x})|\mathbf{y}]) = \mathbb{V}(\Phi(\mathbf{x})) - \mathbb{E}[\mathbb{V}(\Phi(\mathbf{x})|\mathbf{y})]. \tag{5}$$

From Thm. 2, the second term of the RHS in Eq.(5) equals to $\mathbf{\Sigma_{xx|y}}$ (changing the role of $\mathbf{x}$ and $\mathbf{y}$ in Eq.(4)), assuming that the inverse regression, $\mathbb{E}[\mathbf{f}(\mathbf{x})|\mathbf{y}]$, is a smooth function of $\mathbf{y}$ for any $\mathbf{f} \in \mathcal{H}_\mathbf{x}$ (*i.e.*, $\exists \mathbf{g} \in \mathcal{H}_\mathbf{y}$ s.t. $\mathbb{E}[\mathbf{f}(\mathbf{x})|\mathbf{y}] = \mathbf{g}(\mathbf{y})$). As $\mathbb{V}(\Phi(\mathbf{x})) = \text{Cov}(\Phi(\mathbf{x}), \Phi(\mathbf{x})) = \mathbf{\Sigma_{xx}}$, we have:

$$\mathbb{V}(\mathbb{E}[\Phi(\mathbf{x})|\mathbf{y}]) = \mathbf{\Sigma_{xy}} \mathbf{\Sigma_{yy}^{-1}} \mathbf{\Sigma_{yx}}. \tag{6}$$

[8] A more precise definition would be: for $\forall \mathbf{g} \in \mathcal{H}_\mathbf{y}$ and $\forall \mathbf{f} \in \mathcal{H}_\mathbf{x}$, $\langle \mathbf{g}, \mathbf{\Sigma_{yx}} \mathbf{f} \rangle = \mathbb{E}[(\mathbf{g}(\mathbf{y}) - \mathbb{E}\mathbf{g}(\mathbf{y}))(\mathbf{f}(\mathbf{x}) - \mathbb{E}\mathbf{f}(\mathbf{x}))]$.

[9] $\mathbb{V}(Y) = \mathbb{E}[\mathbb{V}(Y|X)] + \mathbb{V}(\mathbb{E}[Y|X])$ for any X, Y.

Given the data $\{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^n$, the sample estimate of Eq.(6) can be written as $\widehat{\mathbb{V}}(\mathbb{E}[\Phi(\mathbf{x})|\mathbf{y}]) = \widehat{\mathbf{\Sigma}}_\mathbf{xy} \widehat{\mathbf{\Sigma}}_\mathbf{yy}^{-1} \widehat{\mathbf{\Sigma}}_\mathbf{yx}$. The sample covariance *operators* ($\widehat{\mathbf{\Sigma}}$) can be estimated similarly. For instance, $\widehat{\mathbf{\Sigma}}_\mathbf{xy} = \frac{1}{n} \mathbf{W_x} \mathbf{W_y}^\top$, where $\mathbf{W_x} = [\Phi(\mathbf{x}^1), \ldots, \Phi(\mathbf{x}^n)]$ and $\mathbf{W_y} = [\Phi(\mathbf{y}^1), \ldots, \Phi(\mathbf{y}^n)]$. Then $\widehat{\mathbb{V}}(\mathbb{E}[\Phi(\mathbf{x})|\mathbf{y}])$ is:

$$\begin{aligned}
&(\frac{1}{n}\mathbf{W_x}\mathbf{W_y}^\top)(\frac{1}{n}(\mathbf{W_y}\mathbf{W_y}^\top + n\epsilon\mathbf{I}))^{-1}(\frac{1}{n}\mathbf{W_y}\mathbf{W_x}^\top) \\
&= \frac{1}{n}\mathbf{W_x}\mathbf{W_y}^\top(\mathbf{W_y}\mathbf{W_y}^\top + n\epsilon\mathbf{I})^{-1}\mathbf{W_y}\mathbf{W_x}^\top \\
&= \frac{1}{n}\mathbf{W_x}\mathbf{W_y}^\top\mathbf{W_y}(\mathbf{W_y}^\top\mathbf{W_y} + n\epsilon\mathbf{I}_n)^{-1}\mathbf{W_x}^\top \qquad (7) \\
&= \frac{1}{n}\mathbf{W_x}\mathbf{K_y}(\mathbf{K_y} + n\epsilon\mathbf{I}_n)^{-1}\mathbf{W_x}^\top. \qquad (8)
\end{aligned}$$

Here $\mathbf{I}$ is the $(dim(\mathcal{H}_\mathbf{y}) \times dim(\mathcal{H}_\mathbf{y}))$ identity operator, and $\mathbf{I}_n$ is the $(n \times n)$ identity matrix. Note that we add a small positive ϵ to the diagonals of $\mathbf{W_y}\mathbf{W_y}^\top$ to circumvent potential rank deficiency in estimating $\mathbf{\Sigma_{yy}}$ and its inverse. As will be discussed in Sec. 3.2, ϵ plays a crucial role as a kernel regularizer in smoothing the affinity structure of $\mathbf{y}$. In Eq.(7), we use the fact that $\mathbf{W_y}(\mathbf{W_y}^\top\mathbf{W_y} + n\epsilon\mathbf{I}_n) = (\mathbf{W_y}\mathbf{W_y}^\top + n\epsilon\mathbf{I})\mathbf{W_y}$. In Eq.(8), $\mathbf{K_y} = \mathbf{W_y}^\top\mathbf{W_y}$ is the $(n \times n)$ kernel matrix on $\mathbf{y}$, *i.e.*, $\mathbf{K_y}(i,r) = k_\mathbf{y}(\mathbf{y}^i, \mathbf{y}^r)$.

The eigenfunctions of Eq.(8) can be obtained by premultiplying the eigensystem, $\widehat{\mathbb{V}}(\mathbb{E}[\Phi(\mathbf{x})|\mathbf{y}]) \cdot \mathbf{v} = \lambda \cdot \mathbf{v}$, by $\mathbf{W_x}^\top$. From $\mathbf{v} = \sum_{i=1}^n \alpha_i \Phi(\mathbf{x}^i) = \mathbf{W_x}\boldsymbol{\alpha}$, we have:

$$\frac{1}{n}\mathbf{K_y}(\mathbf{K_y} + n\epsilon\mathbf{I}_n)^{-1}\mathbf{K_x}\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}, \tag{9}$$

where $\mathbf{K_x} = \mathbf{W_x}^\top\mathbf{W_x}$ is the kernel matrix on $\mathbf{x}$. Once we obtain $\mathbf{v}$, the Step-3 of KSIR follows to find $\mathbf{b}$. From now on, we denote the inverse regression technique described in this section by *COIR* (*i.e.*, Covariance Operator IR).

COIR has a closed-form solution (Eq.(9)) to the nonlinear central subspace, while making few assumptions on the input distribution. Notably, COIR avoids KSIR's slicing-based estimation by incorporating a smooth output kernel. This makes COIR not only able to handle high-dimensional output reliably, but also robust to potential noise in the output data. Furthermore, we show that, under this formulation, KSIR becomes a special case of COIR.

3.2. KSIR as a special case of COIR

Recall from KSIR that $\mathbf{P}$ is the $(J \times J)$ cluster proportion diagonal matrix with the j-th element, $p_j = |S_j|/n$. We let $\mathbf{C}$ be the $(n \times J)$ cluster indicator $0/1$ matrix whose i-th row has all 0's but 1 at the j-th position where $i \in S_j$. Noticing that $\mathbf{G} = \frac{1}{n}\mathbf{K_x}\mathbf{C}\mathbf{P}^{-1}$, the KSIR equation in Eq.(2) can be written as:

$$\frac{1}{n}\mathbf{K_x}(\frac{1}{n}\mathbf{C}\mathbf{P}^{-1}\mathbf{C}^\top)\mathbf{K_x}\boldsymbol{\alpha} = \lambda\mathbf{K_x}\boldsymbol{\alpha}. \tag{10}$$

On the other hand, the COIR equation in Eq.(9) is (after pre-multiplying by $\mathbf{K_x}$):

$$\frac{1}{n}\mathbf{K_x}(\mathbf{K_y}(\mathbf{K_y}+n\epsilon\mathbf{I}_n)^{-1})\mathbf{K_x}\boldsymbol{\alpha}=\lambda\mathbf{K_x}\boldsymbol{\alpha}. \quad (11)$$

From Eq.(10) and Eq.(11), the equivalence between KSIR and COIR is made by:

$$\mathbf{K_y}(\mathbf{K_y}+n\epsilon\mathbf{I}_n)^{-1}=\frac{1}{n}\mathbf{C}\mathbf{P}^{-1}\mathbf{C}^\top. \quad (12)$$

Now we consider an ideal case where the output data $\{\mathbf{y}^i\}_{i=1}^n$ is collapsed to J distinct points that are infinitely far apart from one another. Assuming an RBF kernel, this in turn makes $\mathbf{K_y}$ a block $0/1$ matrix[10], where each block of 1's corresponds to each of J clusters. For instance, when $n=6, J=3, |S_1|=3, |S_2|=1$, and $|S_3|=2$,

$$\mathbf{K_y}=\begin{bmatrix}\mathbf{E}_3 & 0 & 0\\ 0 & \mathbf{E}_1 & 0\\ 0 & 0 & \mathbf{E}_2\end{bmatrix}, \quad (13)$$

where $\mathbf{E}_m$ denotes the $(m\times m)$ matrix with all 1's.

We show that under the above assumption, Eq.(12) is indeed true when $\epsilon\to 0$. For the block $0/1$ Gram matrix $\mathbf{K_y}$ (*e.g.*, Eq.(13)), the LHS of Eq.(12) can be expressed as:

$$\mathbf{K_y}(\mathbf{K_y}+n\epsilon\mathbf{I}_n)^{-1}=\begin{bmatrix}c_1\mathbf{E}_{|S_1|} & & 0\\ & \ddots & \\ 0 & & c_J\mathbf{E}_{|S_J|}\end{bmatrix}, \quad (14)$$

where $c_j=\frac{1}{|S_j|+n\epsilon}$. One can easily verify this by post-multiplying both sides of Eq.(14) by $(\mathbf{K_y}+n\epsilon\mathbf{I}_n)$. It is also straightforward to rewrite the RHS of Eq.(12) as:

$$\frac{1}{n}\mathbf{C}\mathbf{P}^{-1}\mathbf{C}^\top=\begin{bmatrix}\frac{1}{np_1}\mathbf{E}_{|S_1|} & & 0\\ & \ddots & \\ 0 & & \frac{1}{np_J}\mathbf{E}_{|S_J|}\end{bmatrix}. \quad (15)$$

Therefore, Eq.(12) reduces to:

$$|S_j|+n\epsilon=np_j, \text{ for } j=1,\ldots,J. \quad (16)$$

As $\epsilon\to 0$, Eq.(16) implies that $p_j=|S_j|/n$, which is exactly the maximum likelihood (ML) estimate of the cluster proportion employed by KSIR. That is, KSIR is a special case of COIR having $0/1$ Gram matrix $\mathbf{K_y}$ (from the assumed J-collapsed perfect clustering) with a vanishing ϵ. For a non-negligible ϵ, the equivalence turns into $p_j=|S_j|/n+\epsilon$, where ϵ now serves as a regularizer (or a smoothing prior) in the ML estimation.

For a general (non-$0/1$) kernel matrix $\mathbf{K_y}$, COIR can be naturally viewed as a smoothed extension of KSIR. Hence, COIR exploits the kernel structure of the output space through an effective use of covariance operators in RKHS, where ϵ acts as a kernel regularizer.

[10] After reordering the data according to its slice index.

4. Empirical Evaluation

We demonstrate the benefits of COIR by contrasting it to the existing DRR techniques, SIR and KSIR. We also compare it with an unsupervised dimension reduction technique, the kernel PCA (denoted by KPCA)[11] to illustrate the advantage of DRR. Unless stated otherwise, the kernel-based methods (*i.e.*, COIR, KSIR, and KPCA) employ the RBF kernel. SIR and KSIR use the k-means clustering algorithm for output slicing[12].

4.1. Synthetic curves dataset

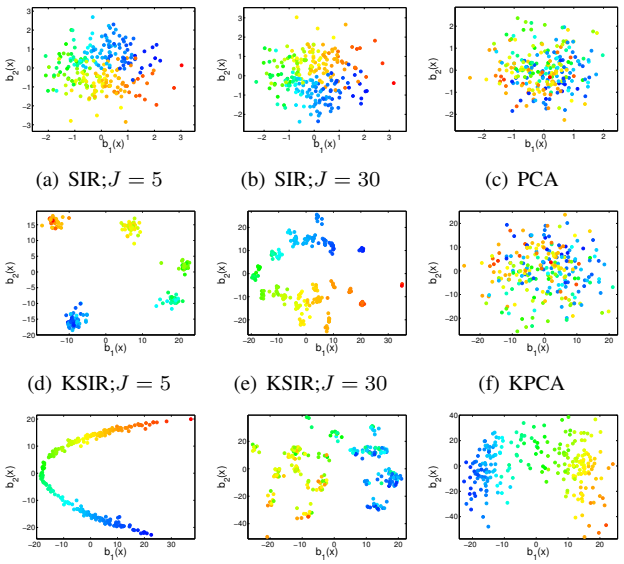
The dataset called *curves* was devised for testing KSIR in [19], where it is generated by: $y=\text{sign}(\mathbf{b}_1^\top\mathbf{x}+\epsilon_1)\cdot\log(|\mathbf{b}_2^\top\mathbf{x}+a_0+\epsilon_2|)$ for some $\mathbf{b}_1,\mathbf{b}_2\in\mathbb{R}^{15}$, $\mathbf{x}\sim\mathcal{N}(0,\mathbf{I}_{15})$, $\epsilon_1,\epsilon_2\sim\mathcal{N}(0,1)$, and a constant a_0. The input is 15-dim, but the central subspace is at most 2-dim as y is decided by $\{\mathbf{b}_l^\top\mathbf{x}\}_{l=1}^2$. The output y is 1-dim. For 300 data points generated, we plot the 2D central subspaces estimated by the competing methods in Fig. 1. In the plots, each point is colored by its true y value, depicting higher values as warmer (reddish) and lower as cooler (bluish). For SIR and KSIR, we vary J (#slices), where the extreme case of $J=n$ gives rise to PCA and KPCA, respectively.

In SIR (regardless of J), the data points are roughly grouped into 4 clusters by y values: red, yellow, green, and blue. However, the partial overlap of points in red (higher y) and blue (lower y) can result in significant error in the regression estimation based on it. KSIR is more sensitive to the choice of J. When $J=5$, the clusters are separated better, but mixed within each cluster. Increasing J resolves mixing, but the clusters get closer to one another. On the other hand, COIR (Fig. 1(g)) exhibits smooth and clear discrimination of data along the y values. Moreover, COIR does not require choosing the parameter J, a sensitive task necessary for KSIR. The unsupervised PCA and KPCA produce random clutter since they simply project the isotropic Gaussian data onto a 2D plane with no information about y.

To simulate the noisy nature of real-world data, we devise an interesting setting by adding 4 Gaussian white noise dimensions to y (5-dim in total). The results of KSIR and COIR are shown in Fig. 1(h) and Fig. 1(i), respectively, colored by true (noise-removed) y values. In KSIR, due to the clustering error induced by noise, each cluster contains mixed data points with different y values (*e.g.*, red, yellow, and green points in the same cluster). However, COIR still lays out the data along the y values (from blue/left to red/right), which enables even a simple linear regressor (*e.g.*, linear in the X axis) to produce good predictions of tar-

[11] Although we do not present other unsupervised methods here, their results are not significantly different from those of the KPCA.

[12] Additional experiments and results can be found on http://seqam.rutgers.edu/projects/learning/regression/coir/coir.html.



(a) SIR;$J = 5$ (b) SIR;$J = 30$ (c) PCA

(d) KSIR;$J = 5$ (e) KSIR;$J = 30$ (f) KPCA

(g) COIR (h) KSIR (noisy) (i) COIR (noisy)

Figure 1. Central subspaces for the curves dataset.

get values from the central subspace coordinates. COIR's robustness to noise originates from its utilization of the smooth output kernel.

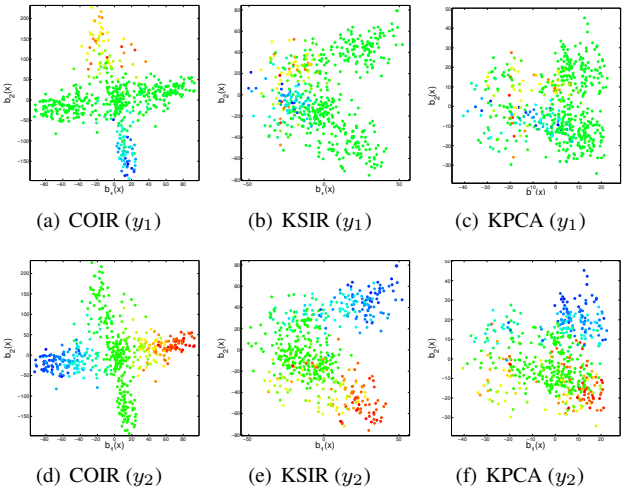
4.2. Head pose estimation

We consider a regression setting that estimates the head (or facial) pose from the face image. We collected $n = 683$ images containing faces of about 100 subjects instructed to move their heads in arbitrary rotation. A standard face detector was applied (with a manual refinement) to locate a tight bounding box around the face. To suppress the in-plane head tilt, we align the face in up-front. The image is further resized to (80×80), a 6400-dim input vector.

For the output, we recorded approximate angles for the out-of-plane rotation along X and Y axes. Note that this introduces substantial amount of noise in the output data. We denote the 2D output by $\mathbf{y} = [y_1, y_2]^\top$, where y_1 and y_2 are vertical and horizontal rotation angles, respectively. In addition, the data is *sparse* in the $\mathbf{y}$ space as most of the data points (about 90%) have one of the angles equal to 0 (*i.e.*, purely horizontal or vertical movement).

Fig. 2 shows the 2D central subspaces estimated by the competing nonlinear methods[13]. For visualization, we colored the dim-reduced input point by each of y_1 and y_2 (*e.g.*, Fig. 2(a) and Fig. 2(d) depict the same points for COIR, but colored by y_1 and y_2, respectively). We see that COIR lays out the data points along the head pose quite obviously, where X and Y axes roughly correspond to horizontal (y_2) and vertical (y_1) angles, respectively. On the other hand,

[13] We skip the (linear) SIR result as its performance is worse than KSIR.



(a) COIR (y_1) (b) KSIR (y_1) (c) KPCA (y_1)

(d) COIR (y_2) (e) KSIR (y_2) (f) KPCA (y_2)

Figure 2. Central subspaces for head pose estimation.

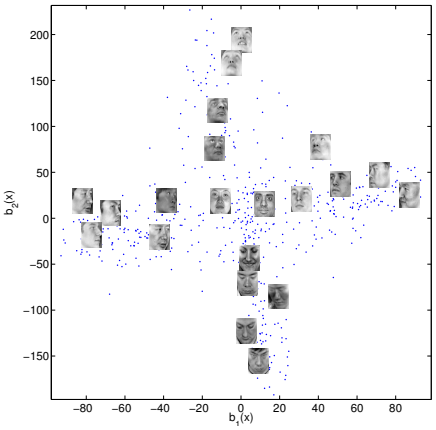


Figure 3. Test face images on the COIR central subspace.

KSIR exhibits severe ambiguity in y_1. For COIR, we also superimpose some of the held-out test face images (whose $\mathbf{y}$ are not known to the algorithm) in Fig. 3. Despite the sparsity and the noise in the output data, COIR generalizes well even for the combined rotation (*e.g.*, the face image at around $(40, 70)$).

4.3. Human body pose estimation

Another interesting problem is to estimate the human body pose from a silhouette image. It is advantageous to apply DRR techniques as one often wants to find the intrinsic low-dimensional (2D or 3D as is widely believed) representation of the input image in predicting the body pose.

We use the walking sequence of length ~ 400 (containing about 3 walking cycles) obtained from the CMU motion capture database[14]. The output $\mathbf{y}$ is composed of 59

[14] Available at http://mocap.cs.cmu.edu.

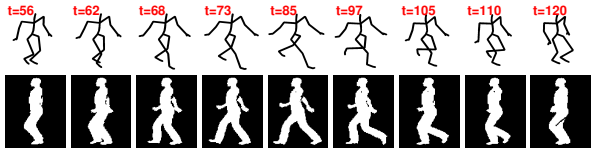


Figure 4. Selected skeleton and silhouette images for a half walking cycle: From stand-up ($t = 56$), right-leg forwarding, left-leg catching-up, and back to stand-up ($t = 120$). The skeleton images are drawn using the 3D joint angles.

Input Space	COIR	KSIR	KPCA	Image $\mathbf{x}$
NN Regression	6.1782	8.9054	8.6592	6.5151
GP Regression	5.8632	8.1602	8.5436	5.9535

Table 1. Test (RMS) errors in 3D body pose estimation.

3D joint angles at 31 articulation points. The input $\mathbf{x}$ is the silhouette image of size (160×100) taken at a side view. As the silhouette image is ambiguous in discriminating two opposite poses with left/right arms/legs switched, we focus on a half cycle as shown in Fig. 4.

Trained on the first 80% of the frames, the 2D central subspaces estimated by COIR, KSIR, and KPCA are shown in Fig. 5. COIR yields a circular trajectory unambiguous within a half cycle. On the other hand, KSIR is distorted at the beginning/end of the half cycle. Especially, the points at $t = 62$ and $t = 110$ adjoin each other too closely, which would result in a large estimation error in pose prediction. This illustrates that KSIR's slicing-based estimation is unreliable for high-dimensional output. Note that the unsupervised KPCA shows a much severer distortion than KSIR. Moreover, COIR generalizes well for the test (red) points.

The actual regression estimation is also conducted. We employ two most popular regression methods: the nearest neighbor (NN) and the Gaussian Process (GP) regression[15] [18]. See Table 1 for the test RMS errors. Even though the regressors are built on the dimension-reduced (2D) input space, the prediction performance of COIR is never worse than that based on the silhouette image $\mathbf{x}$ itself as input. On the other hand, dimension reduction by KSIR (or KPCA) entails significant loss of information in predicting the output.

4.4. Hand-written digit image reconstruction

To test the behavior of COIR on high-dimensional output data we devise an image denoising experiment with the USPS hand-written digit images [8]. By adding random scratch lines with varying thickness and orientation on the normalized (16×16) digit images, the task is to denoise or reconstruct the image. So, the regression problem is to predict the original unscratched image (output $\mathbf{y}$) from the scratched image (input $\mathbf{x}$). Both $\mathbf{y}$ and $\mathbf{x}$ are of 256-dim.

[15]For the multiple output regression, we assume independent GP priors, which results in independent GP prediction for each output dimension [7].

Input Space	COIR	KSIR	Image $\mathbf{x}$
NN Regression	8.5334	11.4909	9.3605
GP Regression	8.1454	10.7259	9.1036

Table 2. Test (RMS) errors in scratched digit image denoising.

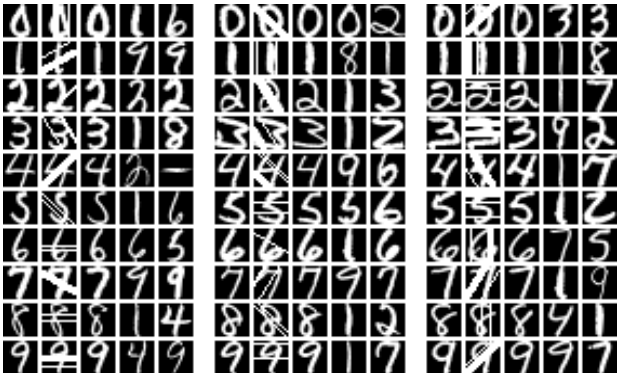
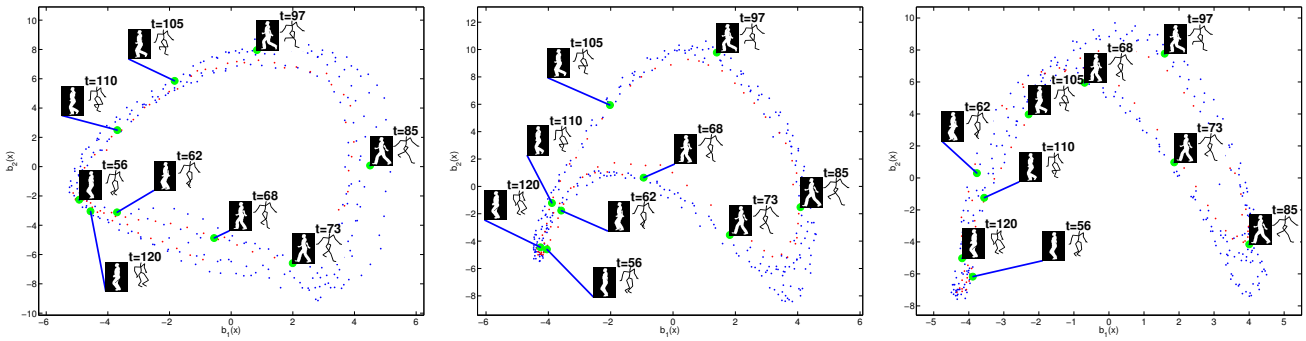


Figure 6. Denoising USPS scratched digit images. Each 5-tuple is composed of, from left to right, (1^{st}) the noise-free test image, (2^{nd}) randomly scratched image, (3^{rd}) denoised by NN on COIR, (4^{th}) NN on KSIR, and (5^{th}) NN on the scratched image $\mathbf{x}$ itself.

From the database, we use a subset of 2000 images for training and another 2000 for testing. The central subspace dimension is chosen as 30. The test reconstruction (denoising) RMS errors are shown in Table 2, while some of the denoised test images by the NN regression are depicted in Fig. 6. We can see that COIR is robust to noise with improved prediction accuracy compared to the regression based on the image input itself. KSIR again suffers from unreliable slicing-based estimation in the high dimensional output space.

5. Conclusion

The DRR framework, as a supervised dimension reduction with a real multivariate label, is useful for visualization of high dimensional data, efficient regressor design with a reduced input dimension, and elimination of noise in input data by uncovering the essential information for predicting output. We have proposed a novel nonlinear method for DRR that exploits the kernel matrices of the input and the output using the covariance operators in RKHS. In a comprehensive set of evaluations, we have demonstrated that our approach can successfully discover central subspaces reliably and robustly for high dimensional noisy data. In future work, we plan to extend the framework to a semi-supervised setting to take advantage of large datasets with sparsely labeled data.



(a) COIR (b) KSIR (c) KPCA

Figure 5. Central subspaces for silhouette images from walking motion: The blue (red) points indicate train (test) data points.

Acknowledgments

This work was supported in part by the NSF IIS grant #0413105.

References

[1] C. R. Baker. Joint measures and cross-covariance operators. *Transactions of the American Mathematical Society*, 186:273–289, 1973.

[2] M. Belkin and P. Niyogi. Laplacian Eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[3] R. D. Cook. *Regression graphics*. Wiley Inter-Science, 1998.

[4] K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 2004.

[5] A. Globerson and S. Roweis. Metric learning by collapsing classes, 2005. Neural Information Processing Systems (NIPS).

[6] N. D. Lawrence. Gaussian process models for visualisation of high dimensional data, 2003. Neural Information Processing Systems (NIPS).

[7] N. D. Lawrence. The Gaussian process latent variable model, 2006. Technical Report CS-06-03, The University of Sheffield, Department of Computer Science.

[8] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network, 1989. Neural Information Processing Systems (NIPS).

[9] K.-C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 1991.

[10] J. Nilsson, F. Sha, and M. I. Jordan. Regression on manifolds using kernel dimension reduction, 2007. International Conference on Machine Learning (ICML).

[11] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[12] B. Schölkopf, R. Herbrich, and A. Smola. A Generalized representer theorem. *Computational Learning Theory*, 2111:416–426, 2001.

[13] B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

[14] B. Schölkopf, A. J. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[15] J. B. Tenenbaum, V. D. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[16] G. Wahba. *Spline Models for Observational Data*. SIAM, 1990.

[17] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification, 2005. Neural Information Processing Systems (NIPS).

[18] C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression, 1996. Neural Information Processing Systems (NIPS).

[19] H. M. Wu. Kernel sliced inverse regression with applications on classification, 2006. ICSA Applied Statistics Symposium.

[20] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information, 2002. Neural Information Processing Systems (NIPS).

[21] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. on PAMI*, 29(1):40–51, 2007.

Appendix: Proof of Thm. 2

To prove Thm. 2, we need the following lemma which states that $\boldsymbol{\Sigma}_{\mathbf{yy}|\mathbf{x}}$ is tightly related to the optimal linear (in RKHS) regressor in terms of the variance of the error. More specifically, when we regress from $\mathbf{x}$ to $\mathbf{g}(\mathbf{y})$ for a given $\mathbf{g} \in \mathcal{H}_{\mathbf{y}}$, the variance of the prediction error cannot be smaller than $\mathbf{g}^\top \boldsymbol{\Sigma}_{\mathbf{yy}|\mathbf{x}} \mathbf{g}$.

Lemma 3. *For any* $\mathbf{g} \in \mathcal{H}_{\mathbf{y}}$, $\inf_{\mathbf{f} \in \mathcal{H}_{\mathbf{x}}} \mathbb{V}(\mathbf{g}(\mathbf{y}) - \mathbf{f}(\mathbf{x})) = \inf_{\mathbf{f} \in \mathcal{H}_{\mathbf{x}}} \mathbb{V}(\mathbf{g}^\top \Phi(\mathbf{y}) - \mathbf{f}^\top \Phi(\mathbf{x})) = \mathbf{g}^\top \boldsymbol{\Sigma}_{\mathbf{yy}|\mathbf{x}} \mathbf{g}$.

Proof. From the co-linearity of $\mathrm{Cov}(\cdot, \cdot)$, $\mathbb{V}(\mathbf{g}^\top \Phi(\mathbf{y}) - \mathbf{f}^\top \Phi(\mathbf{x})) = \mathrm{Cov}(\mathbf{g}^\top \Phi(\mathbf{y}) - \mathbf{f}^\top \Phi(\mathbf{x}), \mathbf{g}^\top \Phi(\mathbf{y}) - \mathbf{f}^\top \Phi(\mathbf{x})) = \mathbf{g}^\top \boldsymbol{\Sigma}_{\mathbf{yy}} \mathbf{g} - 2\mathbf{f}^\top \boldsymbol{\Sigma}_{\mathbf{xy}} \mathbf{g} + \mathbf{f}^\top \boldsymbol{\Sigma}_{\mathbf{xx}} \mathbf{f}$. As the latter is quadratic (convex) in $\mathbf{f}$, by taking the gradient to 0, namely, $\partial \mathbf{f} = -2\boldsymbol{\Sigma}_{\mathbf{xy}} \mathbf{g} + 2\boldsymbol{\Sigma}_{\mathbf{xx}} \mathbf{f} = 0$, its infimum is found at $\mathbf{f}^* = \boldsymbol{\Sigma}_{\mathbf{xx}}^{-1} \boldsymbol{\Sigma}_{\mathbf{xy}} \mathbf{g}$. Plugging $\mathbf{f}^*$ back yields $\mathbf{g}^\top \boldsymbol{\Sigma}_{\mathbf{yy}|\mathbf{x}} \mathbf{g}$. □

From Lemma 3 and the well-known *E-V-V-E* identity, we have $\mathbf{g}^\top \boldsymbol{\Sigma}_{\mathbf{yy}|\mathbf{x}} \mathbf{g} = \inf_{\mathbf{f} \in \mathcal{H}_{\mathbf{x}}} \mathbb{V}(\mathbf{g}(\mathbf{y}) - \mathbf{f}(\mathbf{x})) = \inf_{\mathbf{f} \in \mathcal{H}_{\mathbf{x}}} \Big\{ \mathbb{E}[\mathbb{V}(\mathbf{g}(\mathbf{y}) - \mathbf{f}(\mathbf{x})|\mathbf{x})] + \mathbb{V}(\mathbb{E}[\mathbf{g}(\mathbf{y}) - \mathbf{f}(\mathbf{x})|\mathbf{x}]) \Big\} = \mathbb{E}[\mathbb{V}(\mathbf{g}(\mathbf{y})|\mathbf{x})] + \inf_{\mathbf{f} \in \mathcal{H}_{\mathbf{x}}} \mathbb{V}(\mathbb{E}[\mathbf{g}(\mathbf{y})|\mathbf{x}] - \mathbf{f}(\mathbf{x}))$. Note that the second term is non-negative. From the assumption, as there always exists $\mathbf{f} \in \mathcal{H}_{\mathbf{x}}$ that makes the second term 0, $\mathbf{g}^\top \boldsymbol{\Sigma}_{\mathbf{yy}|\mathbf{x}} \mathbf{g} = \mathbb{E}[\mathbb{V}(\mathbf{g}(\mathbf{y})|\mathbf{x})] = \mathbf{g}^\top \mathbb{E}[\mathbb{V}(\Phi(\mathbf{y})|\mathbf{x})] \mathbf{g}$ for any $\mathbf{g} \in \mathcal{H}_{\mathbf{y}}$. This completes the proof.