MULTIMODAL METRIC LEARNING WITH LOCAL CCA

Or Yair and Ronen Talmon

Technion - Israel Institute of Technology, Haifa 32000, Israel

{oryair@campus, ronen@ee}.technion.ac.il

ABSTRACT

In this paper, we address the problem of multimodal signal processing from a kernel-based manifold learning standpoint. We propose a data-driven method for extracting the common hidden variables from two multimodal sets of nonlinear high-dimensional observations. To this end, we present a metric based on local canonical correlation analysis (CCA). Our approach can be viewed both as an extension of CCA to a nonlinear setting as well as an extension of manifold learning to multiple data sets. We test our method in simulations, where we show that it indeed discovers the common variables hidden in high-dimensional nonlinear observations without assuming prior rigid model assumptions.

Index Terms— CCA, Diffusion Maps, Metric Learning, Multimodal

1. INTRODUCTION

Nowadays, many devices and systems, e.g., mobile-phones, laptops, and wearable-devices incorporate multiple sensors. In addition, massive data sets of medical recordings and healthcare-related information are acquired and stored routinely, for example, in operation rooms, intensive care units, and clinics. This extensive collection and storage of multimodal data call for the development of data fusion methods and techniques [1]. Here, the specific problem of discovering the common variable underlying two sensor observations is considered. This problem has been studied in the last several decades and has been approached from various research directions. The classic approach to this problem is Canonical Correlation Analysis (CCA) [2], where linear projections maximizing the correlation between the two sensor data are constructed. This strictly linear setting has been extended by Kernel CCA (KCCA) [3], where the maximal correlation criterion is applied in a kernel space. Recently, a gamut of work extending CCA based on various combinations and manipulations of kernels has been presented, e.g., [4–8].

In the current work, a manifold learning approach is presented. The core of manifold learning resides in the construction of a kernel representing affinities between data samples based on pairwise distance metrics [9–11]. Such distance metrics define local relationships, which are then aggregated into a global *nonlinear* representation of the entire data set. Indeed, in recent studies, various local distance metrics extending the usage of the prototypical Euclidean metric in the context of kernel-based manifold learning have been introduced, e.g. [12–15].

In this paper, we propose a manifold learning method, which recovers a nonlinear parametrization of the common variables underlying two sensor observations. The main contribution is the construction of a Riemannian metric to measure affinities, which is shown to extract the common variables locally. This metric learning can be viewed from two different, yet complementing standpoints. First, the metric can be interpreted as the Euclidean distance between locally linear projections of the data samples obtained by locally applying CCA. Second, it can be seen as an extension of the Mahalanobis distance, which has recently been studied in the context of manifold learning [12–14], to multiple data sets [16, 17]. Once we build this Riemannian metric that locally extracts the common variables, we incorporate it in a kernel and proceed, similarly to standard manifold learning approaches, to obtain a global nonlinear parametrization.

The work was supported by the EU Seventh Framework Programme (FP7) under Marie Curie Grant No. 630657.

2. PROBLEM FORMULATION AND BACKGROUND

We assume a set of d_z isotropic variables $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{I}_d)$, $\boldsymbol{z} \in \mathbb{R}^{d_z}$, representing the hidden state of the system of interest. Our only access to these variables is via two possibly nonlinear and locally invertable observation functions (representing two sensors) that introduce additional variables and are given by

$$\boldsymbol{x} = f(\boldsymbol{z}, \boldsymbol{\epsilon}),\ \boldsymbol{y} = g(\boldsymbol{z}, \boldsymbol{\eta}) \qquad \boldsymbol{x} \in \mathbb{R}^{d_x},\ \boldsymbol{y} \in \mathbb{R}^{d_y}$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^{d_\epsilon}$ and $\boldsymbol{\eta} \in \mathbb{R}^{d_\eta}$ are (hidden) sensors-specific variables, whose probability densities are unknown. We assume that the common variables $\boldsymbol{z}$ and the sensor-specific variables $\boldsymbol{\epsilon}$ and $\boldsymbol{\eta}$ are uncorrelated, i.e., $\boldsymbol{\Sigma}_{z\epsilon} = \boldsymbol{\Sigma}_{z\eta} = \boldsymbol{\Sigma}_{\epsilon\eta} = \boldsymbol{0}$, where $\boldsymbol{\Sigma}_{ab} = \mathbb{E}\left[\boldsymbol{a}\boldsymbol{b}^T\right]$. In addition, we assume that the observations are in higher dimension, i.e., $d_z + d_\epsilon \leq d_x$, $d_z + d_\eta \leq d_y$. Given N realizations of the hidden variables, $\{\boldsymbol{z}_i, \boldsymbol{\epsilon}_i, \boldsymbol{\eta}_i\}_{i=1}^N$, we obtain two data sets of observations:

$$\mathcal{X} = \left\{ \boldsymbol{x}_i \middle| \boldsymbol{x}_i = f(\boldsymbol{z}_i, \boldsymbol{\epsilon}_i) \right\}_{i=1}^N,\ \mathcal{Y} = \left\{ \boldsymbol{y}_i \middle| \boldsymbol{y}_i = g(\boldsymbol{z}_i, \boldsymbol{\eta}_i) \right\}_{i=1}^N$$

Our goal is to derive a parametrization of the hidden common variables $\boldsymbol{z}$ from the two sets $\mathcal{X}$ and $\mathcal{Y}$. This is achieved by kernel-based manifold learning with a proper metric, which discards the sensor-specific variables, namely, $\boldsymbol{\epsilon}$ and $\boldsymbol{\eta}$. More specifically, we derive a pairwise metric D_{ij} from the sets $\mathcal{X}$ and $\mathcal{Y}$ that corresponds to the Euclidean distance between the common variables $\boldsymbol{z}$:

$$D_{ij} \approx \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2,\ i, j = 1, \ldots, N.$$

In the remainder of this section, we briefly describe the CCA algorithm and define the notation that we use throughout the paper. Given two random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ with zero mean, the CCA algorithm results in two sets of d directions $\boldsymbol{P}_x \in \mathbb{R}^{d_x \times d}$ and $\boldsymbol{P}_y \in \mathbb{R}^{d_y \times d}$ where $d \triangleq \min\left(\operatorname{rank}\left(\boldsymbol{\Sigma}_{xx}\right), \operatorname{rank}\left(\boldsymbol{\Sigma}_{yy}\right)\right)$. The first column $\boldsymbol{p}_x$ of $\boldsymbol{P}_x$ and the first column $\boldsymbol{p}_y$ of $\boldsymbol{P}_y$ are the directions that maximize the correlation between the projected entries $\boldsymbol{p}_x^T\boldsymbol{x}$ and $\boldsymbol{p}_y^T\boldsymbol{y}$, i.e.

$$\rho_1 \triangleq \max_{\boldsymbol{p}_x, \boldsymbol{p}_y} \frac{\mathbb{E}\left[\boldsymbol{p}_x^T\boldsymbol{x}\boldsymbol{p}_y^T\boldsymbol{y}\right]}{\sqrt{\mathbb{E}\left[\left(\boldsymbol{p}_x^T\boldsymbol{x}\right)^2\right]\mathbb{E}\left[\left(\boldsymbol{p}_y^T\boldsymbol{y}\right)^2\right]}}$$

978-1-4673-7802-4/16/$31.00 ©2016 IEEE

For unique solution, the following constraint is applied $\mathbb{E}\left[(\boldsymbol{p}_x^T\boldsymbol{x})^2\right] = \mathbb{E}\left[(\boldsymbol{p}_y^T\boldsymbol{y})^2\right] = 1$. The remaining columns of $\boldsymbol{P}_x$ and $\boldsymbol{P}_y$ are obtained in a similar manner, such that the k-th column is the k-th direction that maximizes the correlation between the projected data, where the k-th projected data is orthogonal to the previous $1,\ldots,k-1$ projections. For more details, see [2]. Overall, applying CCA to the two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ results in the matrices $\boldsymbol{P}_x$, $\boldsymbol{P}_y$ and a diagonal matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{d\times d}$, with $\Lambda_{ii} = \rho_i$.

3. LEARNING THE LOCAL METRIC

3.1. Linear Case

For simplicity, we first describe a linear case, and then extend the results to the general nonlinear case. Consider the case where f and g are two linear functions:

$$\boldsymbol{x} = \boldsymbol{J}_x \begin{bmatrix}\boldsymbol{z}\\ \boldsymbol{\epsilon}\end{bmatrix},\ \boldsymbol{y} = \boldsymbol{J}_y \begin{bmatrix}\boldsymbol{z}\\ \boldsymbol{\eta}\end{bmatrix} \tag{1}$$

where $\boldsymbol{J}_x \in \mathbb{R}^{d_x\times(d_z+d_\epsilon)}$ and $\boldsymbol{J}_y \in \mathbb{R}^{d_y\times(d_z+d_\eta)}$. Note that the assumption $d_z + d_\epsilon \le d_x$, $d_z + d_\eta \le d_y$ entails that the set of equations (1) are over determined. Applying CCA to the random vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ results in the following matrices:

$$\boldsymbol{P}_x = \left(\begin{bmatrix}\boldsymbol{\Phi}_z & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Phi}_\epsilon\end{bmatrix}\boldsymbol{J}_x^\dagger\right)^T,\ \boldsymbol{P}_y = \left(\begin{bmatrix}\boldsymbol{\Phi}_z & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Phi}_\eta\end{bmatrix}\boldsymbol{J}_y^\dagger\right)^T \tag{2}$$

$$\boldsymbol{\Lambda} = \begin{bmatrix}\boldsymbol{I}_{d_z} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix} \qquad \boldsymbol{\Lambda} \in \mathbb{R}^{d\times d} \tag{3}$$

where $\boldsymbol{\Phi}_z, \boldsymbol{\Phi}_\epsilon, \boldsymbol{\Phi}_\eta$ are arbitrary unitary matrices, and $\boldsymbol{J}_x^\dagger$ and $\boldsymbol{J}_y^\dagger$ are the Moore-Penrose pseudo inverse of $\boldsymbol{J}_x$ and $\boldsymbol{J}_y$, respectively, i.e., $\boldsymbol{J}_x^\dagger\boldsymbol{J}_x = \boldsymbol{I}_{(d_z+d_\epsilon)\times(d_z+d_\epsilon)}$ and $\boldsymbol{J}_y^\dagger\boldsymbol{J}_y = \boldsymbol{I}_{\left(d_z+d_\eta\right)\times\left(d_z+d_\eta\right)}$.

Proposition 1. *In the linear case, the Euclidean distance between any two realizations $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ of the random variable $\boldsymbol{z}$ is given by:*

$$\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2 \quad = \quad (\boldsymbol{x}_i - \boldsymbol{x}_j)^T\boldsymbol{P}_x\boldsymbol{\Lambda}\boldsymbol{P}_x^T(\boldsymbol{x}_i - \boldsymbol{x}_j)$$

Proof.

$$\begin{aligned}\Delta\boldsymbol{x}^T\boldsymbol{P}_x\boldsymbol{\Lambda}\boldsymbol{P}_x^T\Delta\boldsymbol{x} &= \left\|\boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{P}_x^T\Delta\boldsymbol{x}\right\|_2^2\\ &= \left\|\begin{bmatrix}\boldsymbol{I} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{\Phi}_z & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{\Phi}_\epsilon\end{bmatrix}\boldsymbol{J}_x^\dagger\boldsymbol{J}_x\begin{bmatrix}\Delta\boldsymbol{z}\\ \Delta\boldsymbol{\epsilon}\end{bmatrix}\right\|_2^2\\ &= \left\|\begin{bmatrix}\boldsymbol{\Phi}_z & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{z}\\ \Delta\boldsymbol{\epsilon}\end{bmatrix}\right\|_2^2 = \|\Delta\boldsymbol{z}\|_2^2\end{aligned}$$

where $\Delta\boldsymbol{x} = \boldsymbol{x}_i - \boldsymbol{x}_j$, $\Delta\boldsymbol{z} = \boldsymbol{z}_i - \boldsymbol{z}_j$, and $\Delta\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_i - \boldsymbol{\epsilon}_j$. □

Note that the matrix $\boldsymbol{\Lambda}$ filters out the sensor-specific variables $\boldsymbol{\epsilon}$, resulting in a metric that takes into account *only* the common hidden variables $\boldsymbol{z}$. The Euclidean distance between realizations of $\boldsymbol{z}$ can be expressed in an analogous manner based on realizations of $\boldsymbol{y}$.

3.2. Non-Linear Case

We now consider the case where $f(\boldsymbol{z},\boldsymbol{\epsilon})$ and $g(\boldsymbol{z},\boldsymbol{\eta})$ are nonlinear and expand f via its Taylor series around some point $\boldsymbol{v}_j \triangleq \begin{bmatrix}\boldsymbol{z}_j^T & \boldsymbol{\epsilon}_j^T\end{bmatrix}^T$:

$$\boldsymbol{x}_i = \boldsymbol{x}_j + \boldsymbol{J}_x(\boldsymbol{v}_j)[\boldsymbol{v}_i - \boldsymbol{v}_j] + \mathcal{O}\left(\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2\right)$$

where $\boldsymbol{x}_j = f(\boldsymbol{v}_j)$. Consider a local neighborhood around $\boldsymbol{x}_j$ where the second and higher order terms are negligible. Consequently, confined to such a neighborhood, we can locally view f as linear and the linear part (Jacobian) $\boldsymbol{J}_x(\boldsymbol{v}_j)$ is equivalent to the linear function considered in Section 3.1. Define the matrices $\boldsymbol{P}_x(\boldsymbol{x}_j)$ and $\boldsymbol{\Lambda}(\boldsymbol{x}_j)$ similarly to (2) and (3) with $\boldsymbol{J}_x(\boldsymbol{v}_j)$ being the linear function.

Proposition 2. *In the nonlinear case, the Euclidean distance between any two realizations $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ of the random variable $\boldsymbol{z}$ is given by:*

$$\begin{aligned}\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2 = \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T[\boldsymbol{A}(\boldsymbol{x}_i) + \boldsymbol{A}(\boldsymbol{x}_j)](\boldsymbol{x}_i - \boldsymbol{x}_j)\\ + \mathcal{O}\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^4\right)\end{aligned} \tag{4}$$

where $\boldsymbol{A}(\boldsymbol{x}_i) \triangleq \boldsymbol{P}_x(\boldsymbol{x}_i)\boldsymbol{\Lambda}(\boldsymbol{x}_i)\boldsymbol{P}_x^T(\boldsymbol{x}_i)$.

In Section 3.3, we show that the matrices $\boldsymbol{A}(\boldsymbol{x}_i)$, and hence, the first term in the right hand side of (4) can be computed from the given data sets. Similarly to Proposition 1, Proposition 2 can be analogously formulated based on realizations of $\boldsymbol{y}$.

Proof. In the proof of Proposition 1 we show that one can write the common variables $\boldsymbol{z}$ up to some rotation by $\boldsymbol{\Phi}_z\boldsymbol{z} = \tilde{\boldsymbol{P}}_x^T\boldsymbol{x}$ where $\tilde{\boldsymbol{P}}_x$ are the d_z leftmost columns of $\boldsymbol{P}_x$. Notice that since $\boldsymbol{\Phi}_z$ is unitary $\|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2 = \|\boldsymbol{\Phi}_z\boldsymbol{z}_i - \boldsymbol{\Phi}_z\boldsymbol{z}_j\|_2^2$. Thus, since the norm is invariant to rotation, without loss of generality, we can recover $\boldsymbol{z}$ up to rotation. With a slight abuse of notation, let f^{-1} denote the local inverse function of f restricted to $\boldsymbol{z}$. By expanding the k-th entry of f^{-1} via its Taylor series around the point $\boldsymbol{x}_j$, where the linear part is given by $\boldsymbol{z}_i = \boldsymbol{z}_j + \tilde{\boldsymbol{P}}_x^T(\boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)$ and $\boldsymbol{z}_j = f^{-1}(\boldsymbol{x}_j)$, we have:

$$\begin{aligned}(z_i)_k = (z_j)_k + \left(\boldsymbol{p}_x^{(k)}(\boldsymbol{x}_j)\right)^T(\boldsymbol{x}_i - \boldsymbol{x}_j)\\ + (\boldsymbol{x}_i - \boldsymbol{x}_j)^T\boldsymbol{H}^{(k)}(\boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j) + \mathcal{O}\left(\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^3\right)\end{aligned} \tag{5}$$

where $\left(\boldsymbol{p}_x^{(k)}(\boldsymbol{x})\right)$ is the k-th column of $\boldsymbol{P}_x(\boldsymbol{x})$, and $\boldsymbol{H}^{(k)}(\boldsymbol{x})$ is the Hessian of the k-th entry of f^{-1}. Expanding $(z_j)_k$ around $\boldsymbol{x}_i$ in the same manner yields:

$$\begin{aligned}(z_j)_k = (z_i)_k + \left(\boldsymbol{p}_x^{(k)}(\boldsymbol{x}_i)\right)^T(\boldsymbol{x}_j - \boldsymbol{x}_i)\\ + (\boldsymbol{x}_j - \boldsymbol{x}_i)^T\boldsymbol{H}^{(k)}(\boldsymbol{x}_i)(\boldsymbol{x}_j - \boldsymbol{x}_i) + \mathcal{O}\left(\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^3\right)\end{aligned} \tag{6}$$

Averaging (5) and (6) and summing over all the elements result in (4). □

3.3. Implementation

Based on Proposition 2 we define a pairwise metric for the N realizations in $\mathcal{X}$.

Definition 3. Let D_{ij} be the metric between each pair of realizations $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, given by:

$$D_{ij} \triangleq \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T[\boldsymbol{A}(\boldsymbol{x}_i) + \boldsymbol{A}(\boldsymbol{x}_j)](\boldsymbol{x}_i - \boldsymbol{x}_j) \tag{7}$$

We can also define an analogous metric between any two realizations $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ in $\mathcal{Y}$.

In the remainder of this section, we discuss the computation of the matrices $\boldsymbol{A}(\boldsymbol{x}_i)$ from $\mathcal{X}$ and $\mathcal{Y}$. Proposition 2 implies that $\boldsymbol{A}(\boldsymbol{x}_i)$ can be computed from the matrices $\boldsymbol{P}_x(\boldsymbol{x}_i)$ and $\boldsymbol{\Lambda}(\boldsymbol{x}_i)$. In addition, $D_{ij} = \|\boldsymbol{z}_i - \boldsymbol{z}_j\|_2^2$ when the distance $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^4$ is negligible. Let $\mathcal{X}_i \subset \mathcal{X}$ and $\mathcal{Y}_i \subset \mathcal{Y}$ be two subsets of realizations defining a small neighborhood $(\mathcal{X}_i, \mathcal{Y}_i)$ around $(\boldsymbol{x}_i, \boldsymbol{y}_i)$. The neighborhoods can be obtained from data in several ways. For example, if the input data is a time series, then one can consider a time frame of samples as a neighborhood. Another possibility is to find the *k nearest neighbors* of each realization. If $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^4$ between any two realizations in the neighborhood is indeed negligible, applying CCA to the two sets $\mathcal{X}_i$ and $\mathcal{Y}_i$ results in the estimation of $\boldsymbol{P}_x(\boldsymbol{x}_i)$ and $\boldsymbol{\Lambda}(\boldsymbol{x}_i)$ as desired.

Proposition 4. *In the absence of the sensor-specific variables, the metric D_{ij} can be written as*

$$D_{ij} = \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \left[\boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x}_i) + \boldsymbol{\Sigma}_{xx}^{-1}(\boldsymbol{x}_j)\right](\boldsymbol{x}_i - \boldsymbol{x}_j) \quad (8)$$

where $\boldsymbol{\Sigma}_{xx}(\boldsymbol{x}_i)$ is the covariance of the random variable $\boldsymbol{x}$ at the point $\boldsymbol{x}_i$ (noting that the covariance changes from point to point due to the nonlinearity of the observation function f).

In other words, the metric we build based on local applications of CCA, when there are no sensor-specific variables $\boldsymbol{\epsilon} = \boldsymbol{\eta} = \boldsymbol{0}$, is a modified *Mahalanobis distance*, which was presented and analyzed in [12–14] for the purpose of recovering the intrinsic representation from nonlinear observation data.

Proof. In the absence of the sensor-specific variables, the matrices $\boldsymbol{\Lambda}(\boldsymbol{x})$ become the identity, namely, $\boldsymbol{\Lambda}(\boldsymbol{x}) = \boldsymbol{I}$ for all $\boldsymbol{x}$. In addition, a known property of CCA links between the matrix $\boldsymbol{P}_x$ and the covariance matrix $\boldsymbol{\Sigma}_{xx}$ [18]:

$$\boldsymbol{P}_x^T(\boldsymbol{x}) = \boldsymbol{U}^T(\boldsymbol{x})\boldsymbol{\Sigma}_{xx}^{-\frac{1}{2}}(\boldsymbol{x}) \quad (9)$$

where $\boldsymbol{U}(\boldsymbol{x})$ is a unitary matrix. Substituting $\boldsymbol{\Lambda}(\boldsymbol{x}) = \boldsymbol{I}$ and (9) into (4) results in (8). □

4. GLOBAL PARAMETRIZATION

Our main goal eventually is to obtain a parametrization that corresponds to the hidden common variables $\boldsymbol{z}$. The metric D_{ij} (7) only approximates the Euclidean distance between the common variables $\boldsymbol{z}$, and it is restricted to small distances. To obtain a global parametrization from the local metric we use a kernel method, Diffusion Maps [11]. The entire method is presented in Algorithm 1. Note that our implementation of Diffusion Maps is based on a Gaussian kernel $K_{ij} = \exp(-D_{ij}/\sigma)$ which defines a notion of locality. Namely, for σ in the order of D_{ij}, the error term in Proposition 2 becomes negligible due to the exponential decay of the Gaussian kernel. Therefore, by appropriately tuning the value σ, K_{ij} accurately represents an affinity between the common variables.

Algorithm 1 Diffusion Maps of Two Datasets

Input: Two sets of observations $\mathcal{X}$ and $\mathcal{Y}$.
Output: Low dimensional parametrization of the common variables $\boldsymbol{z}$.

1. For each pair of points $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in (\mathcal{X}, \mathcal{Y})$:
   (a) Construct the subsets $\mathcal{X}_i, \mathcal{Y}_i$ by choosing all pairs $(\boldsymbol{x}_j, \boldsymbol{y}_j)$ such that $\boldsymbol{x}_j$ is in the neighborhood of $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ is in the neighborhood of $\boldsymbol{y}_i$.
   (b) Apply (linear) CCA to the sets $\mathcal{X}_i, \mathcal{Y}_i$ and obtain the matrices $\boldsymbol{P}_x(\boldsymbol{x}_i)$ and $\boldsymbol{\Lambda}(\boldsymbol{x}_i)$.
2. For each two observations $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$, construct the affinity metric D_{ij} according to (7).
3. Apply Diffusion Maps:
   (a) Construct the kernel: $K_{ij} = \exp(-D_{ij}/\sigma)$, where σ is set to the median value of $\{D_{ij}\}, \forall i, j$.
   (b) Normalize the kernel $\boldsymbol{M} = \boldsymbol{\Omega}^{-1}\boldsymbol{K}$, where $\boldsymbol{\Omega}$ is a diagonal matrix with $\Omega_{ii} = \sum_j K_{ij}$
   (c) Compute the eigenvectors and eigenvalues of the matrix $\boldsymbol{M}$, i.e., $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{U}^{-1}$.
4. Form the parametrization of $\boldsymbol{z}_i, \forall i = 1, \ldots, N$ using the d_z eigenvectors (the columns of $\boldsymbol{U}$) associated with the largest d_z eigenvalues (without the first trivial one), i.e., $(U_{i1}, \ldots, U_{id_z})^T$ for $i \in 1, \ldots, N$.

5. SIMULATION RESULTS

5.1. Partial Linear Example

First, we demonstrate the ability to approximate the metric D_{ij} when one observation is linear whereas the other is nonlinear. Consider the case where f and g are

$$f(z, \epsilon) = \begin{bmatrix} 2 & 1 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} z \\ \epsilon \end{bmatrix}, g(z, \eta) = \begin{bmatrix} (z + 0.2\eta)\cos(20z) \\ (z + 0.2\eta)\sin(20z) \end{bmatrix}$$

We uniformly sample $N = 800$ triplets of scalars $\{z_i, \epsilon_i, \eta_i\}_{i=1}^N$ from the cubic $[0, 1]^3$ and obtain the two sets $\mathcal{X}$ and $\mathcal{Y}$ from the observation functions f and g, respectively. We apply Step 1 and Step 2 of Algorithm 1 to the data sets. Fig. 1(a) and Fig. 1(b) depict the samples in $\mathcal{X}$ and in $\mathcal{Y}$, respectively, colored by the common variable. In Fig. 1(c) we plot the values of D_{ij} as a function of the true Euclidean distances $\|z_i - z_j\|_2^2$. We observe that D_{ij} is indeed an accurate approximation of the Euclidean distance between the (hidden) common variable z. Moreover, we observe that D_{ij} attains a more accurate approximation as the distance $\|z_i - z_j\|$ is smaller, which coincides with Proposition 2.

5.2. Nonlinear High-dimensional Example

In this simulation we show that Algorithm 1 recovers an accurate parametrization of the common variables underlying two nonlinear high-dimensional observations and outperforms competing methods. We generate two high-dimensional movies of three rotating Chess pieces: a bishop, a knight and a tower. Each movie captures only two pieces: the bishop and the knight in the first, and the knight and the tower in the second. In the movies, each piece is rotating in a constant angular speed; the angular speeds of the bishop, the knight and the tower are $17°, 40°, 66°$ per frame, respectively, as demonstrated in Fig. 2(a). Consequently, the angular speed of the knight is the hidden common variable z, whereas the angular speeds of the bishop and the tower are the hidden sensor-specific variables ϵ and η. In order to illustrate the ability of our method to handle two different modalities, the frames of each movie are projected into different spaces using two sets of random projections. Two consecutive frames and their projections are depicted in Fig. 2. Each set of nonlinear high-dimensional observations $\mathcal{X} = \{\boldsymbol{x}_i\}$ and $\mathcal{Y} = \{\boldsymbol{y}_i\}$ con-

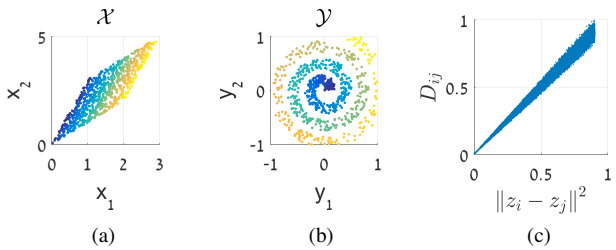

Fig. 1: The two sets of $N = 800$ observations $\mathcal{X}$ and $\mathcal{Y}$: (a) linear, and (b) nonlinear, where the samples are colored by the common variable. (c) shows the *approximation* of the Euclidean distance as a function of the *true* Euclidean distance.

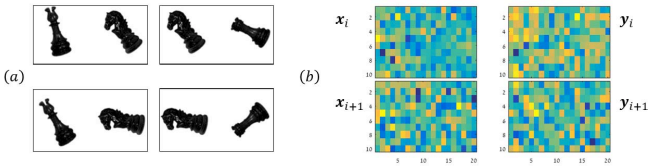

Fig. 2: Two consecutive frames in the movies. (a) The original frames with the Chess pieces. (b) The actual observations (after random projections).

sists of $N = 300$ projected frames. We apply Algorithm 1, where in step $1(a)$, $\mathcal{X}_i$ and $\mathcal{Y}_i$ consist of the projected frames in a time window of length 6 around $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$, respectively. The desired parametrization should convey the fact that the common variable in the sets is the angular speed of the knight, and thus, should be periodic with the same period. To test the obtained parametrization we apply the Fourier transform to the first column of $\boldsymbol{U}$ and present it in Fig. $3(d)$. For comparison, we plot the Fourier transform of the parametrizations obtained by: (a) applying Diffusion Maps to a single movie consisting only of the rotating knight (i.e., when only the common variable is present, without the sensor-specific variables), which serves as a reference, (b) Algorithm 1 with a Euclidean distance, and (c) the KCCA algorithm. Note that the application of Algorithm 1 with the Euclidean distance is simply an application of Diffusion Maps to only one of the movies. In all the figures, the true frequencies of the knight, the bishop, and the tower are marked by vertical red, green and blue lines, respectively.

In Fig. $3(a)$ we see that Diffusion Maps employed on the reference movie identifies the frequency of the knight along with its higher harmonics, whereas the frequencies of the sensor-specific bishop and tower are completely missing, as expected. Fig. $3(b)$ shows that using the Euclidean distance does not identify correctly the common variable nor the sensor-specific variables. In Fig. $3(c)$ we see that KCCA attenuates the sensor-specific variables, however, it captures mostly the second harmonic of the common variable. Finally, in Fig. $3(d)$ we observe that our method captures the true frequency of the common variable while attenuating the sensor-specific variables, in a similar manner to the reference result in Fig. $3(a)$. In summary, without assuming prior knowledge on the structure and content of the data, our method successfully discovers the common variable hidden in high-dimensional and nonlinear observations.

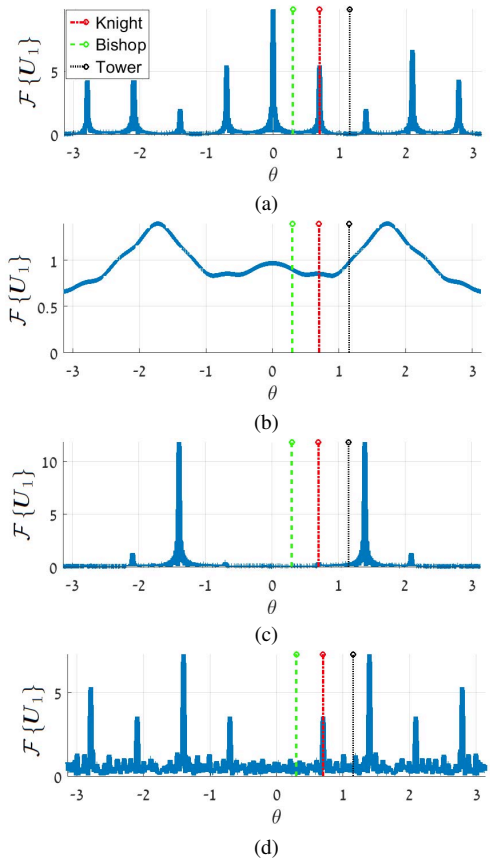

Fig. 3: *(a)* The parametrizations of the common variable obtained by: (a) applying Diffusion Maps to a movie consisting only of the rotating knight, (b) Algorithm 1 with a Euclidean distance, (c) the KCCA algorithm, and (d) our method with the local CCA.

6. CONCLUSIONS

In this paper, we have presented a new manifold learning method for extracting the common hidden variables underlying two multimodal data sets. Our method relies on a local CCA-based metric, which is learned from data in an unsupervised manner. Simulation results show that our method accurately recovers the common variables hidden in two sets of complex, high-dimensional, and multimodal data. Since our method does not require prior rigid model assumptions, it can be applied to a broad variety of multimodal data sets lacking definitive models. Future work will address such applications as well as the extension of this method to more than two sets.

7. REFERENCES

[1] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: an overview of methods, challenges, and prospects," *Proc. IEEE*, vol. 103, no. 9, pp. 1449–1477, 2015.

[2] H. Hotelling, "Relations Between Two Sets of Variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, Dec. 1936.

[3] P. L. Lai and C. Fyfe, "Kernel and nonlinear canonical correlation analysis," *International Journal of Neural Systems*, vol. 10, no. 5, pp. 365–377, 2000.

[4] V. R. de Sa, "Spectral clustering with two views," in *ICML workshop on learning with multiple views*, 2005, pp. 20–27.

[5] B. Wang, J. Jiang, W. Wang, Z.-H. Zhou, and Z. Tu, "Unsupervised metric fusion by cross diffusion," in *IEEE CVPR*, 2012, pp. 2997–3004.

[6] B. Boots and G. Gordon, "Two-manifold problems with applications to nonlinear system identification," in *ICML*, 2012.

[7] R. R. Lederman and R. Talmon, "Learning the geometry of common latent variables using alternating-diffusion," *Appl. Comput. Harmon. Anal.*, 2015.

[8] K. Todros and A. O. Hero, "On measure transformed canonical correlation analysis," *IEEE Transactions on Signal Processing*, vol. 60, no. 9, pp. 4570–4585, 2012.

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

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

[11] R. R. Coifman and S. Lafon, "Diffusion maps," *Appl. Comput. Harmon. Anal.*, vol. 21, no. 1, pp. 5–30, 2006.

[12] A. Singer and R. R. Coifman, "Non-linear independent component analysis with diffusion maps," *Appl. Comput. Harmon. Anal.*, vol. 25, no. 2, pp. 226 – 239, 2008.

[13] R. Talmon and R. R. Coifman, "Empirical intrinsic geometry for nonlinear modeling and time series filtering," *Proceedings of the National Academy of Sciences*, vol. 110, no. 31, pp. 12 535–12 540, 2013.

[14] ——, "Intrinsic modeling of stochastic dynamical systems using empirical geometry," *Appl. Comput. Harmon. Anal.*, vol. 39, no. 1, pp. 138 – 160, 2015.

[15] T. Berry and T. Sauer, "Local kernels and the geometric structure of data," *Appl. Comput. Harmon. Anal.*, 2015.

[16] M. Davenport, C. Hegde, M. Duarte, and R. Baraniuk, "Joint manifolds for data fusion," *IEEE Transactions on Image Processing*, vol. 19, no. 10, pp. 2580–2594, Oct 2010.

[17] Y. Keller, R. Coifman, S. Lafon, and S. Zucker, "Audio-visual group recognition using diffusion maps," *IEEE Transactions on Signal Processing*, vol. 58, no. 1, pp. 403–413, Jan 2010.

[18] L. M. Ewerbring and F. T. Luk, "Canonical correlations and generalized SVD: applications and new algorithms," in *32nd Annual Technical Symposium*, 1989, pp. 206–222.