IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.37, no.9, pp.1764–1776, 2015.

Cross-Domain Matching with Squared-Loss Mutual Information

Makoto Yamada
Yahoo Labs, USA
makotoy@yahoo-inc.com

Leonid Sigal
Disney Research Pittsburgh, USA
lsigal@disneyresearch.com

Michalis Raptis
Comcast Labs, USA
mraptis@cs.ucla.edu

Machiko Toyoda
NTT Labs, Japan
toyoda.machiko@lab.ntt.co.jp

Yi Chang
Yahoo Labs, USA
yichang@yahoo-inc.com

Masashi Sugiyama
The University of Tokyo, Japan
sugi@k.u-tokyo.ac.jp

Abstract

The goal of *cross-domain matching* (CDM) is to find correspondences between two sets of objects in different domains in an unsupervised way. CDM has various interesting applications, including photo album summarization where photos are automatically aligned into a designed frame expressed in the Cartesian coordinate system, and *temporal alignment* which aligns sequences such as videos that are potentially expressed using different features. In this paper, we propose an information-theoretic CDM framework based on *squared-loss mutual information* (SMI). The proposed approach can directly handle non-linearly related objects/sequences with different dimensions, with the ability that hyper-parameters can be objectively optimized by cross-validation. We apply the proposed method to several real-world problems including image matching, unpaired voice conversion, photo album summarization, cross-feature video and cross-domain video-to-mocap alignment, and *Kinect*-based action recognition, and experimentally demonstrate that the proposed method is a promising alternative to state-of-the-art CDM methods.

Keywords

Cross-Domain Object Matching, Cross-Domain Temporal Alignment, Squared-Loss Mutual Information.

1 Introduction

Matching/alignment of objects/time-series from different domains is an important task in machine learning, data mining, and computer vision communities. Applications include photo album summarization, cross-feature video and cross-domain video-to-mocap alignment, activity recognition, temporal segmentation, and curve matching [1, 2, 3, 4, 5, 6]. In this paper, we propose a general information-theoretic cross-domain matching (CDM) framework based on *squared-loss mutual information* [7]. In particular, we address two CDM problems: *cross-domain object matching* and *cross-domain temporal alignment*. The difference between the two CDM problems is subtle. In *object matching* the relative ordering within the sets does not matter, where as in *temporal alignment* the relative ordering within each set must be preserved.

Cross-Domain Object Matching (CDOM): The objective of *cross-domain object matching* (CDOM) is to match two sets of objects in different domains. For instance, in photo album summarization, photos are automatically assigned into a designed frame expressed in the Cartesian coordinate system (see Figure 5(a)). A typical approach of CDOM is to find a mapping from objects in one domain (photos) to objects in the other domain (frame) so that the pairwise dependency is maximized. In this scenario, accurately evaluating the dependence between objects is the key challenge.

Kernelized sorting (KS) [1] tries to find a mapping between two domains that maximizes *mutual information* (MI) [8] under the Gaussian assumption. However, since the Gaussian assumption may not be fulfilled in practice, this method (which we refer to as KS-MI) tends to perform poorly. To overcome the limitation of KS-MI, Quadrianto et al. [2] proposed using the kernel-based dependence measure called the *Hilbert-Schmidt independence criterion* (HSIC) [9] for KS. Since HSIC is a distribution-free independence measure, KS with HSIC (which we refer to as KS-HSIC) is more flexible than KS-MI. However, HSIC includes the Gaussian kernel width as a tuning parameter, and its choice is crucial in obtaining desired performance (see also [10]).

In this paper, we propose an alternative CDOM method that can naturally address the model selection problem. The proposed method, called *least-squares object matching* (LSOM), employs *squared-loss mutual information* (SMI) [7] as the dependence measure. An advantage of LSOM is that cross-validation (CV) with respect to the SMI criterion is possible. Thus, all the tuning parameters such as the Gaussian kernel width and the regularization parameter can be objectively determined by CV. Through experiments on image matching, unpaired voice conversion, and photo album summarization tasks, LSOM is shown to be a promising alternative to CDOM, outperforming competing methods.

Cross-Domain Temporal Alignment (CDTA): Temporal alignment of sequences is an important problem with many practical applications such as speech recognition [11, 12], activity recognition [4], temporal segmentation [5], curve matching [6], chromatographic and micro-array data analysis [13], synthesis of human motion [14], and temporal alignment of human motion [3, 15].

Dynamic time warping (DTW) is a classical temporal alignment method that aligns two sequences by minimizing the pairwise distance [11, 12] between samples (e.g., under the Euclidean, squared Euclidean, or Manhattan distance measures). An advantage of DTW is that the minimization can be efficiently carried out by *dynamic programming* (DP). [16]. However, due to the typical fixed sample-wise notion of distance, DTW may not be able to find a good alignment where two signals are related in complex ways (e.g., a video and negative of the video are perceptually similar but would result in large sample-to-sample distance and DTW score). Moreover, DTW cannot handle sequences with different dimensions (e.g., video to audio alignment), which limits the range of applications significantly. Even if the dimensionality is the same, it is not clear which distance measure is the most appropriate for a given application.

To overcome the weaknesses of DTW, *canonical time warping* (CTW) was introduced in [3]. CTW performs sequence alignment in a common latent space found by canonical correlation analysis (CCA) [17]. Thus, CTW can naturally handle sequences with different dimensions. However, CTW can only deal with linear subspace projections, and it is difficult to optimize model parameters, such as the regularization parameter used in CCA and the dimensionality of the common latent space. To handle non-linearity, *dynamic manifold temporal warping* (DMTW) was recently proposed in [4]. DMTW first projects original data onto a one-dimensional non-linear manifold and then finds an alignment on this manifold using DTW. Although DMTW is highly flexible by construction, its performance depends heavily on the choice of the non-linear transformation and, moreover, it implicitly assumes the smoothness of sequences.

In this paper, we propose a novel information-theoretic CDTA method based on dependence maximization. Our method, which we call *least-squares dynamic time warping* (LSDTW), employs SMI as a dependency measure. Our method can naturally deal with non-linearity and non-Gaussianity in data and CV is available for model selection. Furthermore, LSDTW does not require strong assumptions on the topology of the latent manifold (e.g., smoothness). Thus, LSDTW is expected to perform well in a broader range of applications. Through experiments on synthetic data, video sequence alignment, and *Kinect* action recognition tasks, LSDTW is shown to be a promising alternative to existing temporal alignment methods.

Preliminary version of this work appeared in [18] which only focused on SMI-based CDOM. In this journal version, we further explore SMI-based CDTA and provide a more extensive experimental evaluation.

2 Squared-Loss Mutual Information

We first review squared-loss mutual information (SMI) [7].

SMI is defined and expressed as

$$\begin{aligned}\mathrm{SMI} &= \frac{1}{2}\iint\left(\frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}-1\right)^2 p(\boldsymbol{x})p(\boldsymbol{y})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}\\ &= \frac{1}{2}\iint\left(\frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}\right)p(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}-\frac{1}{2}. \end{aligned} \tag{1}$$

Note that SMI is the *Pearson divergence* [19] from $p(\boldsymbol{x},\boldsymbol{y})$ to $p(\boldsymbol{x})p(\boldsymbol{y})$, while the ordinary MI is the *Kullback-Leibler divergence* [20] from $p(\boldsymbol{x},\boldsymbol{y})$ to $p(\boldsymbol{x})p(\boldsymbol{y})$. SMI is non-negative and takes zero if and only if $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent, as the ordinary MI.

SMI cannot be directly computed since it contains unknown densities $p(\boldsymbol{x},\boldsymbol{y})$, $p(\boldsymbol{x})$, and $p(\boldsymbol{y})$. Here, we briefly review an SMI estimation method called *least-squares mutual information* (LSMI) [7].

Suppose that we are given n independent and identically distributed (i.i.d.) paired samples $\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^n$ drawn from a joint distribution with density $p(\boldsymbol{x},\boldsymbol{y})$. A key idea of LSMI is to directly estimate the *density ratio*,

$$r(\boldsymbol{x},\boldsymbol{y}) = \frac{p(\boldsymbol{x},\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})},$$

without going through density estimation of $p(\boldsymbol{x},\boldsymbol{y})$, $p(\boldsymbol{x})$, and $p(\boldsymbol{y})$.

In LSMI, the density ratio function $r(\boldsymbol{x},\boldsymbol{y})$ is directly modeled by the following linear-in-parameter model:

$$r_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y}) = \sum_{\ell=1}^{b}\alpha_\ell\varphi_\ell(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{\alpha}^\top\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y}), \tag{2}$$

where b is the number of basis functions, $\boldsymbol{\alpha} = (\alpha_1,\ldots,\alpha_b)^\top$ are parameters, $\boldsymbol{\varphi}(\boldsymbol{x},\boldsymbol{y}) = (\varphi_1(\boldsymbol{x},\boldsymbol{y}),\ldots,\varphi_b(\boldsymbol{x},\boldsymbol{y}))^\top$ are basis functions, and $^\top$ denotes the transpose. Here, we use the *product kernel* of the following form for $b = n$ as basis functions:

$$\varphi_\ell(\boldsymbol{x},\boldsymbol{y}) = K(\boldsymbol{x},\boldsymbol{x}_\ell)L(\boldsymbol{y},\boldsymbol{y}_\ell),$$

where $K(\boldsymbol{x},\boldsymbol{x}')$ and $L(\boldsymbol{y},\boldsymbol{y}')$ are reproducing kernels for $\boldsymbol{x}$ and $\boldsymbol{y}$. In this paper, we use the Gaussian kernel.

The parameters $\boldsymbol{\alpha}$ are estimated so that the following squared-error J is minimized:

$$\begin{aligned}J(\boldsymbol{\alpha}) &= \frac{1}{2}\iint(r(\boldsymbol{x},\boldsymbol{y})-r_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y}))^2 p(\boldsymbol{x})p(\boldsymbol{y})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}\\ &= C-\int r_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y})p(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y}\\ &\quad+\frac{1}{2}\iint r_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y})^2 p(\boldsymbol{x})p(\boldsymbol{y})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y},\end{aligned}$$

where we use $r(\boldsymbol{x},\boldsymbol{y})p(\boldsymbol{x})p(\boldsymbol{y}) = p(\boldsymbol{x},\boldsymbol{y})$ and C is a constant.

By using an empirical approximation, the parameter $\boldsymbol{\alpha}$ in the model $r_{\boldsymbol{\alpha}}(\boldsymbol{x},\boldsymbol{y})$ is learned as follows:

$$\widehat{\boldsymbol{\alpha}} = \operatorname*{argmin}_{\boldsymbol{\alpha}} \left[\frac{1}{2}\boldsymbol{\alpha}^\top \widehat{\boldsymbol{H}}\boldsymbol{\alpha} - \widehat{\boldsymbol{h}}^\top\boldsymbol{\alpha} + \frac{\lambda}{2}\boldsymbol{\alpha}^\top\boldsymbol{\alpha}\right], \tag{3}$$

where a regularization term $\lambda\boldsymbol{\alpha}^\top\boldsymbol{\alpha}/2$ is included for avoiding overfitting, and

$$\begin{aligned}
\widehat{\boldsymbol{H}} &= \frac{1}{n^2}\sum_{i,j=1}^{n} \boldsymbol{\varphi}(\boldsymbol{x}_i,\boldsymbol{y}_j)\boldsymbol{\varphi}(\boldsymbol{x}_i,\boldsymbol{y}_j)^\top \\
&= \frac{1}{n^2}(\boldsymbol{K}\boldsymbol{K}^\top)\circ(\boldsymbol{L}\boldsymbol{L}^\top), \\
\widehat{\boldsymbol{h}} &= \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\varphi}(\boldsymbol{x}_i,\boldsymbol{y}_i) \\
&= \frac{1}{n}\left(\boldsymbol{K}\circ\boldsymbol{L}\right)\mathbf{1}_n,
\end{aligned}$$

where $\circ$ denotes the Hadamard product (a.k.a. the element-wise product) and $\mathbf{1}_n = [1,\ldots,1]^\top \in \mathbb{R}^n$.

Differentiating the objective function in Eq.(3) with respect to $\boldsymbol{\alpha}$ and equating it to zero, we can obtain an analytic-form solution:

$$\widehat{\boldsymbol{\alpha}} = (\widehat{\boldsymbol{H}} + \lambda\boldsymbol{I}_b)^{-1}\widehat{\boldsymbol{h}}.$$

Given a density ratio estimator $\widehat{r} = r_{\widehat{\boldsymbol{\alpha}}}$, SMI can be simply approximated as

$$\widehat{\mathrm{SMI}} = \frac{1}{2}\widehat{\boldsymbol{\alpha}}^\top\widehat{\boldsymbol{h}} - \frac{1}{2}. \tag{4}$$

Model selection: In order to determine the kernel parameter and the regularization parameter λ, cross-validation (CV) is available for the SMI estimator: First, the samples $\{(\boldsymbol{x}_i,\boldsymbol{y}_i)\}_{i=1}^n$ are divided into K disjoint subsets $\{\mathcal{S}_k\}_{k=1}^K$, $\mathcal{S}_k = \{(\boldsymbol{x}_{k,i},\boldsymbol{y}_{k,i})\}_{i=1}^{n_k}$ of (approximately) the same size, where n_k is the number of samples in the subset $\mathcal{S}_k$. Then, an estimator $\widehat{\boldsymbol{\alpha}}_{\mathcal{S}_k}$ is obtained using $\{\mathcal{S}_j\}_{j\neq k}$, and the approximation error for the hold-out samples $\mathcal{S}_k$ is computed as

$$J_{\mathcal{S}_k}^{(K\text{-CV})} = \frac{1}{2}\widehat{\boldsymbol{\alpha}}_{\mathcal{S}_k}^\top\widehat{\boldsymbol{H}}_{\mathcal{S}_k}\widehat{\boldsymbol{\alpha}}_{\mathcal{S}_k} - \widehat{\boldsymbol{h}}_{\mathcal{S}_k}^\top\widehat{\boldsymbol{\alpha}}_{\mathcal{S}_k},$$

where, for $[\boldsymbol{K}_{\mathcal{S}_k}]_{ij} = K(\boldsymbol{x}_i,\boldsymbol{x}_{k,j})$, $[\boldsymbol{L}_{\mathcal{S}_k}]_{ij} = L(\boldsymbol{y}_i,\boldsymbol{y}_{k,j})$ $i = 1,\ldots,n, j = 1,\ldots,|\mathcal{S}_k|$,

$$\begin{aligned}
\widehat{\boldsymbol{H}}_{\mathcal{S}_k} &= \frac{1}{n_k^2}(\boldsymbol{K}_{\mathcal{S}_k}\boldsymbol{K}_{\mathcal{S}_k}^\top)\circ(\boldsymbol{L}_{\mathcal{S}_k}\boldsymbol{L}_{\mathcal{S}_k}^\top), \\
\widehat{\boldsymbol{h}}_{\mathcal{S}_k} &= \frac{1}{n_k}\left(\boldsymbol{K}_{\mathcal{S}_k}\circ\boldsymbol{L}_{\mathcal{S}_k}\right)\mathbf{1}_{n_k}.
\end{aligned}$$

This procedure is repeated for $k = 1, \ldots, K$, and its average $J^{(K\text{-CV})}$ is taken as

$$J^{(K\text{-CV})} = \frac{1}{K} \sum_{k=1}^{K} J_{\mathcal{S}_k}^{(K\text{-CV})}.$$

We compute $J^{(K\text{-CV})}$ for all model candidates, and choose the model that minimizes $J^{(K\text{-CV})}$.

3 Cross-Domain Object Matching with SMI

In this section, we propose a CDOM method called *least-squares object matching* (LSOM).

3.1 Overview of Least-Squares Object Matching

The goal of CDOM is, given two sets of samples of the same size, $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{y}_i\}_{i=1}^n$, to find a mapping that well "matches" them. Note that the dimensionality of $\boldsymbol{x}$ and $\boldsymbol{y}$ can be different.

Let π be a permutation function over $\{1, \ldots, n\}$, and let $\boldsymbol{\Pi}$ be the corresponding permutation indicator matrix, i.e.,

$$\boldsymbol{\Pi} \in \{0,1\}^{n \times n}, \ \boldsymbol{\Pi}\mathbf{1}_n = \mathbf{1}_n, \ \text{and} \ \boldsymbol{\Pi}^\top \mathbf{1}_n = \mathbf{1}_n,$$

where $\mathbf{1}_n$ is the n-dimensional vector with all ones.

Let us denote the samples matched by a permutation π by

$$\{(\boldsymbol{x}_i, \boldsymbol{y}_{\pi(i)})\}_{i=1}^n.$$

The optimal permutation, denoted by $\boldsymbol{\Pi}^*$, can be obtained as the maximizer of the SMI between the two sets $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ and $\boldsymbol{Y\Pi} = [\boldsymbol{y}_{\pi(1)}, \ldots, \boldsymbol{y}_{\pi(n)}]$:

$$\boldsymbol{\Pi}^* := \operatorname*{argmax}_{\boldsymbol{\Pi}} \mathrm{SMI}(\boldsymbol{X}, \boldsymbol{Y\Pi}). \tag{5}$$

Based on Eq.(5), we develop the following iterative algorithm for optimizing $\boldsymbol{\Pi}$:

(i) ***Initialization:*** Initialize the alignment matrix $\boldsymbol{\Pi}$.

(ii) ***Dependence estimation:*** For the current $\boldsymbol{\Pi}$, obtain an SMI estimator $\widehat{\mathrm{SMI}}(\boldsymbol{X}, \boldsymbol{Y\Pi})$.

(iii) ***Dependence maximization:*** Given an SMI estimator $\widehat{\mathrm{SMI}}(\boldsymbol{X}, \boldsymbol{Y\Pi})$, obtain the maximum alignment $\boldsymbol{\Pi}$.

(iv) ***Convergence check:*** The above (ii) and (iii) are repeated until $\boldsymbol{\Pi}$ fulfills a convergence criterion.

We call this approach *least-squares object matching* (LSOM).

3.2 Dependence Estimation

In dependence estimation, we compute Eq.(4) with $\boldsymbol{X}$ and $\boldsymbol{Y\Pi}$:

$$\widehat{\mathrm{SMI}}(\boldsymbol{X}, \boldsymbol{Y\Pi}) = \frac{1}{2}\widehat{\boldsymbol{\alpha}}_{\boldsymbol{\Pi}}^{\top}\widehat{\boldsymbol{h}}_{\boldsymbol{\Pi}} - \frac{1}{2}, \tag{6}$$

where

$$\begin{aligned}
\widehat{\boldsymbol{\alpha}}_{\boldsymbol{\Pi}} &= (\widehat{\boldsymbol{H}}_{\boldsymbol{\Pi}} + \lambda \boldsymbol{I}_n)^{-1}\widehat{\boldsymbol{h}}_{\boldsymbol{\Pi}}, \\
\widehat{\boldsymbol{H}}_{\boldsymbol{\Pi}} &= \frac{1}{n^2}(\boldsymbol{K}\boldsymbol{K}^{\top}) \circ (\boldsymbol{\Pi}^{\top}\boldsymbol{L}\boldsymbol{L}^{\top}\boldsymbol{\Pi}), \\
\widehat{\boldsymbol{h}}_{\boldsymbol{\Pi}} &= \frac{1}{n}\left(\boldsymbol{K} \circ (\boldsymbol{\Pi}^{\top}\boldsymbol{L}\boldsymbol{\Pi})\right)\mathbf{1}_n.
\end{aligned}$$

Then, plugging $\widehat{\boldsymbol{h}}_{\boldsymbol{\Pi}}$ into Eq.(6), we get

$$\begin{aligned}
\widehat{\mathrm{SMI}}(\boldsymbol{X}, \boldsymbol{Y\Pi}) &= \frac{1}{2n}\widehat{\boldsymbol{\alpha}}_{\boldsymbol{\Pi}}^{\top}\left(\boldsymbol{K} \circ (\boldsymbol{\Pi}^{\top}\boldsymbol{L}\boldsymbol{\Pi})\right)\mathbf{1}_n - \frac{1}{2} \\
&= \frac{1}{2n}\mathrm{tr}\left(\boldsymbol{\Pi}^{\top}\boldsymbol{L}\boldsymbol{\Pi}\widehat{\boldsymbol{A}}_{\boldsymbol{\Pi}}\boldsymbol{K}\right) - \frac{1}{2},
\end{aligned}$$

where $\widehat{\boldsymbol{A}}$ is the diagonal matrix with diagonal elements given by $\widehat{\boldsymbol{\alpha}}$. Note that we used Eq.(73) and Eq.(75) in [21] for obtaining the above SMI expression. Note, we use the model selection presented in Section 2.

3.3 Dependence Maximization

Dependence maximization of LSOM is formulated as follows:

$$\max_{\boldsymbol{\Pi}} \ \widehat{\mathrm{SMI}}(\boldsymbol{X}, \boldsymbol{Y\Pi}).$$

Since this optimization problem is in general NP-hard, we simply use the same optimization strategy as kernelized sorting [2] (see also Section 5.1.2), i.e., for the current $\boldsymbol{\Pi}^{\mathrm{old}}$, the solution is updated as

$$\boldsymbol{\Pi}^{\mathrm{new}} = (1-\eta)\boldsymbol{\Pi}^{\mathrm{old}} + \eta \operatorname*{argmax}_{\boldsymbol{\Pi}} \ \mathrm{tr}\left(\boldsymbol{\Pi}^{\top}\boldsymbol{L}\boldsymbol{\Pi}^{\mathrm{old}}\widehat{\boldsymbol{A}}_{\boldsymbol{\Pi}^{\mathrm{old}}}\boldsymbol{K}\right). \tag{7}$$

where $0 < \eta \leq 1$ is a step size. The second term is a *linear assignment problem* (LAP) [22], which can be efficiently solved by using the *Hungarian method* [22]. In this paper, a C++ implementation of the Hungarian method provided by Cooper[1] was used for solving Eq.(7); then $\boldsymbol{\Pi}$ is repeatedly updated by Eq.(7) until convergence.

Initialization: In this iterative optimization procedure, the choice of the initial permutation matrix is critical to obtain a good solution. Quadrianto et al. [2] proposed a

[1] http://mit.edu/harold/www/code.html

HSIC-based initialization scheme. HSIC is a kernel-based dependence measure given as follows [9]:

$$\mathrm{HSIC}(\boldsymbol{X}, \boldsymbol{Y}) = \mathrm{tr}(\bar{\boldsymbol{K}}\bar{\boldsymbol{L}}),$$

where $\bar{\boldsymbol{K}} = \boldsymbol{\Gamma}\boldsymbol{K}\boldsymbol{\Gamma}$ and $\bar{\boldsymbol{L}} = \boldsymbol{\Gamma}\boldsymbol{L}\boldsymbol{\Gamma}$ are the centered kernel matrices for $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. Note that smaller HSIC scores mean that X and Y are closer to be independent. In the HSIC-based initialization scheme, the alignment that maximizes HSIC between X and Y is used.

Suppose that the kernel matrices $\bar{\boldsymbol{K}}$ and $\bar{\boldsymbol{L}}$ are rank one, i.e., for some $\boldsymbol{f}$ and $\boldsymbol{g}$, $\bar{\boldsymbol{K}}$ and $\bar{\boldsymbol{L}}$ can be expressed as $\bar{\boldsymbol{K}} = \boldsymbol{f}\boldsymbol{f}^\top$ and $\bar{\boldsymbol{L}} = \boldsymbol{g}\boldsymbol{g}^\top$. Then HSIC can be written as

$$\mathrm{HSIC}(\boldsymbol{X}, \boldsymbol{Y}\boldsymbol{\Pi}) = \|\boldsymbol{f}^\top\boldsymbol{\Pi}\boldsymbol{g}\|^2. \tag{8}$$

The initial permutation matrix is determined so that Eq.(8) is maximized. According to Theorems 368 and 369 in [23], the maximum of Eq.(8) is attained when the elements of $\boldsymbol{f}$ and $\boldsymbol{\Pi}\boldsymbol{g}$ are ordered in the same way. That is, if the elements of $\boldsymbol{f}$ are ordered in the ascending manner (i.e., $f_1 \leq f_2 \leq \cdots \leq f_n$), the maximum of Eq.(8) is attained by ordering the elements of $\boldsymbol{g}$ in the same ascending way. However, since the kernel matrices $\bar{\boldsymbol{K}}$ and $\bar{\boldsymbol{L}}$ may not be rank one in practice, the principal eigenvectors of $\bar{\boldsymbol{K}}$ and $\bar{\boldsymbol{L}}$ were used as $\boldsymbol{f}$ and $\boldsymbol{g}$ in [2]. We call this *eigenvalue-based initialization*.

4 Cross Domain Temporal Alignment via SMI

Next, we propose *cross-domain temporal alignment* (CDTA) based on SMI [7, 24]. The key difference between temporal alignment and object matching is that sample ordering within each set must be strictly preserved in temporal alignment, as that accounts for the temporal order of samples.

4.1 Overview of Least-Squares Dynamic Time Warping (LS-DTW)

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{n_\mathrm{x}}]$ and $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{n_\mathrm{y}}]$ be sequences, represented by ordered samples $\boldsymbol{x}_i \in \mathbb{R}^{d_\mathrm{x}}$ and $\boldsymbol{y}_i \in \mathbb{R}^{d_\mathrm{y}}$, from different domains. Our goal is to find temporal alignment such that the statistical dependency between two sequences of samples is maximized. Note that n_x and d_x can, in general, be different from n_y and d_y.

Let $\boldsymbol{\pi}^\mathrm{x}$ and $\boldsymbol{\pi}^\mathrm{y}$ be alignment functions over $\{1, \ldots, n_\mathrm{x}\}$ and $\{1, \ldots, n_\mathrm{y}\}$, and let $\boldsymbol{\Pi}$ be the corresponding alignment matrix:

$$\begin{aligned}
\boldsymbol{\Pi} &:= [\boldsymbol{\pi}^\mathrm{x}\ \boldsymbol{\pi}^\mathrm{y}]^\top \in \mathbb{R}^{2\times m},\\
\boldsymbol{\pi}^\mathrm{x} &:= [\pi^\mathrm{x}_1, \ldots, \pi^\mathrm{x}_m]^\top \in \{1, \ldots, n_\mathrm{x}\}^{m\times 1},\\
\boldsymbol{\pi}^\mathrm{y} &:= [\pi^\mathrm{y}_1, \ldots, \pi^\mathrm{y}_m]^\top \in \{1, \ldots, n_\mathrm{y}\}^{m\times 1},
\end{aligned}$$

where m is the number of indexes needed to align the sequences and $^\top$ denotes the transpose. $\boldsymbol{\Pi}$ needs to satisfy the following constraints:

- **Boundary condition:** $[\pi_1^{\mathrm{x}}\ \pi_1^{\mathrm{y}}]^\top = [1\ 1]^\top$ and $[\pi_m^{\mathrm{x}}\ \pi_m^{\mathrm{y}}]^\top = [n_{\mathrm{x}}\ n_{\mathrm{y}}]^\top$.
- **Continuity condition:** $\pi_t^{\mathrm{x}} - \pi_{t-1}^{\mathrm{x}} \in \{0,1\}$ and $\pi_t^{\mathrm{y}} - \pi_{t-1}^{\mathrm{y}} \in \{0,1\}$.

Note that the continuity condition implies monotonicity: $t_1 \geq t_2 \Rightarrow \pi_{t_1}^{\mathrm{x}} \geq \pi_{t_2}^{\mathrm{x}}, \pi_{t_1}^{\mathrm{y}} \geq \pi_{t_2}^{\mathrm{y}}$.

Let us denote the aligned sequences by using $\boldsymbol{\pi}^{\mathrm{x}}$ and $\boldsymbol{\pi}^{\mathrm{y}}$ as

$$\boldsymbol{X}_{\boldsymbol{\pi}^{\mathrm{x}}} = [\boldsymbol{x}_{\pi_1^{\mathrm{x}}}, \boldsymbol{x}_{\pi_2^{\mathrm{x}}}, \ldots, \boldsymbol{x}_{\pi_m^{\mathrm{x}}}],$$
$$\boldsymbol{Y}_{\boldsymbol{\pi}^{\mathrm{y}}} = [\boldsymbol{y}_{\pi_1^{\mathrm{y}}}, \boldsymbol{y}_{\pi_2^{\mathrm{y}}}, \ldots, \boldsymbol{y}_{\pi_m^{\mathrm{y}}}].$$

Then, the optimal alignment, denoted by $\boldsymbol{\Pi}^*$, is defined as the maximizer of SMI between the two sequences $\boldsymbol{X}_{\boldsymbol{\pi}^{\mathrm{x}}}$ and $\boldsymbol{Y}_{\boldsymbol{\pi}^{\mathrm{y}}}$. The optimization problem of LSDTW is defined as

$$\boldsymbol{\Pi}^* := \operatorname*{argmax}_{\boldsymbol{\Pi}} \mathrm{SMI}(\boldsymbol{X}_{\boldsymbol{\pi}^{\mathrm{x}}}, \boldsymbol{Y}_{\boldsymbol{\pi}^{\mathrm{y}}}). \tag{9}$$

Based on Eq.(9), we develop the following iterative algorithm for estimating $\boldsymbol{\Pi}$:

(i) ***Initialization:*** Initialize the alignment matrix $\boldsymbol{\Pi}$.

(ii) ***Dependence estimation:*** For the current $\boldsymbol{\Pi}$, obtain an SMI estimator $\widehat{\mathrm{SMI}}(\boldsymbol{X}_{\boldsymbol{\pi}^{\mathrm{x}}}, \boldsymbol{Y}_{\boldsymbol{\pi}^{\mathrm{y}}})$.

(iii) ***Dependence maximization:*** Given an SMI estimator $\widehat{\mathrm{SMI}}(\boldsymbol{X}_{\boldsymbol{\pi}^{\mathrm{x}}}, \boldsymbol{Y}_{\boldsymbol{\pi}^{\mathrm{y}}})$, obtain the maximum alignment $\boldsymbol{\Pi}$.

(iv) ***Convergence check:*** (ii) and (iii) are repeated until $\boldsymbol{\Pi}$ fulfills a convergence criterion.

We call this method as *least-squares dynamic time warping* (LSDTW).

4.2 Dependence Estimation

In dependence estimation of LSDTW, we compute Eq.(4) from $\boldsymbol{X}_{\boldsymbol{\pi}^{\mathrm{x}}}$ and $\boldsymbol{Y}_{\boldsymbol{\pi}^{\mathrm{y}}}$ as

$$\begin{aligned}\widehat{\mathrm{SMI}}(\boldsymbol{X}_{\boldsymbol{\pi}^{\mathrm{x}}}, \boldsymbol{Y}_{\boldsymbol{\pi}^{\mathrm{y}}}) &= \frac{1}{2}\widehat{\boldsymbol{\alpha}}_{\boldsymbol{\Pi}}^\top \widehat{\boldsymbol{h}}_{\boldsymbol{\Pi}} - \frac{1}{2} \\ &= \frac{1}{2m}\sum_{i=1}^{m} r_{\widehat{\boldsymbol{\alpha}}_{\boldsymbol{\Pi}}}(\boldsymbol{x}_{\pi_i^{\mathrm{x}}}, \boldsymbol{y}_{\pi_i^{\mathrm{y}}}) - \frac{1}{2},\end{aligned} \tag{10}$$

where

$$\begin{aligned} r_{\widehat{\boldsymbol{\alpha}}_{\boldsymbol{\Pi}}}(\boldsymbol{x}, \boldsymbol{y}) &= \widehat{\boldsymbol{\alpha}}_{\boldsymbol{\Pi}}^\top \boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{y}), \\ \widehat{\boldsymbol{\alpha}}_{\boldsymbol{\Pi}} &= (\widehat{\boldsymbol{H}}_{\boldsymbol{\Pi}} + \lambda \boldsymbol{I}_n)^{-1} \widehat{\boldsymbol{h}}_{\boldsymbol{\Pi}}, \\ \widehat{H}_{\boldsymbol{\Pi},\ell,\ell'} &:= \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} K(\boldsymbol{x}_{\pi_i^{\mathrm{x}}}, \boldsymbol{x}_{\pi_\ell^{\mathrm{x}}}) L(\boldsymbol{y}_{\pi_j^{\mathrm{y}}}, \boldsymbol{y}_{\pi_\ell^{\mathrm{y}}}) \\ &\qquad \times K(\boldsymbol{x}_{\pi_i^{\mathrm{x}}}, \boldsymbol{x}_{\pi_{\ell'}^{\mathrm{x}}}) L(\boldsymbol{y}_{\pi_j^{\mathrm{y}}}, \boldsymbol{y}_{\pi_{\ell'}^{\mathrm{y}}}), \\ h_{\boldsymbol{\Pi},\ell} &:= \frac{1}{m}\sum_{i=1}^{m} K(\boldsymbol{x}_{\pi_i^{\mathrm{x}}}, \boldsymbol{x}_{\pi_\ell^{\mathrm{x}}}) L(\boldsymbol{y}_{\pi_i^{\mathrm{y}}}, \boldsymbol{y}_{\pi_\ell^{\mathrm{y}}}). \end{aligned}$$

$\boldsymbol{\varphi}(\boldsymbol{x}, \boldsymbol{y})$ is the basis function (See Eq.(2) for details). We select model parameters of SMI using the approach in Section 2.

4.3 Dependence Maximization

Based on the empirical estimate of SMI given by Eq.(10), the dependence maximization problem is given as

$$\max_{\boldsymbol{\Pi}} \ \widehat{\mathrm{SMI}}(\boldsymbol{X}_{\boldsymbol{\pi}^{\mathrm{x}}}, \boldsymbol{Y}_{\boldsymbol{\pi}^{\mathrm{y}}}).$$

We here provide a computationally efficient approximation algorithm based on *dynamic programming* (DP) [16].

Given the empirical estimate of SMI computed at the dependence estimation step, the dependence maximization problem is given from Eq.(10) as

$$\begin{aligned}&\max_{\boldsymbol{\Pi}} \ \widehat{\mathrm{SMI}}(\boldsymbol{X}_{\boldsymbol{\pi}^{\mathrm{x}}}, \boldsymbol{Y}_{\boldsymbol{\pi}^{\mathrm{y}}})\\ &= \max_{\boldsymbol{\Pi}} \ \frac{1}{2m}\sum_{i=1}^{m}\sum_{\ell=1}^{m_{\mathrm{old}}} \widehat{\alpha}_\ell K(\boldsymbol{x}_{\pi_i^{\mathrm{x}}}, \boldsymbol{x}_{\pi_{\ell\,\mathrm{old}}^{\mathrm{x}}}) L(\boldsymbol{y}_{\pi_i^{\mathrm{y}}}, \boldsymbol{y}_{\pi_{\ell\,\mathrm{old}}^{\mathrm{y}}}).\end{aligned}$$

Based on the constraints on the alignment functions $\boldsymbol{\Pi}$ described in Section 4.1, this optimal alignment can be computed by DP [16]. In order to verify this, we define the prefix sequences $\boldsymbol{X}_n := [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n]$ and $\boldsymbol{Y}_{n'} := [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_{n'}]$, with $n \leq n_{\mathrm{x}}$ and $n' \leq n_{\mathrm{y}}$, and

$$\widehat{\mathrm{SMI}}(\boldsymbol{X}_n, \boldsymbol{Y}_{n'}) =: \mathrm{SMI}(n, n') = A(n, n')/M(n, n').$$

This denotes the optimal SMI for the aligned prefix sequences $\boldsymbol{X}_n$ and $\boldsymbol{Y}_{n'}$, where $A(n, n') := \sum_{i=1}^{m} r_{\widehat{\boldsymbol{\alpha}}_{\boldsymbol{\Pi}_{\mathrm{old}}}}(\boldsymbol{x}_{\pi_i^{\mathrm{x}}}, \boldsymbol{y}_{\pi_i^{\mathrm{y}}})$ and $M(n, n') = m$.

Based on the continuity and monotonicity conditions, the optimal SMI for the aligned prefix sequences $\boldsymbol{X}_n$ and $\boldsymbol{Y}_{n'}$ is computed as

$$\mathrm{SMI}(n, n') = A(n, n')/M(n, n'),$$

$$A(n,n') = \begin{cases} A(n,n'-1)+\gamma_{n,n'}, & (\mu = \mathrm{SMI}(n,n'-1)) \\ A(n-1,n')+\gamma_{n,n'}, & (\mu = \mathrm{SMI}(n-1,n')) \\ A(n-1,n'-1)+\gamma_{n,n'}, & (\mu = \mathrm{SMI}(n-1,n'-1)) \end{cases},$$

$$M(n,n') = \begin{cases} M(n,n'-1)+1, & (\mu = \mathrm{SMI}(n,n'-1)) \\ M(n-1,n')+1, & (\mu = \mathrm{SMI}(n-1,n')) \\ M(n-1,n'-1)+1, & (\mu = \mathrm{SMI}(n-1,n'-1)) \end{cases},$$

$\mu = \max\{\mathrm{SMI}(n-1,n'-1), \mathrm{SMI}(n-1,n'), \mathrm{SMI}(n,n'-1)\}$, $\gamma_{n,n'} = r_{\widehat{\boldsymbol{\alpha}}_{\boldsymbol{\Pi}_{\mathrm{old}}}}(\boldsymbol{x}_n, \boldsymbol{y}_{n'})$, for $1 < n \leq n_{\mathrm{x}}$ and $1 < n' \leq n_{\mathrm{y}}$, where the boundary conditions of the alignment functions is given as follows:

$$\begin{aligned}\mathrm{SMI}(1,1) &= r_{\widehat{\boldsymbol{\alpha}}_{\boldsymbol{\Pi}_{\mathrm{old}}}}(\boldsymbol{x}_1, \boldsymbol{y}_1),\\ A(1,1) &= \mathrm{SMI}(1,1),\\ M(1,1) &= 1.\end{aligned}$$

Therefore, the optimal $\widehat{\mathrm{SMI}}(\boldsymbol{X}_{\boldsymbol{\pi}^{\mathrm{x}}}, \boldsymbol{Y}_{\boldsymbol{\pi}^{\mathrm{y}}}) = \frac{1}{2}A(n_{\mathrm{x}}, n_{\mathrm{y}})/M(n, n') - \frac{1}{2}$ can be computed in time complexity $O(n_{\mathrm{x}}n_{\mathrm{y}})$. Given the accumulated cost matrix $B_{n,n'} = \mathrm{SMI}(n, n')$, we can compute the optimal alignment $\boldsymbol{\Pi}$ using backtracking.

Initialization: Due to the greedy nature of the algorithms, using a good initial alignment is highly important for the success of LSDTW. Here, from the alignment obtained using CTW [3] and the simple uniform initialization,

$$\boldsymbol{\pi}^{\mathrm{x}} = [1, \lfloor 1 + n_{\mathrm{x}}/m \rfloor, \lfloor 1 + 2n_{\mathrm{x}}/m \rfloor, \ldots, n_{\mathrm{x}}]^\top \in \mathbb{R}^{m \times 1},$$
$$\boldsymbol{\pi}^{\mathrm{y}} = [1, \lfloor 1 + n_{\mathrm{y}}/m \rfloor, \lfloor 1 + 2n_{\mathrm{y}}/m \rfloor, \ldots, n_{\mathrm{y}}]^\top \in \mathbb{R}^{m \times 1},$$

where $m = \min(n_{\mathrm{x}}, n_{\mathrm{y}})$ and $\lfloor c \rfloor$ denotes the largest integer not greater than c. Out of the two resulting alignments, one for each initialization, we choose the one with the larger cross-validation score as the final result of LSDTW.

5 Related Methods

In this section, we review related methods for CDOM and CDTA.

5.1 Cross-Domain Object Matching

First, we review relevant CDOM methods and point out their potential weaknesses.

5.1.1 Kernelized Sorting with Mutual Information

Kernelized sorting with mutual information (KS-MI) [1] matches objects in different domains so that MI between matched pairs is maximized. We review KS-MI following the alternative derivation provided in [2].

MI is one of the popular dependence measures between random variables. For random variables $\boldsymbol{X}$ and $\boldsymbol{Y}$, MI is defined as follows [8]:

$$\mathrm{MI}(\boldsymbol{X}, \boldsymbol{Y}) := \iint p(\boldsymbol{x}, \boldsymbol{y}) \log \frac{p(\boldsymbol{x}, \boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})} \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{y},$$

where $p(\boldsymbol{x}, \boldsymbol{y})$ denotes the joint density of $\boldsymbol{x}$ and $\boldsymbol{y}$, and $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ are the marginal densities of $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively.

Now, let us assume that $\boldsymbol{x}$ and $\boldsymbol{y}$ are jointly normal in some reproducing Kernel Hilbert Spaces (RKHSs) endowed with joint kernel $K(\boldsymbol{x}, \boldsymbol{x}')L(\boldsymbol{y}, \boldsymbol{y}')$, where $K(\boldsymbol{x}, \boldsymbol{x}')$ and $L(\boldsymbol{y}, \boldsymbol{y}')$ are reproducing kernels for $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. Then KS-MI is formulated as follows:

$$\min_{\boldsymbol{\Pi}} \log |\boldsymbol{\Gamma}(\boldsymbol{K} \circ (\boldsymbol{\Pi}^\top \boldsymbol{L} \boldsymbol{\Pi}))\boldsymbol{\Gamma}|, \tag{11}$$

where $\boldsymbol{K} = \{K(\boldsymbol{x}_i, \boldsymbol{x}_j)\}_{i,j=1}^n$ and $\boldsymbol{L} = \{L(\boldsymbol{y}_i, \boldsymbol{y}_j)\}_{i,j=1}^n$ are kernel matrices, $\boldsymbol{\Gamma} = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$ is the centering matrix, and $\boldsymbol{I}_n$ is the n-dimensional identity matrix.

A critical weakness of KS-MI is the Gaussian assumption, which may not be fulfilled in practice.

5.1.2 Kernelized Sorting with Hilbert-Schmidt Independence Criterion

Kernelized sorting with Hilbert-Schmidt independence criterion (KS-HSIC) matches objects in different domains so that HSIC between matched pairs is maximized.

HSIC is a kernel-based dependence measure given as follows [9]:

$$\mathrm{HSIC}(\boldsymbol{X}, \boldsymbol{Y}) = \mathrm{tr}(\bar{\boldsymbol{K}}\bar{\boldsymbol{L}}),$$

where $\bar{\boldsymbol{K}} = \boldsymbol{\Gamma}\boldsymbol{K}\boldsymbol{\Gamma}$ and $\bar{\boldsymbol{L}} = \boldsymbol{\Gamma}\boldsymbol{L}\boldsymbol{\Gamma}$ are the centered kernel matrices for $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. Note that the smaller the HSIC score is, the closer $\boldsymbol{X}$ and $\boldsymbol{Y}$ are to be independent.

KS-HSIC is formulated as follows [2]:

$$\max_{\boldsymbol{\Pi}} \mathrm{HSIC}(\boldsymbol{X}, \boldsymbol{Y}\boldsymbol{\Pi}), \tag{12}$$

where

$$\mathrm{HSIC}(\boldsymbol{X}, \boldsymbol{Y}\boldsymbol{\Pi}) = \mathrm{tr}(\bar{\boldsymbol{K}}\boldsymbol{\Pi}^\top\bar{\boldsymbol{L}}\boldsymbol{\Pi}). \tag{13}$$

This optimization problem is called the *quadratic assignment problem* (QAP) [25], and it is known to be *NP-hard*. There exists several QAP solvers based on, e.g., simulated annealing, tabu search, and genetic algorithms. However, those QAP solvers are not easy to use in practice since they contain various tuning parameters.

Another approach to solving Eq.(12) based on a *linear assignment problem* (LAP) [22] was proposed in [2], which is explained below. Let us relax the permutation indicator matrix $\boldsymbol{\Pi}$ to take real values:

$$\boldsymbol{\Pi} \in [0,1]^{n\times n},\ \boldsymbol{\Pi}\mathbf{1}_n = \mathbf{1}_n,\ \text{and}\ \boldsymbol{\Pi}^\top\mathbf{1}_n = \mathbf{1}_n. \tag{14}$$

Then, the update formula of KS-HSIC can be given as [2]

$$\boldsymbol{\Pi}^{\mathrm{new}} = (1-\eta)\boldsymbol{\Pi}^{\mathrm{old}} + \eta \operatorname*{argmax}_{\boldsymbol{\Pi}} \mathrm{tr}\left(\boldsymbol{\Pi}^\top\bar{\boldsymbol{L}}\boldsymbol{\Pi}^{\mathrm{old}}\bar{\boldsymbol{K}}\right), \tag{15}$$

where $0 < \eta \leq 1$ is a step size. The second term is an LAP subproblem, which can be efficiently solved by using the *Hungarian method* [22].

In the original KS-HSIC paper [2], a C++ implementation of the Hungarian method provided by Cooper[1] was used for solving Eq.(15); then $\boldsymbol{\Pi}$ is kept updated by Eq.(15) until convergence.

Since HSIC is a distribution-free dependence measure, KS-HSIC is more flexible than KS-MI. However, a critical weakness of HSIC is that its performance is sensitive to the choice of kernels [10]. A practical heuristic is to use the Gaussian kernel with width set to the median distance between samples [26], but this does not always work well in practice.

[1] http://mit.edu/harold/www/code.html

5.1.3 Kernelized Sorting with Normalized Cross-Covariance Operator

The kernel-based dependence measure based on the *normalized cross-covariance operator* (NOCCO) [27] is given as follows [27]:

$$\mathrm{D}_{\mathrm{NOCCO}}(Z) = \mathrm{tr}(\widetilde{\boldsymbol{K}}\widetilde{\boldsymbol{L}}),$$

where $\widetilde{\boldsymbol{K}} = \bar{\boldsymbol{K}}(\bar{\boldsymbol{K}} + n\epsilon \boldsymbol{I}_n)^{-1}$, $\widetilde{\boldsymbol{L}} = \bar{\boldsymbol{L}}(\bar{\boldsymbol{L}} + n\epsilon \boldsymbol{I}_n)^{-1}$, and $\epsilon > 0$ is a regularization parameter. $\mathrm{D}_{\mathrm{NOCCO}}$ was shown to be asymptotically independent of the choice of kernels. Thus, KS with $\mathrm{D}_{\mathrm{NOCCO}}$ (KS-NOCCO) is expected to be less sensitive to the kernel parameter choice than KS-HSIC [18].

The dependency measure for $Z(\boldsymbol{\Pi})$ can be written as [18]

$$\mathrm{D}_{\mathrm{NOCCO}}(Z(\boldsymbol{\Pi})) = \mathrm{tr}(\widetilde{\boldsymbol{K}}\boldsymbol{\Pi}^\top \widetilde{\boldsymbol{L}}\boldsymbol{\Pi}).$$

Since this is essentially the same form as HSIC, a local optimal solution may be obtained in the same way as KS-HSIC:

$$\boldsymbol{\Pi}^{\mathrm{new}} = (1-\eta)\boldsymbol{\Pi}^{\mathrm{old}} + \eta \operatorname*{argmax}_{\boldsymbol{\Pi}} \mathrm{tr}\left(\boldsymbol{\Pi}^\top \widetilde{\boldsymbol{L}}\boldsymbol{\Pi}^{\mathrm{old}}\widetilde{\boldsymbol{K}}\right). \tag{16}$$

However, the property that $\mathrm{D}_{\mathrm{NOCCO}}$ is independent of the kernel choice holds only asymptotically. Thus, with finite samples, $\mathrm{D}_{\mathrm{NOCCO}}$ does still depend on the choice of kernels as well as the regularization parameter ϵ which needs to be manually tuned.

5.2 Cross-Domain Temporal Alignment

Next, we review relevant temporal alignment methods which are based on pairwise distance minimization (not dependence maximization) and point out their potential weaknesses.

5.2.1 Dynamic Time Warping (DTW)

The goal of *dynamic time warping* (DTW) is, given two sequences of the *same* dimensionality with *different* lengths, $\boldsymbol{X}$ and $\boldsymbol{Y}$, to find an alignment such that the sum of pairwise distances between two sequences is minimized [11, 12]:

$$\min_{\boldsymbol{W}_{\mathrm{x}}, \boldsymbol{W}_{\mathrm{y}}} \|\boldsymbol{X}\boldsymbol{W}_{\mathrm{x}}^\top - \boldsymbol{Y}\boldsymbol{W}_{\mathrm{y}}^\top\|_{\mathrm{Frob}}^2,$$

where $\|\cdot\|_{\mathrm{Frob}}$ is the Frobenius norm, $\boldsymbol{W}_{\mathrm{x}} \in \{0,1\}^{m \times n_{\mathrm{x}}}$ and $\boldsymbol{W}_{\mathrm{y}} \in \{0,1\}^{m \times n_{\mathrm{y}}}$ are binary selection matrices that need to be estimated to align $\boldsymbol{X}$ and $\boldsymbol{Y}$. The above DTW optimization problem can be efficiently solved by DP with time complexity $O(n_{\mathrm{x}} n_{\mathrm{y}})$. However, DTW tends to fail if the magnitude of two sequences are different. To deal with this issue, the *Derivative dynamic time warping* (DDTW) [28], which aligns the first order derivative of sequences, is useful.

Potential weaknesses of DTW and DDTW are that they cannot handle sequences with different dimensionalities such as image-to-audio alignment. Moreover, even when the dimensionality of the sequences is the same, DTW and DDTW may not be able to find a good alignment of sequences with different characteristics such as sequences with different amplitudes. These drawbacks highly limit the applicability of DTW and DDTW.

5.2.2 Iterative Motion Warping (IMW)

The optimization problem of the *iterative motion warping* (IWM) [29] is given as

$$\min_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{W}} \|(\boldsymbol{X} \circ \boldsymbol{A}_{\mathrm{x}} + \boldsymbol{B}_{\mathrm{x}})\boldsymbol{W}_{\mathrm{x}}^{\top} - (\boldsymbol{Y} \circ \boldsymbol{A}_{\mathrm{y}} + \boldsymbol{B}_{\mathrm{y}})\boldsymbol{W}_{\mathrm{y}}^{\top}\|_{\mathrm{Frob}}^2 + R(\boldsymbol{A}_{\mathrm{x}}, \boldsymbol{A}_{\mathrm{y}}, \boldsymbol{B}_{\mathrm{x}}, \boldsymbol{B}_{\mathrm{y}}),$$

where $\boldsymbol{A}_{\mathrm{x}} \in \mathbb{R}^{d \times n_{\mathrm{x}}}$ and $\boldsymbol{A}_{\mathrm{y}} \in \mathbb{R}^{d \times n_{\mathrm{y}}}$ are the scaling matrices, $\boldsymbol{B}_{\mathrm{x}} \in \mathbb{R}^{d \times n_{\mathrm{x}}}$ and $\boldsymbol{B}_{\mathrm{y}} \in \mathbb{R}^{d \times n_{\mathrm{x}}}$ are the translation matrices, $\circ$ is the Hadamard product, $R(\boldsymbol{A}_{\mathrm{x}}, \boldsymbol{A}_{\mathrm{y}}, \boldsymbol{B}_{\mathrm{x}}, \boldsymbol{B}_{\mathrm{y}})$ is the regularization term to avoid overfitting.

IMW can successfully deal with sequences with different characteristics, e.g., having different amplitudes. However, similarly to the original DTW, IMW cannot handle sequences with different dimensionalities.

5.2.3 Canonical Time Warping (CTW)

Canonical time warping (CTW) can align sequences with different dimensionalities by considering a common latent space [3, 15].

The CTW optimization problem is given as

$$\min_{\boldsymbol{W}_{\mathrm{x}},\boldsymbol{W}_{\mathrm{y}},\boldsymbol{V}_{\mathrm{x}},\boldsymbol{V}_{y}} \|\boldsymbol{V}_{\mathrm{x}}^{\top}\boldsymbol{X}\boldsymbol{W}_{\mathrm{x}}^{\top} - \boldsymbol{V}_{\mathrm{y}}^{\top}\boldsymbol{Y}\boldsymbol{W}_{\mathrm{y}}^{\top}\|_{\mathrm{Frob}}^2, \tag{17}$$

where $\boldsymbol{V}_{\mathrm{x}} \in \mathbb{R}^{d_{\mathrm{x}} \times b}$ and $\boldsymbol{V}_{\mathrm{y}} \in \mathbb{R}^{d_{\mathrm{y}} \times b}$ ($b \leq \min(d_{\mathrm{x}}, d_{\mathrm{y}})$) are linear projection matrices of $\boldsymbol{x}$ and $\boldsymbol{y}$ onto a common latent space, respectively. The above optimization problem can be efficiently solved by alternately performing CCA and DTW, where the alignment matrix obtained using DTW is usually used as an initial alignment matrix.

Generalized time warping (GTW) [15] can be regarded as CTW if we align two sequences and use the dynamic programming to obtain an alignment.

A limitation of CTW is that, since CTW finds a common latent space using CCA, it can only deal with linear and Gaussian temporal alignment problems. Thus, CTW cannot properly deal with multi-modal and non-Gaussian data. Another limitation of CTW is that comparing the alignment quality over different model parameters is not straightforward. This is because, for different model parameters, common latent spaces found by CCA are generally different and thus the metrics of the pairwise distance (17) are also different. For this reason, a systematic model selection method for the regularization parameter, the dimensionality of the common latent space, and the initial alignment matrix has not been properly addressed so far, to the best of our knowledge.

5.2.4 Kernelized Canonical Time Warping (KCTW)

Let us transform $\boldsymbol{X}$ and $\boldsymbol{Y}$ to higher dimensional matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ and define $\boldsymbol{V}_{\mathrm{x}} = \boldsymbol{\Phi}\widetilde{\boldsymbol{V}}_{\mathrm{x}}$ and $\boldsymbol{V}_{\mathrm{y}} = \boldsymbol{\Psi}\widetilde{\boldsymbol{V}}_{\mathrm{y}}$. Then, we have a nonlinear version of CTW (Eq.(17)) as

$$\min_{\boldsymbol{W}_{\mathrm{x}},\boldsymbol{W}_{\mathrm{y}},\widetilde{\boldsymbol{V}}_{\mathrm{x}},\widetilde{\boldsymbol{V}}_{y}} \|\widetilde{\boldsymbol{V}}_{\mathrm{x}}^{\top}\boldsymbol{K}_{\mathrm{x}}\boldsymbol{W}_{\mathrm{x}}^{\top} - \widetilde{\boldsymbol{V}}_{\mathrm{y}}^{\top}\boldsymbol{K}_{\mathrm{y}}\boldsymbol{W}_{\mathrm{y}}^{\top}\|_{\mathrm{Frob}}^{2}, \tag{18}$$

where $\boldsymbol{K}_{\mathrm{x}} = \boldsymbol{\Phi}^{\top}\boldsymbol{\Phi}$ and $\boldsymbol{K}_{\mathrm{y}} = \boldsymbol{\Psi}^{\top}\boldsymbol{\Psi}$ are the Gram matrices.

Using this formulation, one can handle nonlinearity in the CTW framework. However, it is not clear how to objectively select model parameters such as kernel parameters. That is, the KCCA-based approach works well only when appropriate model parameters are used. However, if the parameters are not chosen carefully, KCTW can perform poorly.

6 Experiments

In this section, we report experimental results.

6.1 Cross-Domain Object Matching

First, we experimentally evaluate the performance of our proposed CDOM method in image matching, unpaired voice conversion, and photo album summarization.

6.1.1 Setup

In all the methods, we use the Gaussian kernels:

$$K(\boldsymbol{x},\boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\sigma_{\mathrm{x}}^2}\right),$$
$$L(\boldsymbol{y},\boldsymbol{y}') = \exp\left(-\frac{\|\boldsymbol{y}-\boldsymbol{y}'\|^2}{2\sigma_{\mathrm{y}}^2}\right),$$

and we experimentally set the maximum number of iterations for updating permutation matrices to 20 and the step size η to 1. To avoid falling into undesirable local optima, optimization is carried out 10 times with different initial permutation matrices, which are determined by the eigenvalue-based initialization heuristic with Gaussian kernel widths

$$(\sigma_{\mathrm{x}},\sigma_{\mathrm{y}}) = c \times (m_{\mathrm{x}},m_{\mathrm{y}}),$$

where $c = 1^{1/2}, 2^{1/2}, \ldots, 10^{1/2}$, and

$$m_{\mathrm{x}} = 2^{-1/2}\mathrm{median}(\{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|\}_{i,j=1}^{n}),$$
$$m_{\mathrm{y}} = 2^{-1/2}\mathrm{median}(\{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|\}_{i,j=1}^{n}).$$

In KS-HSIC and KS-NOCCO, we use the Gaussian kernel with the following widths:

$$(\sigma_{\mathrm{x}}, \sigma_{\mathrm{y}}) = c' \times (m_{\mathrm{x}}, m_{\mathrm{y}}),$$

where $c' = 1^{1/2}, 10^{1/2}$. In KS-NOCCO, we use the following regularization parameters:

$$\epsilon = 0.01, 0.05.$$

In KS-HSIC (CV), we choose the model parameters of HSIC, σ_{x} and σ_{y} by 2-fold CV from

$$(\sigma_{\mathrm{x}}, \sigma_{\mathrm{y}}) = c \times (m_{\mathrm{x}}, m_{\mathrm{y}}),$$

where we use the cross-validation approach proposed in [30].

In LSOM, we choose the model parameters of LSMI, σ_{x}, σ_{y}, and λ by 2-fold CV[1] from

$$\begin{aligned}(\sigma_{\mathrm{x}}, \sigma_{\mathrm{y}}) &= c \times (m_{\mathrm{x}}, m_{\mathrm{y}}),\\ \lambda &= 10^{-1}, 10^{-2}, 10^{-3}.\end{aligned}$$

6.1.2 Image Matching

Let us consider a toy image matching problem. In this experiment, we use images with the RGB format used in [2], which were originally extracted from *Flickr*[2]. We first convert the images from the RGB space to the Lab space and resize them to 40×40 pixels. Then, we vertically divide the images in the middle, and make two sets of half-images of 40×20 pixels. We denote the vectorized images by $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{y}_i\}_{i=1}^n$, which are 2400-dimensional vectors ($2400 = 40 \times 20 \times 3$). We then decouple them by randomly permuting $\{\boldsymbol{y}_i\}_{i=1}^n$, and try to recover the correct correspondence by a CDOM method.

Figure 1 summarizes the average correct matching rate over 100 runs as functions of the number of images, showing that the proposed LSOM method tends to outperform the optimally tuned KS-HSIC, KS-HSIC (CV), and KS-NOCCO methods. Moreover, through experiments, we observed that the optimally tuned KS-HSIC compares favorably with KS-HSIC (CV). Figure 2 depicts an example of image matching results obtained by LSOM, showing that most of the images are correctly matched. Moreover, we plot the learning curve of LSOM in Figure 1(c) and it converges in 10 steps. Note that the tuning parameters of LSOM (σ_{x}, σ_{y}, and λ) are automatically tuned by CV.

6.1.3 Unpaired Voice Conversion

Next, we consider an unpaired voice conversion task, which is aimed at matching the voice of a source speaker with that of a target speaker.

In this experiment, we use 200 short utterance samples recorded from two male speakers in French, with sampling rate 44.1kHz. We first convert the utterance samples to

[1] We choose 2-fold cross validation to reduce the computational cost.

[2] http://www.flickr.com

50-dimensional *line spectral frequency* (LSF) vectors [31]. We denote the source and target LSF vectors by $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. Then the voice conversion task can be regarded as a multi-dimensional regression problem of learning a function from $\boldsymbol{x}$ to $\boldsymbol{y}$. However, different from a standard regression setup, paired training samples are not available; instead, only unpaired samples $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{y}_i\}_{i=1}^n$ are given.

By CDOM, we first match $\{\boldsymbol{x}_i\}_{i=1}^n$ and $\{\boldsymbol{y}_i\}_{i=1}^n$, and then we train a multi-dimensional kernel regression model [32] using the matched samples $\{(\boldsymbol{x}_{\pi(i)}, \boldsymbol{y}_i)\}_{i=1}^n$ as

$$\min_{\boldsymbol{W}} \sum_{i=1}^{n} \|\boldsymbol{y}_i - \boldsymbol{W}^\top \boldsymbol{k}(\boldsymbol{x}_{\pi(i)})\|^2 + \frac{\delta}{2}\mathrm{tr}(\boldsymbol{W}^\top \boldsymbol{W}),$$

where

$$\boldsymbol{k}(\boldsymbol{x}) = (K(\boldsymbol{x}, \boldsymbol{x}_{\pi(1)}), \ldots, K(\boldsymbol{x}, \boldsymbol{x}_{\pi(n)}))^\top,$$
$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\tau^2}\right).$$

Here, τ is a Gaussian kernel width and δ is a regularization parameter; they are chosen by 2-fold CV.

We repeat the experiments 100 times by randomly shuffling training and test samples, and evaluate the voice convergence performance by *log-spectral distance* called the spectral distortion [33] for 8000 test samples. Note that the smaller the spectral distortion is, the better the quality of voice conversion is. Figure 4 shows the true spectral envelope and their estimates, and Figure 3 shows the average performance over 100 runs as functions of the number of training samples. These results show that the proposed LSOM tends to outperform KS-NOCCO and KS-HSIC.

6.1.4 Photo Album Summarization

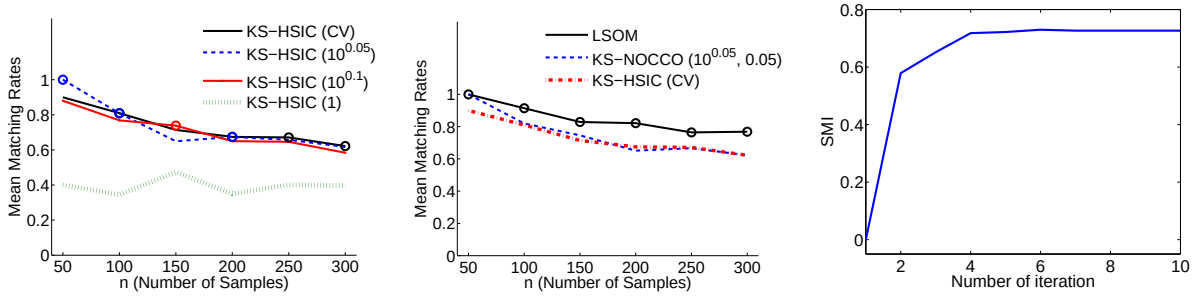
Finally, we apply the proposed LSOM method to a photo album summarization problem, where photos are automatically aligned into a designed frame expressed in the Cartesian coordinate system.

We align the Flickr[2], Frey face[34], and USPS images [35] into complex frames—a Japanese/Chinese character 'mountain', a smiley-face shape, and a '777' digit shape. The results depicted in Figure 5 show that images with similar profiles are located in nearby grid-coordinate cells.

6.2 Cross-Domain Temporal Alignment

Next, we experimentally evaluate the performance of our proposed CDTA method on synthetic, video sequence alignment, and *Kinect* action recognition tasks.

[2]http://www.flickr.com



(a) KS-HSIC with different Gaussian kernel widths.

(b) LSOM (tuned by CV), optimally-tuned KS-NOCCO, and optimally-tuned KS-HSIC.

(c) Learning curve of LSOM.

Figure 1: Image matching results. The best method in terms of the mean error and comparable methods according to the t-test at the significance level 1% are specified by '○'.

Figure 2: Image matching result by LSOM. In this case, 234 out of 320 images (73.1%) are matched correctly.

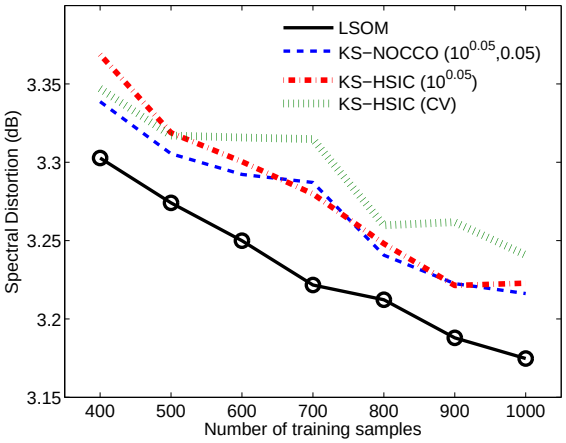


Figure 3: Unpaired voice conversion results. The best method in terms of the mean spectral distortion and comparable methods according to the t-test at the significance level 1% are specified by '○'.

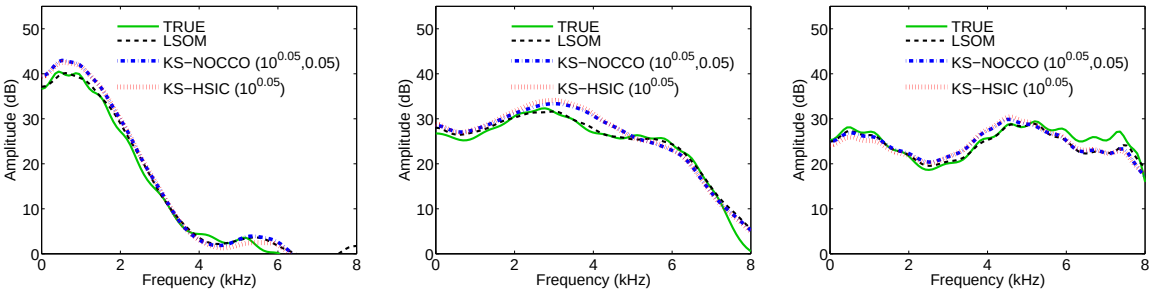
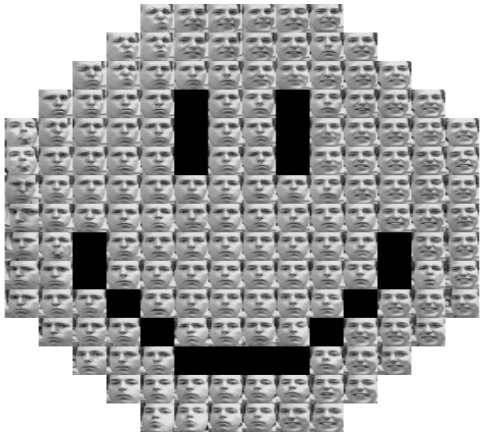
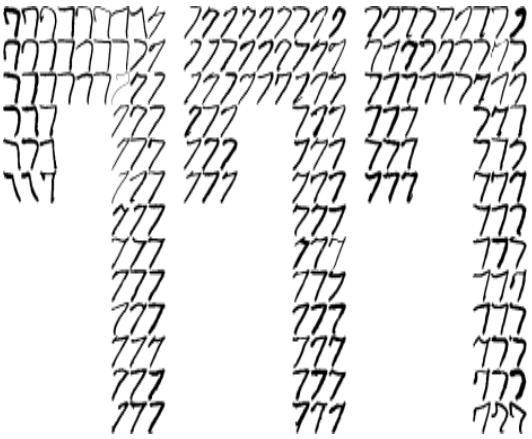


Figure 4: True spectral envelopes and their estimates.

(a) Layout of 120 images into a Japanese character 'mountain' by LSOM.



(b) Layout of 153 facial images into 'smiley' by LSOM.



(c) Layout of 199 digit '7' into '777' by LSOM.

Figure 5: Images are automatically aligned into complex grid frames expressed in the Cartesian coordinate system.

6.2.1 Setup

In LSDTW, we use the Gaussian kernels:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\sigma_{\mathrm{x}}^2}\right),$$
$$L(\boldsymbol{y}, \boldsymbol{y}') = \exp\left(-\frac{\|\boldsymbol{y} - \boldsymbol{y}'\|^2}{2\sigma_{\mathrm{y}}^2}\right),$$

where σ_{x}, σ_{y}, and the regularization parameter λ are chosen by 3-fold CV from

$$(\sigma_{\mathrm{x}}, \sigma_{\mathrm{y}}) = c \times (m_{\mathrm{x}}, m_{\mathrm{y}}),$$
$$c = 2^{-1/2}, 1.8^{-1/2}, \ldots, 0.2^{-1/2},$$
$$\lambda = 10^{-1}, 10^{-2},$$

and

$$m_{\mathrm{x}} = 2^{-1/2}\mathrm{median}(\{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|\}_{i,j=1}^{n_{\mathrm{x}}}),$$
$$m_{\mathrm{y}} = 2^{-1/2}\mathrm{median}(\{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|\}_{i,j=1}^{n_{\mathrm{y}}}).$$

Comparisons: We compare the performance of LSDTW with DTW and CTW. For DTW and CTW, we use the publicly available implementations provided by the authors of the original papers [3, 15][2]. For CTW, we choose the dimensionality of CCA to preserve 90% of the total correlation, and we fix the regularization parameter at 0.01. We use the alignment given by DTW as the initial alignment for CTW. In the video sequence alignment and the real-world Kinect action recognition experiments, we also compare LSDTW to kernel CTW (KCTW), derivative DTW (DDTW) [28], and iterative motion warping (IMW) [29]. In KCTW, we use the Gaussian kernel and set the kernel width at m_x and m_y, which is a common heuristic [32]. For existing methods, we use the same parameters as those used in [15].

Evaluation: To evaluate the alignment results, we use the following standard alignment error [15]:

$$\mathrm{Error} = \frac{\mathrm{dist}(\boldsymbol{\Pi}^*, \widehat{\boldsymbol{\Pi}}) + \mathrm{dist}(\widehat{\boldsymbol{\Pi}}, \boldsymbol{\Pi}^*)}{m^* + \widehat{m}},$$

where

$$\mathrm{dist}(\boldsymbol{\Pi}_1, \boldsymbol{\Pi}_2) = \sum_{i=1}^{m_1} \min(\{\|\boldsymbol{\pi}_1^{(i)} - \boldsymbol{\pi}_2^{(j)}\|\}_{j=1}^{m_2}),$$

$\boldsymbol{\Pi}^*$ and $\widehat{\boldsymbol{\Pi}}$ are true and estimated alignment matrices, and $\boldsymbol{\pi}_1^{(i)}, \boldsymbol{\pi}_2^{(j)} \in \mathbb{R}^{2\times 1}$ are the i-th and j-th columns of $\boldsymbol{\Pi}_1$ and $\boldsymbol{\Pi}_2$, respectively.

6.2.2 Synthetic Dataset

We first illustrate the behavior of the proposed LSDTW method on aligning two non-linearly related non-stationary sequences using a synthetic dataset.

We use the following function:

$$x_i = i/200 + 0.4\sin(\pi i/100) + e_i,\ i = 1, \ldots, 200,$$
$$y_j = ((j-1)\times 2 + 1)/200 + e_j,\ j = 1, \ldots, 100,$$

where e_i and e_j are randomly generated additive Gaussian noise with standard deviation 0.01 (see Figure 6(a) and (b)). Note that, sample-wise, for a given value of x_i there may be multiple y_j's.

[2] `www.f-zhou.com/ta_code.html`.

Figure 6(c) shows the alignment path obtained using LSDTW, CTW, and DTW, respectively. In this experiment, we initialize CTW and DTW with the true alignment matrix and check whether the corresponding methods perform well. On the other hand, we use the CTW and uniform initialization (true alignment) for LSDTW and choose the one that has the highest SMI score. As can be observed, LSDTW can find a better alignment in the middle region (i.e., a highly non-linear region) than DTW and CTW. This shows that the LSDTW objective is more appropriate than alternatives when it comes to more complex data (with noise), due to its use of more universal information-theoretic metric that is sensitive to statistical dependence (not absolute distance) and is insensitive to noise. Figure 6(d) depicts the SMI score with respect to the number of iterations in LSDTW, showing that SMI score does not change after 5 iterations. LSDTW took 14.1 seconds on 16 core Intel Xeon 2.4GHz CPU with 24G memory vs. CTW that took 2.4 seconds on the same machine.

6.2.3 Sequence Alignment

Videos with Different Features: In this experiment, we align two video sequences taken from the Weizmann database [36], which consists of 10 motion sequences performed by 9 people. Each video sequence is encoded with two different visual features; we align sequences of pairs of subjects each encoded by different feature representation. Based on [15] we extract two feature representations based on silhouettes obtained with background subtraction: (i) Euclidean distance transform [37] and (ii) solution of Poisson equation as features [38] (2450 dimensional). To reduce the dimensionality of inputs, we used the top M principal components that preserve 99.9% of the total energy of the features[3]. For evaluation, we randomly pick two walking sequences from different subjects and compute the error between the estimated alignment and the ground-truth alignment. Note that the ground-truth alignment is computed using DTW applied to the same features (see [15] for further experimental details). For competing methods, we use the same parameter setting as that used in [15]. Figure 7(a) shows the mean and variance of alignment error for different methods. LSDTW gives the smallest alignment error (with CTW and KCTW the next best).

Cross-Domain Sequence Alignment: To illustrate the capability of our method in dealing with sequences with different dimensionalities in alignment, we align sequences of different people performing a similar activity but recorded with different sensors. We use one motion capture sequence from the CMU motion capture database and one video sequence from the Weizmann database [36]. For the mocap data, we use 60-dimensional feature (the imaginary portion of a unit quaternion computed for each of the 20 joints) vectors to describe body configuration, while we use the solution of Poisson equation as features (2450 dimensional) for video. Again, to reduce the dimensionality, we apply PCA to each modality preserving 95% of total energy, resulting in the final representation

[3]We set $M = min(M_{\mathrm{x}}, M_{\mathrm{y}})$ where M_{x} is the dimensionality of distance transform features and M_{y} is dimensionality of Poisson features that preserve 99.9% of the total energy respectively.

for mocap $\boldsymbol{x}_i \in \mathbb{R}^{11}$ and image features $\boldsymbol{y}_i \in \mathbb{R}^{45}$. See [15] for the detail of the feature extraction procedure. Figure 7(b) shows key frames after alignment by LSDTW. It can be seen that LSDTW can align two sequences well, despite the fact that they are represented by signals with different dimensionalities.

6.2.4 Kinect Action Recognition

We also evaluate the proposed LSDTW method on the publicly available *Kinect* action recognition dataset[4] [39]. This dataset consists of human skeleton data (15 joints) obtained using a *Kinect* sensor, and there are 16 subjects and 16 actions with 5 trials. Instead of using the raw skeleton data, we compute a 105-dimensional feature vector for each pose, where each element of the feature vector is the Euclidean distance between joint pairs.

We carry out *unsupervised* action recognition experiments and evaluate the performance of each alignment, looking at the classification accuracy. More specifically, we first divide the action recognition dataset into two disjoint subsets: 8 subjects (#9-#16) for training database (640 sequences), and the remaining 8 subjects (#1-#8) for testing (640 sequences). At test time, we retrieve $N \leq 10$ similar sequences for each test action from the training database using DTW, KCTW, CTW, DDTW, IMW, and LSDTW; we use the pairwise Euclidean distance based on the estimated alignment to measure the similarity between sequences. We consider retrieval/classification as being correct if one of the retrieved nearest neighbor sequences has a correct action label.

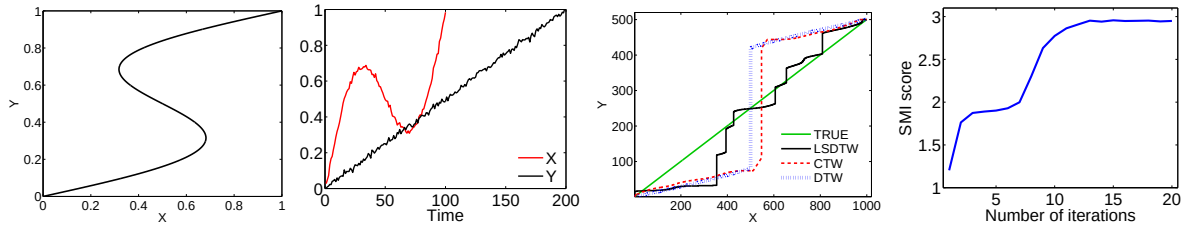
Figure 8 shows the mean classification accuracy as functions of the number of retrieved sequences, N, where three different database sizes are tested. The graphs clearly show that LSDTW compares favorably with existing methods in terms of classification accuracy across all settings. For example in Figure 8(a), the proposed method achieves more than 70% classification accuracy (the number of extracted actions is 2) while best existing methods give about 65% classification accuracy.

7 Conclusion

In this paper, we first proposed least-squares object matching (LSOM) for the *cross-domain object matching* (CDOM) problem. LSOM adopts *squared-loss mutual information* as a dependence measure, and it is estimated by the method of *least-squares mutual information* (LSMI). A notable advantage of LSOM is that it is equipped with a natural cross-validation procedure that allows us to objectively optimize tuning parameters such as the Gaussian kernel width and the regularization parameter in a data-dependent fashion.

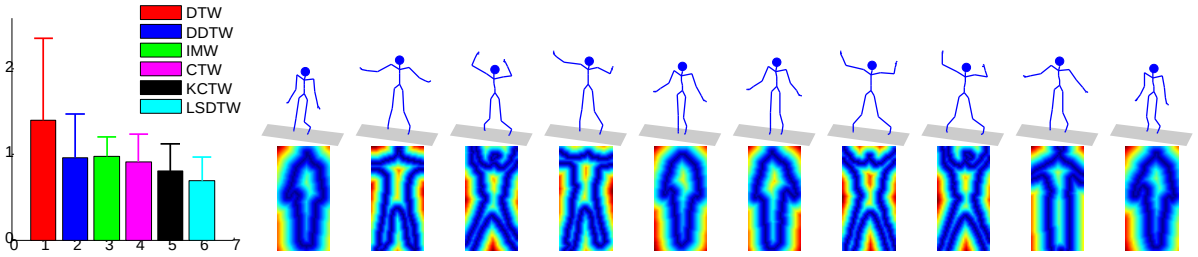
Moreover, we proposed a novel cross-domain temporal alignment framework, based on SMI maximization, that we call *least-squares dynamic time warping* (LSDTW). Similarly to LSOM, LSDTW includes its natural ability to deal with non-linearly related sequences

[4] `www.cs.ucf.edu/~smasood/datasets/UCFKinect.zip`



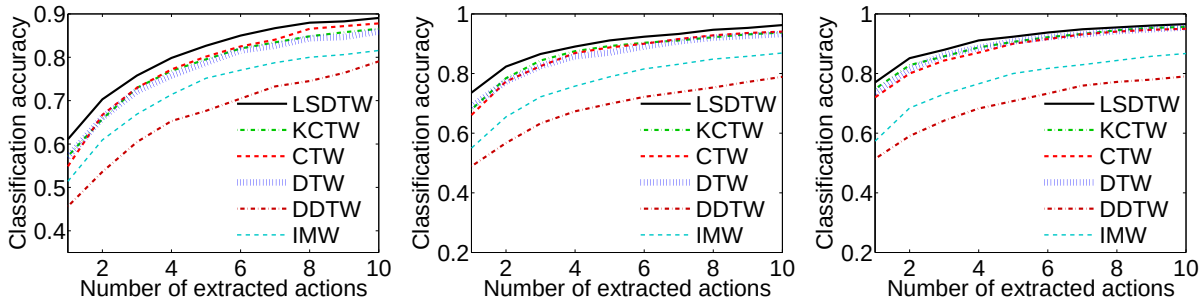
(a) Synthetic data ($\boldsymbol{X}$ vs. $\boldsymbol{Y}$) (b) Synthetic data ($\boldsymbol{X}$ and $\boldsymbol{Y}$ vs. t) (c) Alignment path (d) SMI score

Figure 6: Results of synthetic experiments. (a) Synthetic data ($\boldsymbol{X}$ vs. $\boldsymbol{Y}$). (b) Synthetic temporal signals as a function of time. (c) Alignment paths. Here, the alignment error of LSDTW, CTW, and DTW are 31.8, 69.3, and 73.9, respectively. (d) SMI score as a function of the number of iterations.



(a) Alignment error (b) Mocap (top) and video (bottom) key frames aligned using LSDTW

Figure 7: Results of video sequence alignment. (a) The mean and variance of alignment error (lower is better) for different methods. (b) The key frames after alignment using LSDTW.



(a) (b) (c)

Figure 8: Mean action classification accuracy with respect to the number of retrieved sequences. (a) Only sequences of subject#9 are used in training to form a database. (b) Sequences of Subjects#$9-12$ are used in training. (c) Sequences of all subjects (#$9-16$) are used in training.

with different dimensionalities (with non-Gaussian noise) and its ability to optimize model parameters, such as the Gaussian kernel width and the regularization parameter, by cross-validation.

We applied the proposed methods to various problems including image matching, unpaired voice conversion, photo album summarization, cross-feature video alignment, cross-domain video-to-mocap alignment, and *Kinect* action recognition, and quantitatively showed that LSOM and LSDTW are promising alternatives to state-of-the-art cross-domain matching methods.

There are several remaining issues that we leave for future work. For example, matching/alignment of multiple objects/sequences, similar to [15], can be addressed by computing squared-loss mutual information for more than two variables [40]. Moreover, one can integrate dimensionality reduction into SMI estimation [41], potentially further improving the temporal alignment performance. Finally, CDOM methods cannot handle the matching problem more than 10K samples. Recently, several efficient graph matching algorithms including a path following algorithm [42] and deformable graph matching [43] are proposed. Thus, scaling up the KS and LSOM using the state-of-the-art graph matching algorithms is also an interesting problem.

Acknowledgments

We thank Dr. Fernando Villavicencio and Dr. Akisato Kimura for their valuable comments. We also thank Dr. Feng Zhou and Dr. Fernando de la Torre for data and valuable discussions. Makoto Yamada was supported by the JST PRESTO program, and Masashi Sugiyama was supported by AOARD and KAKENHI 25700022.

References

[1] T. Jebara, "Kernelized sorting, permutation, and alignment for minimum volume PCA," in *Conference on Computational Learning theory (COLT)*, 2004, pp. 609–623.

[2] N. Quadrianto, A. Smola, L. Song, and T. Tuytelaars, "Kernelized sorting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1809–1821, October 2010.

[3] F. Zhou and F. De la Torre, "Canonical time warping for alignment of human behavior," in *Advances in Neural Information Processing Systems 22*, 2009, pp. 2286–2294.

[4] D. Gong and G. G. Medioni, "Dynamic manifold warping for view invariant action recognition," in *ICCV*, 2011.

[5] F. Zhou, F. De la Torre, and J. K. Hodgins, "Aligned cluster analysis for temporal segmentation of human motion," in *Automatic Face & Gesture Recognition (FG)*, 2008, pp. 1–7.

[6] T. B. Sebastian, P. N. Klein, and B. B. Kimia, "B.b.: On aligning curves," *IEEE TPAMI*, pp. 116–124, 2003.

[7] T. Suzuki, M. Sugiyama, T. Kanamori, and J. Sese, "Mutual information estimation reveals global associations between stimuli and biological processes," *BMC Bioinformatics*, vol. 10, no. S52, 2009.

[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2006.

[9] A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *16th International Conference on Algorithmic Learning Theory (ALT 2005)*, 2005, pp. 63–78.

[10] J. Jagarlamudi, S. Juarez, and H. Daumé III, "Kernelized sorting for natural language processing," in *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI2010)*, Atlanta, Georgia, U.S.A, Jul. 11-15 2010, pp. 1020–1025.

[11] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, no. 1, pp. 43–49, 1978.

[12] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*. Prentice-Hall Signal Processing Series, 1993.

[13] J. Listgarten, R. M. Neal, S. T. Roweis, and A. Emili, "Multiple alignment of continuous time series," in *Advances in neural information processing systems 18*, 2005, pp. 817–824.

[14] E. Hsu, K. Pulli, and J. Popovi, "Style translation for human motion," *ACM Transactions on Graphics*, vol. 24, pp. 1082–1089, 2005.

[15] F. Zhou and F. De la Torre, "Generalized time warping for multi-modal alignment of human motion," in *CVPR*, 2012, pp. 1282–1289.

[16] R. Bellman, "On the Theory of Dynamic Programming," in *Proceedings of the National Academy of Sciences*, vol. 38, 1952, pp. 716–719.

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

[18] M. Yamada and M. Sugiyama, "Cross-domain object matching with model selection," *AISTATS*, pp. 807–815, 2011.

[19] K. Pearson, "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *Philosophical Magazine*, vol. 50, pp. 157–175, 1900.

[20] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, pp. 79–86, 1951.

[21] T. P. Minka, "Old and new matrix algebra useful for statistics," MIT Media Lab, Tech. Rep., 2000.

[22] H. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[23] G. H. Hardy, J. E. Littlewood, and G. Pólya, *Inequalities*. Cambridge: Cambridge University Press, 1952.

[24] M. Sugiyama, "Machine learning with squared-loss mutual information," *Entropy*, vol. 15, no. 1, pp. 80–112, 2013.

[25] G. Finke, R. E. Burkard, and F. Rendl, "Quadratic assignment problems," *Annals of Discrete Mathematics*, vol. 31, pp. 61–82, 1987.

[26] K. Fukumizu, F. R. Bach, and M. Jordan, "Kernel dimension reduction in regression," *The Annals of Statistics*, vol. 37, no. 4, pp. 1871–1905, 2009.

[27] K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf, "Kernel measures of conditional dependence," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Botton, Eds. Cambridge, MA: MIT Press, 2009, pp. 489–496.

[28] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *SDM*, 2001.

[29] E. Hsu, K. Pulli, and J. Popović, "Style translation for human motion," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 1082–1089, 2005.

[30] M. Sugiyama and M. Yamada, "On kernel parameter selection in hilbert-schmidt independence criterion," *IEICE TRANSACTIONS on Information and Systems*, vol. 95, no. 10, pp. 2564–2567, 2012.

[31] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," in *Proceedings of 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP1998)*, Washington, DC, U.S.A, May. 12–15 1988, pp. 285–288.

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

[33] S. R. Quackenbush, T. P. Barnwell, and M. A. Clements, *Objective Measures of Speech Quality*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1988.

[34] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[35] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer, 2001.

[36] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *ICCV*, 2005.

[37] C. R. Maurer Jr, R. Qi, and V. Raghavan, "A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions," *IEEE TPAMI*, vol. 25, no. 2, pp. 265–270, 2003.

[38] L. Gorelick, M. Galun, E. Sharon, R. Basri, and A. Brandt, "Shape representation and classification using the Poisson equation," *IEEE TPAMI*, vol. 28, no. 12, pp. 1991–2005, 2006.

[39] S. Z. Masood, A. Nagaraja, N. Khan, J. Zhu, and M. F. Tappen, "Correcting cuboid corruption for action recognition in complex environment," in *ICCV Workshops*, 2011.

[40] T. Suzuki and M. Sugiyama, "Least-squares independent component analysis," *Neural Computation*, vol. 23, no. 1, pp. 284–301, 2011.

[41] M. Sugiyama, M. Kawanabe, and P. L. Chui, "Dimensionality reduction for density ratio estimation in high-dimensional spaces," *Neural Networks*, vol. 23, no. 1, pp. 44–59, 2010.

[42] M. Zaslavskiy, F. Bach, and J.-P. Vert, "A path following algorithm for the graph matching problem," *IEEE TPAMI*, vol. 31, no. 12, pp. 2227–2242, 2009.

[43] F. Zhou and F. De la Torre, "Deformable graph matching," in *CVPR*. IEEE, 2013, pp. 2922–2929.