JMLR: Workshop and Conference Proceedings 101:551–566, 2019 ACML 2019

Canonical Soft Time Warping

Keisuke Kawano KAWANO@MOSK.TYTLABS.CO.JP
Satoshi Koide KOIDE@MOSK.TYTLABS.CO.JP
Takuro Kutsuna KUTSUNA@MOSK.TYTLABS.CO.JP
Toyota Central R&D Labs., 41-1 Yokomichi, Nagakute, Aichi, Japan

Editors: Wee Sun Lee and Taiji Suzuki

Abstract

Alignment of two given sequences (i.e., computing correspondence between frames considering local time shifting) is a fundamental operation for various applications such as computer vision and bioinformatics. To obtain an alignment between high-dimensional sequences, several methods have been proposed, including canonical time warping (CTW). However, the optimization problem for CTW, and its extensions, often fall into poor local minima when the initial solution is far from the global optima. In this paper, we propose *canonical soft time warping (CSTW)* in which an alignment is modeled as a probabilistic variable that follows the Gibbs distribution with temperature γ. We also propose the annealing CSTW (ACTW), a variant of CSTW that gradually decreases γ. ACTW is useful when underlying applications require hard alignments. Using synthetic and real-world data, we experimentally demonstrate that our proposed methods outperform previous methods, including CTW, in estimating alignments. In particular, our method does not suffer from poor local minima, as a consequence of the probabilistic treatment of alignments.

Keywords: sequence alignment, temporal alignment, dynamic time warping, soft dynamic time warping, canonical correlation analysis

1. Introduction

Alignment of two given sequences (i.e., computing correspondence between frames considering local time shifting) is a fundamental operation in a wide range of applications including computer vision (Chang et al., 2019), bioinformatics (Altschul et al., 1990; Aach and Church, 2001), and human activity analysis (Sheikh et al., 2005; Gritai et al., 2009). An important challenge in this area is dealing with high dimensional data, such as movies and dynamic point clouds. To address this issue, Zhou and De la Torre (2009) proposed canonical time warping (CTW), which combines canonical correlation analysis (CCA) (Hardoon et al., 2004) with dynamic time warping (DTW) (Sakoe and Chiba, 1978).

CTW aligns two high dimensional sequences by (1) finding a low dimensional latent representation and (2) applying DTW against those low dimensional sequences. The authors of CTW proposed an optimization method that alternately applies CCA and DTW. One problem of this alternating optimization is that CTW often converges to poor local minima, especially when initial values of the optimization variables are far from global solutions. For example, when the optimal alignment is far from the diagonal (i.e., the number of frames aligned to each frame has large variance), CTW shows low performance if the optimization starts from the diagonal alignment, as we will see experimentally in Section 5.2. It is

noteworthy that extensions of CTW, such as Zhou and De la Torre (2012) and Trigeorgis et al. (2016), also suffer from the same problem.

In this paper, we address this issue by considering the alignment as a random variable, which differs from the deterministic alignment used in previous studies. In our model, *canonical soft time warping (CSTW)*, we can obtain the latent representation considering all possible alignments with the assumption that the alignment follows the Gibbs distribution with temperature γ. The benefits of CSTW are summarized as follows: (1) CSTW is proved to be a generalization of CTW under some mild conditions. We can obtain a solution for CTW by solving CSTW with γ annealing to 0. (2) By annealing γ, CSTW can mitigate the poor local minima problem in CTW. The distribution of alignment is controlled from the uniform distribution to a deterministic one during the optimization by controlling γ, which makes the optimization less dependent on the inital value. (3) A solution of CSTW can achieve better performance than that of CTW in terms of the latent representation learning if γ is tuned properly.

CSTW is hard to solve directly, because the number of possible alignments can be exponential. To avoid this issue, we consider an upper bound of the objective function of CSTW. The upper bound is then optimized by our proposed two-stage algorithm similar to the expectation-maximization (EM) algorithm (Dempster et al., 1977). Instead of dealing with an exponential number of alignments directly, our algorithm only need to evaluate the expectation of alignment in each optimization step, which can be efficiently computed by using recently proposed soft-DTW (Cuturi and Blondel, 2017).

Our contributions are summarized as follows.

1. For high-dimensional sequences, we propose a novel alignment framework called CSTW, which considers an alignment as a random variable. CSTW is proved to be a generalization of CTW under mild conditions (Section 3.1).

2. We also propose an efficient optimization method for solving CSTW similar to the EM-algorithm (Section 3.2).

3. We propose annealing CSTW (ACTW) as a variant of CSTW, which mitigates the poor local minima problem in CTW (Section 3.4).

4. Experimentally, we demonstrate that CSTW and ACTW outperforms the existing alignment methods in several scenarios using synthetic and real datasets (Section 5).

This paper is organized as follows. Section 2 introduces CTW and related methods. In Section 3, we formulate CSTW and discuss its relationship to CTW. An efficient algorithm to solve CSTW is also proposed in Section 3. Section 4 gives a literature review of related work. We show our experimental results in Section 5 and conclude this paper in Section 6.

2. Preliminaries

2.1. Notations

Given a matrix A, we denote its (i, j)-th element and i-th row vector by $[A]_{i,j}$ and $[A]_i$, respectively. For $x \in \mathbb{R}^T$, $\mathrm{diag}(x)$ is the $T \times T$ matrix with diagonal x and zero otherwise. We denote a vector of ones as $\mathbf{1}_N = [1, \ldots, 1]^\top \in \mathbb{R}^N$ and a vector of zeros as $\mathbf{0}_N = [0, \ldots, 0]^\top \in$

$\mathbb{R}^N$. We also denote a matrix of zeros as $\mathbf{0}_{N\times M} = [[0,\dots 0]^\top,\dots,[0,\dots 0]^\top] \in \mathbb{R}^{N\times M}$. Let $\mathbb{I}_d$ be an identity matrix of size $d\times d$, and let $\mathcal{D}_d$ be a set of diagonal matrices of size $d\times d$. The Frobenius inner-product of two matrices of the same size A and B is $\langle A,B\rangle = \mathrm{tr}(A^\top B)$.

2.2. CCA

Here, we briefly introduce CCA (Hardoon et al., 2004). Given a set of N pairs of multivariate data denoted by $\mathcal{D} := \{(x_i,y_i) \in \mathbb{R}^{d_x}\times\mathbb{R}^{d_y} \mid i = 1,\cdots,N\}$, CCA extracts a latent representation that maximizes the correlation between the latent vectors of x_i and y_i. Let $X = [x_1,\dots,x_N] \in \mathbb{R}^{d_x\times N}$ and $Y = [y_1,\dots,y_N] \in \mathbb{R}^{d_y\times N}$ be data matrices corresponding to $\mathcal{D}$. Mathematically, CCA calculates projection matrices $U \in \mathbb{R}^{d_x\times d}$ and $V \in \mathbb{R}^{d_y\times d}$ by solving the following optimization problem:

$$
\begin{aligned}
\text{(CCA)}\ \min_{U,V}\quad & \|U^\top X - V^\top Y\|_F^2, \\
\text{s.t.}\quad & X\mathbf{1}_N = \mathbf{0}_{d_x}, Y\mathbf{1}_N = \mathbf{0}_{d_y} && \text{(zero mean)}, \\
& U^\top\Sigma_{xx}U = \mathbb{I}_d,\quad V^\top\Sigma_{yy}V = \mathbb{I}_d && \text{(identity)}, \\
& U^\top\Sigma_{xy}V \in \mathcal{D}_d && \text{(orthogonality)},
\end{aligned}
$$

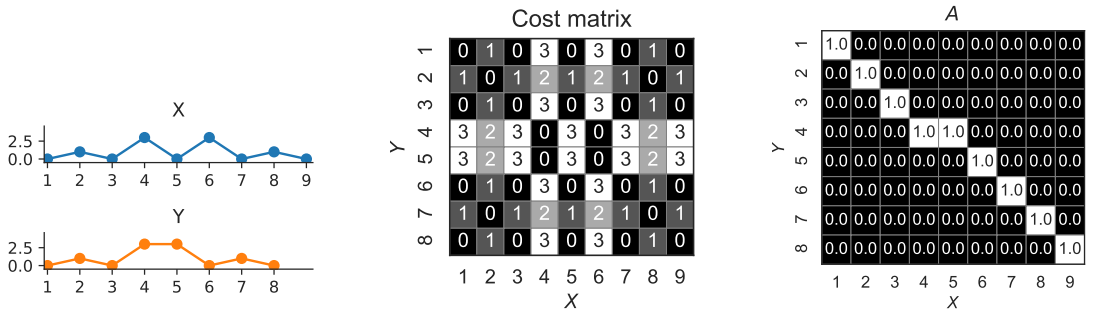
where $\Sigma_{xx} = XX^\top, \Sigma_{yy} = YY^\top$ and $\Sigma_{xy} = XY^\top$ are variance-covariance matrices with the first two constraints. The last three constraints are imposed for the scale and rotation invariance of projection matrices. These constraints avoid meaningless solutions such as both U and V being zeros matrices. It is well known that the above problem can be reformulated into the generalized eigenvalue problem to be solved efficiently.

2.3. DTW

DTW is a discrepancy between two sequences $X = [x^{(1)},\dots,x^{(T_x)}] \in \mathbb{R}^{d\times T_x}$ and $Y = [y^{(1)},\dots,y^{(T_y)}] \in \mathbb{R}^{d\times T_y}$. Comparing two sequences with different lengths, DTW finds the best alignment, which has the lowest cumulative cost of pair-wise costs. Pair-wise costs are denoted as a cost matrix $\Delta(X,Y) \in \mathbb{R}^{T_x\times T_y}$, where $[\Delta(X,Y)]_{i,j} = \|x^{(i)} - y^{(j)}\|_2^2$. An alignment matrix $A \in \{0,1\}^{T_x\times T_y}$ satisfies the following constraints: (1) boundary conditions ($[A]_{1,1} = [A]_{T_x,T_y} = 1$) and (2) monotonicity and continuity ($[A]_{i+1,j} + [A]_{i,j+1} \le 1,\ \forall i,j$ and if $[A]_{i,j} = 1$ then $0 < [A]_{i+1,j} + [A]_{i,j+1} + [A]_{i+1,j+1} \le 2$). We denote a set of alignment matrices that satisfy the constraints as $\mathcal{A}$ (i.e., $A \in \mathcal{A}$). Here, we can represent the optimization problem for DTW as

$$
\text{(DTW)}\quad \min_{A\in\mathcal{A}} \langle A, \Delta(X,Y)\rangle\,.
$$

It is well known that the solution can be obtained via dynamic programming (Sakoe and Chiba, 1978). We show an example of two uni-dimensional sequences X and Y in Figure 1(a) and their cost matrix in Figure 1(b). Figure 1(c) shows an example of an alignment matrix in which the alignment path is shown by white cells. An alignment path starts from the top-left cell and ends at the bottom-right cell, where three types of moves are allowed at each cell (going right, going down, and going down-right), corresponding to the constraints of the alignment matrix.



(a) Input sequences (X, Y) (b) Cost matrix $(\Delta(X, Y))$ (c) Alignment matrix (A)

Figure 1: An example of DTW

DTW with warping matrices. Given an alignment matrix A, we can obtain warping paths $p_x(A) \in \{1 : T_x\}^{l_w(A)}$ and $p_y(A) \in \{1 : T_y\}^{l_w(A)}$ uniquely, which store the x and y coordinates of A, respectively. $l_w(A) = \sum_{i,j}[A]_{i,j}$ represents a length of the warping path. For the example in Figure 1(c), the warping paths are $p_x(A) = [1, 2, 3, 4, 5, 6, 7, 8, 9]$ and $p_y(A) = [1, 2, 3, 4, 4, 5, 6, 7, 8]$. Here, we can reformulate the alignment problem with *warping matrices* $W_x(A) \in \{0, 1\}^{l_w(A) \times T_x}$ and $W_y(A) \in \{0, 1\}^{l_w(A) \times T_y}$ defined as follows:

$$[W_x(A)]_{i,j} = \begin{cases} 1 & \text{if } j = p_x(A)^{(i)}, \\ 0 & \text{otherwise,} \end{cases}$$

$$[W_y(A)]_{i,j} = \begin{cases} 1 & \text{if } j = p_y(A)^{(i)}, \\ 0 & \text{otherwise.} \end{cases}$$

We can easily check that $A = W_x(A)^\top W_y(A)$. Thus, the optimization problem for DTW can be rewritten as

$$\min_{A \in \mathcal{A}} \|XW_x(A)^\top - YW_y(A)^\top\|_F^2 .$$

Note that every row of the warping matrices is a one-hot vector.

2.4. Canonical Time Warping

DTW is not directly applicable when comparing two sequences that have different feature dimensions. CTW allows us to align high-dimensional sequences by integrating CCA and DTW. Here, we consider two sequences $X = [x^{(1)}, \ldots, x^{(T_x)}] \in \mathbb{R}^{d_x \times T_x}$ and $Y = [y^{(1)}, \ldots, y^{(T_y)}] \in \mathbb{R}^{d_y \times T_y}$, even if $d_x \neq d_y$. Intuitively, CTW applies DTW for the two latent sequences that are embedded by CCA. Formally, CTW solves the following optimization problem to obtain the projection matrices $U \in \mathbb{R}^{d_x \times d}, V \in \mathbb{R}^{d_y \times d}$ as well as the

alignment matrix A.

$$\begin{aligned}
\text{(CTW)} \min_{U,V,A} \quad & \|U^\top X W_x(A)^\top - V^\top Y W_y(A)^\top\|_F^2 \\
\text{s.t. } & X\tilde{w}_x = \mathbf{0}_{d_x},\ Y\tilde{w}_y = \mathbf{0}_{d_y}, && \text{(zero mean)} \\
& U^\top \tilde{\Sigma}_{xx} U = \mathbb{I}_d, && \text{(identity)} \\
& V^\top \tilde{\Sigma}_{yy} V = \mathbb{I}_d, && \text{(identity)} \\
& U^\top \tilde{\Sigma}_{xy} V \in \mathcal{D}_d, && \text{(orthogonality)}
\end{aligned}$$

where $\tilde{w}_x = W_x(A)^\top \mathbf{1}_{l_w(A)}$ and $\tilde{w}_y = W_y(A)^\top \mathbf{1}_{l_w(A)}$. $\tilde{\Sigma}_{xx} := XW_x(A)^\top W_x(A)X^\top$, $\tilde{\Sigma}_y := YW_y(A)^\top W_y(A)Y^\top$, and $\tilde{\Sigma}_{xy} := XW_x(A)^\top W_y(A)Y^\top$ are variance-covariance matrices for warped sequences. The objective function above can be written as

$$\min_{U,V,A} \left\langle A, \Delta(U^\top X, V^\top Y) \right\rangle$$

from the discussion in the previous subsection. Note that the objective function for CCA can be written in a similar form as $\min_{U,V} \left\langle \mathbb{I}_N, \Delta(U^\top X, V^\top Y) \right\rangle$,where N denotes the number of data points.

To solve the optimization problem in CTW, a method that alternately solves CCA and DTW was proposed by Zhou and De la Torre (2009). In the CCA-phase, the projection matrices U and V are updated by solving the CCA with the alignment matrix A to be fixed. In the DTW-phase, A is updated by solving the DTW with U and V to be fixed. Several initialization methods are proposed for solving CTW, including the identity initialization (Zhou and De la Torre, 2009) and the uniform time warping (UTW) initialization (Zhou and De la Torre, 2012). In the identity initialization, U and V are initialized using an identity matrix and a matrix of zeros (i.e., $U^{(1)} = [\mathbb{I}_{d_x}, \mathbf{0}_{d_x \times (d-d_x)}]^\top$, $V^{(1)} = [\mathbb{I}_{d_y}, \mathbf{0}_{d_y \times (d-d_y)}]^\top$). Especially when $T_x = T_y$, UTW setting initializes A with an identity matrix in $\mathbb{R}^{T_x \times T_y}$

2.5. Soft-DTW

Soft-DTW is a smoothed version of DTW, which computes a *soft-minimum* of all alignment costs (Cuturi and Blondel, 2017). The soft-DTW is defined as follows:

$$\mathrm{dtw}_\gamma(X, Y) = \min_{A \in \mathcal{A}}{}^{\gamma} \left\langle A, \Delta(X, Y) \right\rangle,$$

where $\gamma \geq 0$ is a temperature parameter and $\min^\gamma$ is the soft-minimum function

$$\min_{A \in \mathcal{A}}{}^{\gamma} f(A) := \begin{cases} \min_{A \in \mathcal{A}} f(A) & \gamma = 0, \\ -\gamma \log \sum_{A \in \mathcal{A}} \exp(-f(A)/\gamma) & \gamma > 0. \end{cases}$$

Note that when $\gamma = 0$, the soft-DTW is equal to the standard DTW.

With the assumption that the alignment matrix A follows the Gibbs distribution $P_\gamma(A) \propto \exp(-\left\langle A, \Delta(X, Y) \right\rangle / \gamma)$, it is shown in (Cuturi and Blondel, 2017) that the expectation $\mathbb{E}_{A \sim P_\gamma(A)}[A]$ can be obtained as a gradient of soft-DTW w.r.t. the cost matrix $\Delta(X, Y)$ as

$$\mathbb{E}_{A \sim P_\gamma(A)}[A] = \nabla_{\Delta(X,Y)} \mathrm{dtw}_\gamma(X, Y).$$

3. Proposed method: Canonical Soft Time Warping

Experimentally, we found that CTW and its extensions converge to poor local minima when initial alignments are far from the global optima, e.g., when the ground truth alignment matrix is far from a diagonal matrix. One possible reason for this is the deterministic implementation of the alignments in the alternating optimization. In each step of the alternation, the accuracy of the projection matrices $U \in \mathbb{R}^{d_x \times d}$ and $V \in \mathbb{R}^{d_y \times d}$ heavily depends on the alignment. Therefore, when the alignment is far from the ground truth (which can happen especially in the earlier stage of the optimization), the projection matrices also fall into poor solutions and cannot get out of them.

In order to mitigate the problems of CTW, we propose CSTW that regards the alignment as a random variable and estimates its probabilistic distribution. The expectation of the alignment over all possible alignments is considered in the optimization of CSTW, whereas only the deterministic alignment is considered in CTW. This is the key to avoid the poor local minima problem. To solve CSTW, we propose an EM-like alternating optimization, by which the expectation of the alignment matrix is naturally introduced.

3.1. Formulation of CSTW

Given two sequences X and Y, the objective function of CSTW is defined by

$$\min_{A \in \mathcal{A}}{}^{\gamma} \left\langle A, \Delta(U^\top X, V^\top Y) \right\rangle,$$

where $\gamma > 0$ is a temperature parameter. From the following inequality:

$$\min_{A \in \mathcal{A}} \left\{ \left\langle A, \Delta(U^\top X, V^\top Y) \right\rangle \right\} \leq \min_{A \in \mathcal{A}}{}^{\gamma} \left\langle A, \Delta(U^\top X, V^\top Y) \right\rangle + \log \|\mathcal{A}\|,$$

where $\|\mathcal{A}\|$ is the number of possible alignment paths, which is a constant value, we can decrease the objective function of CTW by decreasing that of CSTW.

Since CSTW regards the alignment matrix as a random variable, the constraints in CTW, such as *zero mean*, cannot directly be applied to CSTW. Therefore, we introduce novel constraints for CSTW that involves the expectation of the alignment matrix. The overall optimization problem for CSTW is given as follows:

$$\begin{aligned}
\text{(CSTW)} \quad & \min_{U,V} \min_{A \in \mathcal{A}}{}^{\gamma} \left\langle A, \Delta(U^\top X, V^\top Y) \right\rangle && (1)\\
\text{s.t. } & X \mathbb{E}_{A \sim P_\gamma(A;U,V)} [\tilde{w}_x] = \mathbf{0}_{d_x}, && \text{(zero mean)}\\
& Y \mathbb{E}_{A \sim P_\gamma(A;U,V)} [\tilde{w}_y] = \mathbf{0}_{d_y}, && \text{(zero mean)}\\
& U^\top \mathbb{E}_{A \sim P_\gamma(A;U,V)} \left[\tilde{\Sigma}_{xx} \right] U = \mathbb{I}_d, && \text{(identity)}\\
& V^\top \mathbb{E}_{A \sim P_\gamma(A;U,V)} \left[\tilde{\Sigma}_{yy} \right] V = \mathbb{I}_d, && \text{(identity)}\\
& U^\top \mathbb{E}_{A \sim P_\gamma(A;U,V)} \left[\tilde{\Sigma}_{xy} \right] V \in \mathcal{D}_d, && \text{(orthogonality)}
\end{aligned}$$

where

$$P_\gamma(A;U,V) \propto \exp \left\{ - \left\langle A, \Delta(U^\top X, V^\top Y) \right\rangle / \gamma \right\}$$

is the Gibbs distribution, $\tilde{w}_x = W_x(A)^\top \mathbf{1}_{l_w(A)}$, and $\tilde{w}_y = W_y(A)^\top \mathbf{1}_{l_w(A)}$. $\tilde{\Sigma}_{xx}$, $\tilde{\Sigma}_{yy}$, and $\tilde{\Sigma}_{xy}$ are variance-covariance matrices for warped sequences, which are defined in CTW. When $\gamma \to 0$, a solution of CSTW is also a solution of CTW under some mild conditions (supplementary). This indicates that CSTW is a generalization of CTW. A usual way to solve constraints minimization is Lagrangian-based optimization, however, we cannot obtain the differentials of the Lagrangian w.r.t. the parameters U and V in closed forms. In the following subsection, we propose an efficient optimization algorithm.

3.2. Optimization algorithm for CSTW

In order to minimize the objective function of CSTW with respect to U and V efficiently, we consider an upper bound of the objective function using the Jensen's inequality as follows:

$$\begin{aligned}
&\min_{A\in\mathcal{A}}{}^{\gamma} \left\langle A, \Delta(U^\top X, V^\top Y) \right\rangle \\
&\leq -\gamma \sum_{A\in\mathcal{A}} q(A) \log \frac{\exp\left\{-\left\langle A, \Delta(U^\top X, V^\top Y)\right\rangle/\gamma\right\}}{q(A)} \qquad (2)\\
&= \left\langle \mathbb{E}_{A\sim q(A)}[A], \Delta(U^\top X, V^\top Y) \right\rangle + \gamma \sum_{A\in\mathcal{A}} q(A)\log q(A), \qquad (3)
\end{aligned}$$

where $q(A)$ is an arbitrary distribution of A. The equality holds in (2) if and only if

$$D_{\mathrm{KL}}\left[q(A)||P_\gamma(A;U,V)\right] = 0. \qquad (4)$$

Then, we can minimize the upper bound of the objective function for CSTW via two-stage alternating updates similar to the EM algorithm (Dempster et al., 1977). Here, we denote the projection matrices in step t as $U^{(t)}$ and $V^{(t)}$. In the E-step, we update $q(A)$ such that Eq. (4) holds with the assumption that U and V are fixed to $U^{(t)}$ and $V^{(t)}$, respectively, i.e., $q(A) \leftarrow P_\gamma(A;U^{(t)},V^{(t)})$. Then, in the M-step, U and V are updated to minimize the upper bound (3) with the current $q(A)$. An important point is that we only need to evaluate $\mathbb{E}_{A\sim q(A)}[A]$ in (3) to update U and V in the M-step, which can be efficiently obtained via soft-DTW as follows:

$$\mathbb{E}_{A\sim q(A)}\left[A\right] = \nabla_{\Delta((U^{(t)})^\top X,(V^{(t)})^\top Y)} \mathrm{dtw}^\gamma((U^{(t)})^\top X, (V^{(t)})^\top Y)) \qquad (5)$$

In addition, we employ the following constraints at each M-step to obtain feasible solutions:

$$\begin{aligned}
&X\mathbb{E}_{A\sim P_\gamma(A;U^{(t)},V^{(t)})}[W_x(A)^\top \mathbf{1}_{l_w(A)}] = \mathbf{0}_{d_x}, && \text{(zero mean)}\\
&Y\mathbb{E}_{A\sim P_\gamma(A;U^{(t)},V^{(t)})}[W_y(A)^\top \mathbf{1}_{l_w(A)}] = \mathbf{0}_{d_y}, && \text{(zero mean)}\\
&U^\top X\mathbb{E}_{A\sim P_\gamma(A;U^{(t)},V^{(t)})}\left[W_x(A)^\top W_x(A)\right] X^\top U = \mathbb{I}_d, && \text{(identity)} \qquad (6)\\
&V^\top Y\mathbb{E}_{A\sim P_\gamma(A;U^{(t)},V^{(t)})}\left[W_y(A)^\top W_y(A)\right] Y^\top V = \mathbb{I}_d, && \text{(identity)}\\
&U^\top X\mathbb{E}_{A\sim P_\gamma(A;U^{(t)},V^{(t)})}\left[A\right] Y^\top V \in \mathcal{D}_d. && \text{(orthogonality)}
\end{aligned}$$

When the updates of $U^{(t)}$ and $V^{(t)}$ converge, the projection matrices satisfy the constraints of CSTW. The above constraints include five terms that require the calculation of expectation over the distribution of A. We can efficiently evaluate these terms by applying the

alignment kernel trick to be explained in the next subsection. Once these terms are computed, we can update U and V in the M-step very efficiently by solving the generalized eigenvalue problem in the same way as solving CCA. The algorithm for solving CSTW is summarized in Algorithm 1, in which the identity initialization is utilized. It is also possible to employ the UTW initialization in CSTW. However, as will be shown in our experiment, the convergence of CSTW is less affected by the initialization when the annealing method (described in Section 3.4) is used.

Algorithm 1: CSTW (identity initialization)

Input: X, Y, γ, d
Output: U, V
Initialize U, V by $U \leftarrow [\mathbb{I}_{d_x}, \mathbf{0}_{d_x \times (d-d_x)}]^\top$ and $V \leftarrow [\mathbb{I}_{d_y}, \mathbf{0}_{d_y \times (d-d_y)}]^\top$.
while J *does not converge* **do**
  [E-step] Update $\mathbb{E}[A]$ according to Eq. (5) with U and V fixed.
  [M-step] Update U and V to minimize Eq. (3) s.t. constraints (6) with $\mathbb{E}[A]$ fixed.
  $J \leftarrow \langle \mathbb{E}[A], \Delta(U^\top X, V^\top Y) \rangle$
end

3.3. Alignment Kernel Tricks

We propose techniques called the *alignment kernel tricks* for computing expectation values in the constraints in the M-step. Note that we cannot obtain the expectations of warping matrices $\mathbb{E}_{A\sim q(A)}[W_x]$ and $\mathbb{E}_{A\sim q(A)}[W_y]$ directly by using soft-DTW. The following proposition enables us to compute expectation values in the constraints by using only the expectation of A that can be computed via soft-DTW.

Proposition 1 *For any probabilistic density function $q(A)$, the following equations hold.*

$$\begin{aligned}
\mathbb{E}_{A\sim q(A)}[W_x(A)^\top \mathbf{1}_{l_w(A)}] &= \mathbb{E}_{A\sim q(A)}[A]\mathbf{1}_{T_y}, \\
\mathbb{E}_{A\sim q(A)}[W_y(A)^\top \mathbf{1}_{l_w(A)}] &= \mathbb{E}_{A\sim q(A)}[A]^\top \mathbf{1}_{T_x}, \\
\mathbb{E}_{A\sim q(A)}[W_x(A)^\top W_x(A)] &= \mathrm{diag}\big(\mathbf{1}_{T_y}^\top \mathbb{E}_{A\sim q(A)}[A]^\top\big), \\
\mathbb{E}_{A\sim q(A)}[W_y(A)^\top W_y(A)] &= \mathrm{diag}\big(\mathbf{1}_{T_x}^\top \mathbb{E}_{A\sim q(A)}[A]\big).
\end{aligned}$$

Proof As mentioned in Section 2.3, $[W_y]_k$ is an one-hot vector for all $k = 1, \ldots, l_w(A)$. Therefore, we can obtain the first equation as

$$\begin{aligned}
\mathbb{E}_{A\sim q(A)}[W_x(A)^\top \mathbf{1}_{l_w(A)}] &= \mathbb{E}_{A\sim q(A)}[W_x(A)^\top W_y(A)\mathbf{1}_{T_y}] \\
&= \mathbb{E}_{A\sim q(A)}[A]\mathbf{1}_{T_y}.
\end{aligned}$$

We can obtain the second one in a similar manner to the first one. Next, we show the third one. If $i \neq j$, $[W_x(A)]_{k,i}[W_x(A)]_{k,j} = 0$ since $[W_x(A)]_k$ is an one-hot vector. When $i = j$,

$$\sum_k [W_x(A)]_{k,i}[W_x(A)]_{k,j} = \sum_k [W_x(A)]_{k,i} = [W_x(A)^\top \mathbf{1}_{l_w(A)}]_i = \big[A\mathbf{1}_{T_y}\big]_i .$$

Therefore,

$$\mathbb{E}_{A\sim q(A)}[W_x(A)W_x(A)^\top]_{i,j} = \begin{cases} \left[\mathbb{E}_{A\sim q(A)}[A]\mathbf{1}_{T_y}\right]_i & i = j, \\ 0 & i \neq j. \end{cases}$$

The fourth equation can be obtained in a similar way. ■

3.4. Annealing CSTW

In some applications, deterministic alignment needs to be obtained, while CSTW provides a probabilistic one. In this situation, by annealing the temperature parameter γ from a large value to zero in the alternation, we can obtain deterministic alignments because, with $\gamma \to 0$, the soft min converges to the hard min function in soft-DTW. Compared with the standard CTW, this approach mitigates the poor local minima problem because of the following reasons. For sufficiently large γ, the $\min^\gamma$ function returns a constant value regardless of the input alignment $A \in \mathcal{A}$. Therefore, the objective function does not depend on the initial alignment thus we can focus on optimizing the projection matrices in the early stage of the optimization. As γ decreases, the optimization process gradually begins focusing on the alignment. This approach weakens the dependence on the initial alignment and mitigates the poor local minima problem as shown in our experiments. We call this method annealing CSTW (ACTW).

4. Related Work

Time Series Alignment. Here, we give a brief overview of time series alignment methods. The most fundamental approach for time series alignment is DTW (Sakoe and Chiba, 1978), which has been extended in various ways. One of the extensions of DTW is derivative DTW (DDTW) (Keogh and Pazzani, 2001) using differences of derivatives instead of Euclidean distances as the frame-wise distance in order to align *shapes* of input sequences. In order to align high-dimensional sequences, various machine learning-based methods have been proposed. These methods can be divided into two categories according to their training data types: supervised methods and unsupervised methods. Supervised methods, including neural network-based models (Dogan et al., 2018), require true alignment paths (i.e., frame-by-frame mappings), while unsupervised alignment methods provide alignments only from pairs of sequences. A common solution for unsupervised alignment methods is alternating two-stage optimization: (1) learning discrepancy between frames and (2) optimizing alignment path using the discrepancy. Iterative motion warping (IMW) also employs DTW to obtain alignments by setting a linear projection for each frame in order to align motion capture data (Hsu et al., 2005). In order to reduce the computational cost in DTW, generalized time warping (GTW) estimates a warping path as a linear combination of predefined monotonic functions (Zhou and De la Torre, 2012). For non-linear projection, Vu et al. (2012) proposed manifold warping (MW), which employed an extension of Laplacian eigenmaps to find a low dimensional manifold of high-dimensional sequences (Belkin and Niyogi, 2003). More recently, Trigeorgis et al. (2016) proposed deep CTW, which employs a neural network for the non-linear projection instead of the linear projection in CTW.

CSTW mainly differs from these methods by considering alignments as a random variable and estimating the parameters with the expectation of the alignment matrix.

Annealing method for the EM algorithm. The EM algorithm is the most common approach to maximum likelihood estimation for a probabilistic model with latent variables, but it often falls into poor local minima. The poor local minima problem often happens in the EM algorithm because of unreliable posterior in the early steps of the alternations (Ueda and Nakano, 1998). In order to mitigate the problem, the deterministic annealing EM algorithm (DAEM) (Ueda and Nakano, 1998) introduces a new posterior with an inverse temperature parameter β, which controls the smoothness of the posterior. DAEM algorithm anneals the new posterior from the uniform distribution ($0 < \beta \ll 1$) to the original posterior ($\beta = 1$). DAEM algorithm can be applied to CSTW, while ACTW anneals $\gamma \to 0$ (i.e., $\beta \to \infty$) to obtain a deterministic alignment instead of the posterior distribution (i.e., $\gamma \to 1$) during the alternations. Note that DAEM algorithm is not directory applicable to CTW, because the posterior distribution of alignment in CTW is the Dirac delta, which is not affected by the inverse temperature when $\beta > 0$.

5. Experiments

In this section, we demonstrate the benefit of CSTW against the following methods: DTW (Sakoe and Chiba, 1978), soft-DTW (Cuturi and Blondel, 2017), DDTW (Keogh and Pazzani, 2001), IMW (Hsu et al., 2005), MW (Vu et al., 2012), CTW (Zhou and De la Torre, 2009), GTW (Zhou and De la Torre, 2012), and DCTW (Trigeorgis et al., 2016).

To implement CTW, we employed python implementations for DTW and CCA[1]. For DDTW, IMW, and GTW, we adopted a MATLAB implementation[2] provided by Zhou and De la Torre (2012), and for MW, a python implementation[3] was used. In order to reproduce the result (Trigeorgis et al., 2016) for DCTW, we built two fully connected neural networks based on an official implementation of DCTW[4]. The two networks have three layer 200-100-100 topology (large network: DCTW L) and two layer 50-50 topology (small network: DCTW S) with LReLU ($\alpha = 0.03$) (Maas et al., 2013). We utilized full-batch optimization with Adam (Kingma and Ba, 2015) (learning rate: 0.0005)[5]. We implemented CSTW and ACTW based on PyTorch (Paszke et al., 2017). For CSTW, we set γ to 1. For ACTW, we initialized γ to 100 and multiplied by 0.9 in each step. In order to compare dependencies of initializations of CTW, CSTW and ACTW, we employed two settings: identity initialization (Zhou and De la Torre, 2009) and UTW initialization (Zhou and De la Torre, 2012) (Section 2.4). In the following experiments, we set the dimension of the latent representation to $d = 2$, convergence tolerance to $\epsilon = 10^{-5}$, and 1teAn alignment algorithm finishes when the change of the objective function is less than the convergence

1. DTW: https://github.com/pierre-rouanet/dtw (accessed: 2018-12), CCA: scikit-learn (Pedregosa et al., 2011)
2. https://github.com/LaikinasAkauntas/ctw (accessed: 2019-1)
3. https://github.com/all-umass/ManifoldWarping (accessed: 2018-2)
4. https://github.com/trigeorgis/DCTW (accessed: 2018-9)
5. We attempted to reproduce the score in (Trigeorgis et al., 2016), but because of the lack of information (e.g., learning rate), we could not. One possible reason is a difference of evaluation method, i.e., in the implementation of DCTW; their models were evaluated during the training, while we split the training and evaluation phases.

tolerance or the number of iterations becomes more than the maximum number.. In the first two experiments, we do not consider DCTW because the ground truths are sufficiently represented by linear models.

5.1. Synthetic data

In the first experiment, we synthesized two multi-dimensional sequences ($X \in \mathbb{R}^{3\times T_x}, Y \in \mathbb{R}^{3\times T_y}$) according to the procedure in (Zhou and De la Torre, 2009, 2012). In the following, we briefly explain the procedure of the input signal generation[6].

$$X = \begin{bmatrix} V_x^\top (Z + \boldsymbol{b}_x) M_x^\top \\ \boldsymbol{e}_x \end{bmatrix}, \; Y = \begin{bmatrix} V_y^\top (Z + \boldsymbol{b}_y) M_y^\top \\ \boldsymbol{e}_y \end{bmatrix},$$

where $Z \in \mathbb{R}^{2\times T}$ was a generated curve signal, $\boldsymbol{e}_x \in \mathbb{R}^{1\times T_x}$ and $\boldsymbol{e}_y \in \mathbb{R}^{1\times T_y}$ were zero-mean Gaussian noise, $\boldsymbol{b}_x$ and $\boldsymbol{b}_y$ were randomly generated translation vectors, and $V_x \in \mathbb{R}^{2\times 2}$ and $V_y \in \mathbb{R}^{2\times 2}$ were randomly generated projection matrices. In order to generate the binary selection matrices $M_x \in \{0,1\}^{T_x\times T}$ and $M_y \in \{0,1\}^{T_y\times T}$, $T > \max(T_x, T_y)$, we first set the identity matrix $\mathbb{I}_T$, and then we uniformly randomly picked T_x and T_y columns, respectively. We set $T = 300$, $T_x = \text{round}(\alpha_x T)$, and $T_y = \text{round}(\alpha_y T)$, where α_x and α_y are random numbers from 0.1 to 0.5. We show an example of synthesized data in Figure 2(*a*).

In order to measure the alignment performances, we employed cosine similarity between alignment matrices. We call this measure *alignment similarity* (S_{align}) obtained as

$$S_{\text{align}}(A_{\text{alg}}, A_{\text{true}}) = \frac{1}{\|A_{\text{alg}}\|_F \|A_{\text{true}}\|_F} \langle A_{\text{alg}}, A_{\text{true}} \rangle ,$$

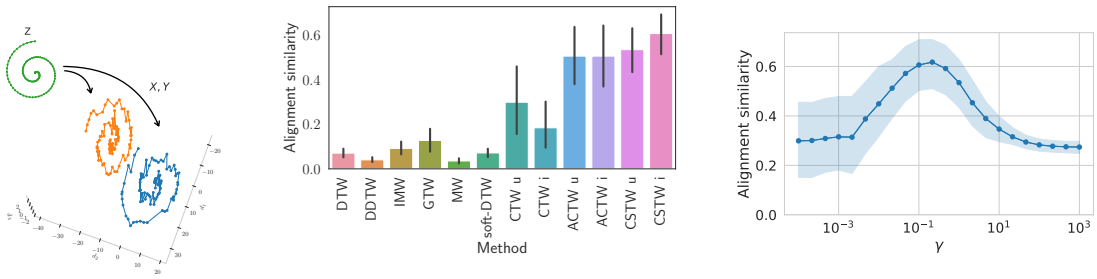
where A_{alg} and $A_{\text{true}} \in \mathcal{A}$ are alignment matrices of algorithm output and ground truth, respectively. We computed A_{true} in the same manner as in (Zhou and De la Torre, 2012).

For a statistical comparison, we synthesized data 10 times with different random seeds. In Figure 2(*b*), we show a bar plot of the mean alignment similarities with 90% bootstrapping confidence intervals. The result shows that our methods outperform the previous methods, and in particular, CSTW (identity initialization) obtains the best performance in this task. In Figure 2(*c*), we show the dependency of the alignment similarity on γ values in CSTW (UTW initialization). CSTW demonstrated poor performance for γ values close to 0 (i.e., CTW), as well as γ values over 10 (considering all warping paths uniformly). This result indicates that the soft-alignment with an appropriate γ value gains better performance than CTW.

5.2. Alignments Far from Diagonal

In this subsection, we demonstrate the superiority of CSTW and ACTW for a more difficult task, where true alignment matrices are far from diagonal matrices. For the experiment, we generated data using the same procedure as in the previous subsection, except M_x and M_y, which were generated by non-uniform random choice. We split the original sequence Z into three intervals and randomly generated their weights with a uniform distribution from 0.1

6. In order to generate the input sequences, we utilized the author's MATLAB implementation, which can be obtained at https://github.com/LaikinasAkauntas/ctw (accessed: 2019-1).



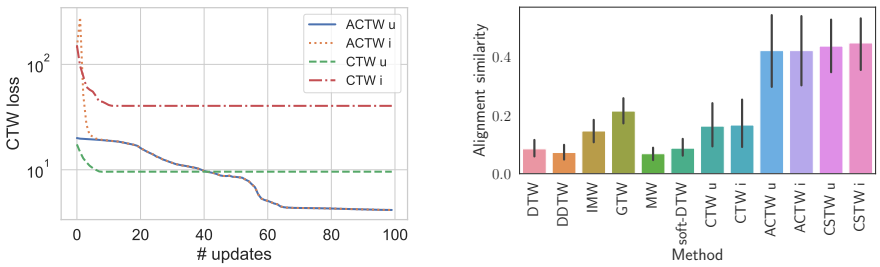
(a) (b) (c)

Figure 2: (a) The synthesized sequences X and Y and their source sequence Z. (b) Comparison of the mean alignment similarities for several methods with 90% bootstrapping confidence intervals in the task using synthetic data (Section 5.1). The suffixes i and u denote the initialization method: identity initialization and UTW initialization, respectively. (c) The average alignment similarities for CSTW (UTW initialization) for different γ values, together with the 90% bootstrapping confidence intervals.

to 1. Then, we randomly chose T_x and T_y columns from the identity matrix $\mathbb{I}_T$ according to the weights. Figure 3(a) depicts examples of the loss values of CTW and ACTW during 100 updates. In order to compare them, we employ the loss function of CTW even when we evaluate ACTW using projection matrices U and V in each step. This result indicates that ACTW avoids the poor local minima problem regardless of the initialization methods. In Figure 3(b), we show a bar plot of the mean alignment similarities for 10 time trials with different random seeds. Figure 3(b) indicates that ACTW and CSTW outperformed the other methods regardless of the initialization, even in the situation where the true alignment matrices are far from diagonals. The projected sequences are illustrated in Figure 4, which show that the other methods, including CTW and GTW, are highly affected by the noise signals, while CSTW is insensitive to noise.

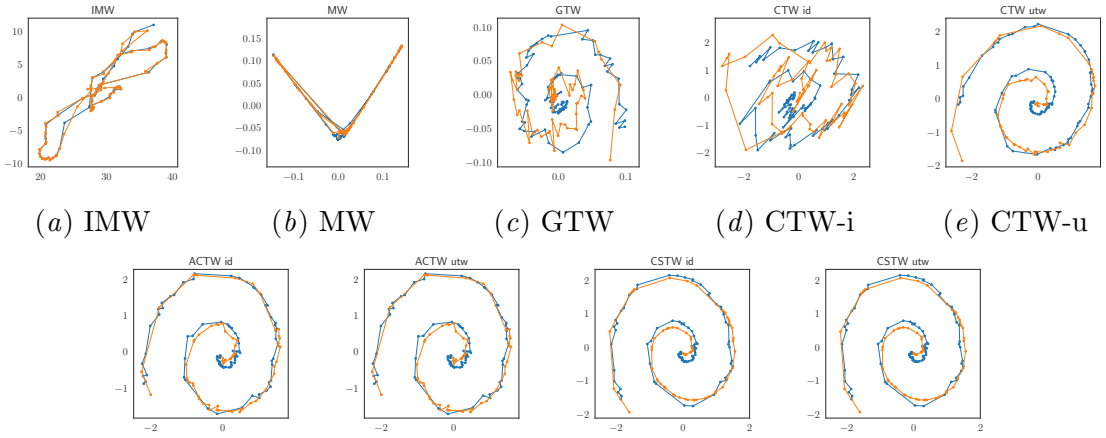
5.3. Alignment of Facial Action Units

In this subsection, we demonstrate the alignment performance of CSTW and ACTW using real-world data. We employed a dataset used by Trigeorgis et al. (2016), which was a subset of the MMI Facial Expression Dataset (Pantic et al., 2005). The dataset contains 10 movies of people smiling. Each face starts from a neutral face, then smiles, and then returns to a neutral face. Each frame of the dataset has one of four labels: **neutral**, **onset**, **apex**, and **offset**. The **neutral** label is corresponding to a face whose muscles related to smiling are inactive. The **apex** label is assigned to a face which has a peak intensity of the muscles. The **onset** label and **offset** label are transient states corresponding to **neutral** $\rightarrow$ **apex** and **apex** $\rightarrow$ **neutral**, respectively. For the inputs of the alignment methods, we employed the same preprocessing (Trigeorgis et al., 2016), which converts each frame to grayscale and crops 40×40 pixels in order to center the face. For example,



(a) (b)

Figure 3: (a) Comparison of CTW and ACTW in terms of the CTW loss. (b) Comparison of the mean alignment similarities for several methods with 90% bootstrapping confidence intervals in the task where alignments are far from diagonal (Section 5.2). The suffixes i and u denote the initialization method, identity initialization, and UTW initialization, respectively.



(a) IMW (b) MW (c) GTW (d) CTW-i (e) CTW-u

(f) ACTW-i (g) ACTW-u (h) CSTW-i (i) CSTW-u

Figure 4: The projection results when true alignments are far from diagonal (Section 5.2)

we show some frames of the preprocessed inputs in Figure 5. Furthermore, we reduced the dimensionality of each frame preserving 99% of the contribution (68 dimensions) using principal component analysis for the movies. In this task, we aligned 45 pairs of movies using the alignment methods, including the batch settings of CSTW and ACTW, which assume a pair of projection matrices for all movies. We employed the UTW initialization for initialization of CTW, CSTW, and ACTW because CTW works better with the UTW initialization from Figure 2(*b*). For DCTW, we conducted the evaluation ten times using different random seeds because they were randomly initialized, unlike the other methods.

In order to evaluate the alignment similarity (S_{align}), true alignment paths are necessary; however, they are unknown in this experiment. For the evaluation, we employed two criteria: *label-matching-ratio* and *DCTW-score*. Similar to the alignment similarity,

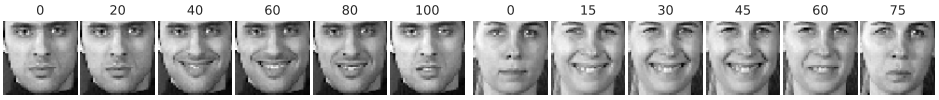


Figure 5: Examples of preprocessed movies that record smiling faces. The numbers displayed on the images are frame numbers, which indicate that the timings of the smiling are different according to the person.

we define label-matching-ratio, which counts number of same labels in aligned frames, as $R_{\text{match}}(A_{\text{alg}}, A_{\text{match}}) := \frac{1}{\|A_{\text{alg}}\|_F^2} \langle A_{\text{alg}}, A_{\text{match}} \rangle$. $A_{\text{match}} \in \{0,1\}^{T_x, T_y}$ is a label matching matrix, which is defined below.

$$[A_{\text{match}}]_{t,s} := \begin{cases} 1 & \text{if } \phi_x^{(t)} = \phi_y^{(s)}, \\ 0 & \text{otherwise,} \end{cases}$$

where $\phi_x \in \{1:K\}^{T_x}$ and $\phi_y \in \{1:K\}^{T_y}$ are two label sequences. K is the number of labels ($K = 4$ in this task). DCTW-score (Trigeorgis et al., 2016) is defined as follows.

$$R_{\text{DCTW}} := \frac{1}{K} \sum_{k=1}^{K} \frac{\left|\{t \mid (\phi_x W_x (A)^\top)^{(t)} = k\} \cap \{s \mid (\phi_y W_y (A)^\top)^{(s)} = k\}\right|}{\left|\{t \mid (\phi_x W_x (A)^\top)^{(t)} = k\} \cup \{s \mid (\phi_y W_y (A)^\top)^{(s)} = k\}\right|}$$

DCTW-score can be regarded as a mean value of label-matching-ratios calculated for each of the labels. In order to evaluate DCTW-score for CSTW (or ACTW), which do not provide warping matrices explicitly, we employed DTW with the input $U^\top X$ and $V^\top Y$, where U, V are solutions of CSTW (or ACTW). The label-matching-ratio and DCTW-score are shown in Table 1. The results show that our methods, including batch CSTW and batch ACTW, outperform the existing methods, especially batch CSTW, which obtains the best scores in both label-matching-ratio and DCTW-score.

6. Conclusion

In this paper, we proposed CSTW as an unsupervised alignment framework for high-dimensional sequences. To avoid poor local minima, we treat an alignment as a random variable instead of deterministically, as in previous studies. More concretely, CSTW models the expectation of the alignment matrix instead of considering only the deterministic alignment in each step of the two-stage optimization. We also proposed an efficient and tractable optimization for CSTW, which utilizes soft-DTW with the alignment kernel tricks. CSTW controls the uniformness of the expectation of the alignment matrix by the temperature parameter and allows us to obtain a deterministic alignment by the annealing method, avoiding poor local minima. We demonstrated the advantages of CSTW over the existing works in the alignment of synthesized data and facial expressions. The results showed that CSTW outperforms the current state-of-the-art in terms of alignment similarities and label-matching-ratio regardless of how the parameters are initialized.

Table 1: Comparison of the label-matching ratio for several methods for the alignment of facial action units (Section 5.3). The suffixes L and S) denote a large network and small network, respectively. We present means and standard deviations of ten trials for DCTW L and DCTW S, which are randomly initialized.

	R_{match}	R_{DCTW}
DTW	0.617	0.505
soft-DTW	0.631	0.505
DDTW	0.600	0.486
IMW	0.611	0.493
CTW	0.611	0.496
GTW	0.622	0.504
MW	0.535	0.426
ACTW	**0.634**	**0.530**
CSTW	0.627	0.519
DCTW L (batch)	0.534 ($\pm$ 0.032)	0.427 ($\pm$ 0.031)
DCTW S (batch)	0.531 ($\pm$ 0.082)	0.433 ($\pm$ 0.065)
ACTW (batch)	0.638	0.529
CSTW (batch)	**0.666**	**0.554**

References

John Aach and George M Church. Aligning gene expression time series with time warping algorithms. *Bioinformatics*, 17(6):495–508, 2001.

Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

Chien-Yi Chang, De-An Huang, Yunan Sui, Li Fei-Fei, and Juan Carlos Niebles. Discriminative Differentiable Dynamic Time Warping for Weakly Supervised Action Alignment and Segmentation. In *CVPR*, 2019.

Marco Cuturi and Mathieu Blondel. Soft-DTW: a differentiable loss function for time-series. In *ICML*, pages 894–903, 2017.

Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–22, 1977.

Pelin Dogan, Boyang Li, Leonid Sigal, and Markus Gross. A Neural Multi-sequence Alignment TeCHnique (NeuMATCH). In *CVPR*, pages 8749–8758, 2018.

Alexei Gritai, Yaser Sheikh, Cen Rao, and Mubarak Shah. Matching trajectories of anatomical landmarks under viewpoint, anthropometric and temporal transforms. *International journal of computer vision*, 84(3):325–343, 2009.

David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

Eugene Hsu, Kari Pulli, and Jovan Popović. Style Translation for Human Motion. *ACM Transactions on Graphics*, 24(3):1082–1089, 2005.

Eamonn J Keogh and Michael J Pazzani. Derivative dynamic time warping. In *SDM*, pages 1–11. SIAM, 2001.

Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, volume 30, page 3, 2013.

Maja Pantic, Michel Valstar, Ron Rademaker, and Ludo Maat. Web-based database for facial expression analysis. In *ICME*, pages 317–321, 2005.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pyTorch. In *NIPS Workshop*, 2017.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1): 43–49, 1978.

Yaser Sheikh, Mumtaz Sheikh, and Mubarak Shah. Exploring the space of a human action. In *ICCV*, volume 1, pages 144–149. IEEE, 2005.

George Trigeorgis, Mihalis A Nicolaou, Stefanos Zafeiriou, and Bjorn W Schuller. Deep canonical time warping. In *CVPR*, pages 5110–5118, 2016.

Naonori Ueda and Ryohei Nakano. Deterministic annealing EM algorithm. *Neural networks*, 11(2):271–282, 1998.

Hoa T Vu, CJ Carey, and Sridhar Mahadevan. Manifold warping: manifold alignment over time. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pages 1155–1161. AAAI Press, 2012.

Feng Zhou and Fernando De la Torre. Canonical time warping for alignment of human behavior. In *NIPS*, pages 2286–2294. Curran Associates Inc., 2009.

Feng Zhou and Fernando De la Torre. Generalized time warping for multi-modal alignment of human motion. In *CVPR*, pages 1282–1289. IEEE, 2012.