Learning Distance for Sequences by Learning a Ground Metric

Bing Su [1] **Ying Wu** [2]

Abstract

Learning distances that operate directly on multi-dimensional sequences is challenging because such distances are structural by nature and the vectors in sequences are not independent. Generally, distances for sequences heavily depend on the ground metric between the vectors in sequences. We propose to learn the distance for sequences through learning a ground Mahalanobis metric for the vectors in sequences. The learning samples are sequences of vectors for which how the ground metric between vectors induces the overall distance is given, and the objective is that the distance induced by the learned ground metric produces large values for sequences from different classes and small values for those from the same class. We formulate the metric as a parameter of the distance, bring closer each sequence to an associated virtual sequence w.r.t. the distance to reduce the number of constraints, and develop a general iterative solution for any ground-metric-based sequence distance. Experiments on several sequence datasets demonstrate the effectiveness and efficiency of our method.

1. Introduction

In many domains, the data are naturally in the form of multi-dimensional sequences. Pairwise distance measures between sequences serve as a proxy to manipulate the structured sequences so that any metric-based machine learning methods can be directly applied. The performances of metric-based algorithms such as the k-nearest neighbor classifier (k-NN) heavily depend on the quality of the distance measures. Therefore, learning distances for sequences from data is especially appealing.

[1]Science & Technology on Integrated Information System Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing, China [2]Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL, USA. Correspondence to: Bing Su <subingats@gmail.com>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

Although metric learning has achieved a considerable maturity level both in practice and in theory (Bellet et al., 2013), propagating these advances to sequence data is not trivial. This is because most existing metric learning methods are developed for static data which are in the form of "flat" feature vectors. An acquiescent assumption is that these vector data are independent and identically distributed, but the elements in sequences exhibit temporal relationships. Much less work has been devoted to metric learning for sequence data, and most of them actually encode each sequence into a vector and simply build the metric upon the vectors, which cannot capture the alignments or relationships among the vectors in sequences explicitly and may lose significant temporal information. Learning distances that operate directly on sequences is challenging, because such distances are naturally structural and combinatorial. Specifically, the major difficulties lie in two aspects.

First, different sequences vary in length, evolution speed, and local temporal duration. Different distance measures for sequences such as (Sakoe & Chiba, 1978; Su & Hua, 2017) perform temporal alignments to eliminate the local temporal discrepancies. Inferring the alignments depends on the metric between elements in sequences. For a specific sequence pair, their alignments cannot be inferred before the underlying metric is learned. Therefore, the objective of learning distances for sequences generally involves latent alignment structures when formulating the distances as a function of the unknown metric, and hence is difficult to manipulate and optimize.

Second, most metric learning methods employ the must-link/cannot-link constraints over positive/negative pairs (Xing et al., 2003; Davis et al., 2007) or the relative constraints over triplets (Schultz & Joachims, 2004; Weinberger & Saul, 2009). The number of constraints is quadratic or cubic in the number of training samples, which easily becomes intractable when more training samples are available. One heuristic is to mine only a subset of the most informative constraints, but such mining is not trivial. Because of the complexity of measuring distances for sequences, the cost of constructing these constraints is larger and it can be computationally prohibitive to update the subset of constraints with the update of the metric during the optimization. Reducing the number of constraints is more crucial for sequence data.

In this paper, we propose a metric learning framework for sequence data to tackle these issues. We unify a wide range of distance measures for sequences into a formulation as a function of the *ground metric* for elements in sequences. The final distances are *meta-distances* built upon the ground metric by inferring the temporal alignments among the element pairs. Thanks to such parameterization, we show that various distances for sequences are amenable to learn via learning a Mahalanobis distance (Mahalanobis, 1936) as the ground metric. More specifically, we treat the alignments as latent variables of the meta-distance function that takes the ground metric as an argument, since inferring them also depends on the ground metric. The formulation of the objective for learning the ground metric incorporates latent variables. We develop an iterative alternating descent algorithm that achieves joint optimization of the metric and the latent alignments, which can be instantiated with any meta-distances using various alignment inference methods.

Another contribution of our work is the extension of the *regressive virtual metric learning (RVML)* (Perrot & Habrard, 2015) method for reducing the number of constraints. RVML requires a linear number of constraints by moving each sample to its corresponding pre-defined virtual point. Our method extends RVML in three ways: (1) it associates each training sequence with a virtual sequence and provides two solutions to generate virtual sequences; (2) it allows to minimize the distances between the training sequences and the virtual sequences w.r.t. various meta-distance measures; (3) it involves latent alignment structures and requires to learn the ground metric and the latent structures simultaneously.

2. Related Work

Differences with conventional metric learning. Most classical metric learning methods for vector data employ either the pair-based or the triplet-based constraints. The pair-based must-link/cannot-link side information was introduced in the seminal work of (Xing et al., 2003), and then widely used in a lot of methods such as *information-theoretic metric learning (ITML)* (Davis et al., 2007), regularized distance metric learning (Jin et al., 2009), and sparse distance metric learning (Qi et al., 2009). Generally, the nearest neighbors based methods, such as neighbourhood component analysis (Goldberger et al., 2005), maximally collapsing metric learning (Globerson & Roweis, 2006), *large margin nearest neighbors (LMNN)* (Weinberger et al., 2006; Weinberger & Saul, 2009), and *sparse compositional metric learning (SCML)* (Shi et al., 2014), used the triple-based constraints to force the distances of each instance to its target neighbors relatively smaller than those to impostors. RVML (Perrot & Habrard, 2015) introduced the virtual point based constraints. Propagating these advances for vector representations to sequence data is not trivial.

Differences with edit distance learning and kernel learning for sequences. In (Bellet et al., 2011; 2012; Paaßen et al., 2018), the string edit distance was learned by learning the cost matrix for edit operations. The elements in sequences were symbols from a fixed finite alphabet and the edit operations for each sequence pair were fixed. In (Cortes et al., 2008), weighted finite-state transducers based rational kernels (Cortes et al., 2004) were learned to measure the similarities between sequences, where the elements were also restricted to a finite alphabet. It is difficult to apply these methods to unconstrained sequences, where the elements are continuous real vectors rather than discrete symbols and the number of all possible elements is infinite. In contrast, our method learns the Mahalanobis distance for real vectors and the latent alignments jointly.

Differences with existing metric learning methods for optimal transport (OT). In (Cuturi & Avis, 2014), the OT distance for histograms was learned by learning the ground metric based on side supervision on specific similarity coefficients of all histogram pairs, where the supporting points for all histograms were fixed. This method cannot be applied to unconstrained sequences because it directly learn a ground matrix containing all pairwise distances for the supporting points. In (Huang et al., 2016), the supervised word mover's distance (SWMD) learned OT distances for documents each consists of a set of unordered words by learning the ground metric, where the words are in a fixed finite dictionary and the weights for these fixed words were learned together. It minimized the leave-one-out kNN error by a gradient-based solution. In contrast, our method minimizes the regression-based loss by non-gradient descent optimization, and is applicable to unconstrained multidimensional sequences where the elements lie in a continuous space.

Differences with existing metric learning methods for sequences. In (Garreau et al., 2014), the ground-truth alignments were used for learning the metric. In contrast, ground-truth alignments are not available and our method learns the ground metric and the alignments jointly. In (Zhao et al., 2016) and (Mei et al., 2016), Mahalanobis distances were learned as ground metrics to enhance the *dynamic time warping (DTW)* distance, where the DTW alignments for all sequence pairs were fixed by using the Euclidean metric. The solutions were sub-optimal since the alignments may change with the learned matrices. In contrast, our method achieves joint optimization for the metric and the latent alignments. In (Mei et al., 2014), LDMLT iteratively updated the ground Mahalanobis metric with the triplets constraints and updated the alignments by DTW to build dynamic triplets. However, the iterative solution is not guaranteed to converge because updating the align-

ments by DTW does not guarantee to decrease the objective of the logDet divergence based metric learning. In contrast, our method is guaranteed to converge, trains much faster, and is applicable to different sequence distances.

Differences with recurrent neural network (RNN) based deep metric learning. Deep metric learning methods (Yi et al., 2014; Song et al., 2016; Che et al., 2017) were typically deep extensions of classical metric learning methods and require large amounts of training sequences. Some works (Bayer et al., 2012; Mokbela et al., 2015) actually encoded the sequences into fixed-length vectors and build metrics upon vectors. In contrary, our method is applied to elements in sequences and the alignments can be explicitly inferred, which are crucial in some applications. Moreover, our method can be applied before sequences are fed into those RNN-based methods.

3. A Unified Perspective on Distance Measures for Sequences

In this section, we present a unified formulation of the distance measures for sequence data and establish the connections between the formulation and two distance measures. Connections to some other distance measures are presented in the supplementary file.

Let Ω be a space and $d(\boldsymbol{M}) : \Omega \times \Omega \to \mathbb{R}$ be the metric on this space, which is parameterized by $\boldsymbol{M}$. Given two sequences $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{L_X}] \in \Omega^{L_X}$ and $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_{L_Y}] \in \Omega^{L_Y}$ with lengths L_X and L_Y, respectively, whose elements $\boldsymbol{x}_i, i = 1, \cdots, L_X$ and $\boldsymbol{y}_j, j = 1, \cdots, L_Y$ are sampled in Ω, the distance between them can be formulated as

$$g_{\boldsymbol{M}}(\boldsymbol{X}, \boldsymbol{Y}) = \langle \boldsymbol{T}^*, \boldsymbol{D}(\boldsymbol{M}) \rangle , \tag{1}$$

where $\langle \boldsymbol{T}, \boldsymbol{D} \rangle = tr(\boldsymbol{T}^T \boldsymbol{D})$ is the Frobenius dot product.

$$\boldsymbol{D}(\boldsymbol{M}) := [d(\boldsymbol{M}, \boldsymbol{x}_i, \boldsymbol{y}_j)]_{ij} \in \mathbb{R}^{L_X \times L_Y} \tag{2}$$

is the cost matrix of all pairwise vector-wise distances between elements in $\boldsymbol{X}$ and $\boldsymbol{Y}$, whose element $\boldsymbol{D}(\boldsymbol{M})_{ij} = d(\boldsymbol{M}, \boldsymbol{x}_i, \boldsymbol{y}_j)$ is the distance between $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ w.r.t. the metric $d(\boldsymbol{M})$. $\boldsymbol{T}^*$ is a matrix indicating the correspondence relationship, where $t^*_{i,j} = \boldsymbol{T}^*(i, j)$ actually measures whether or how the pair $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ corresponds to the same temporal position or structure. Ideally, only the differences between those elements within the same temporal positions reflect the differences between the entire sequences. However, due to the different sampling rates, the non-uniform evolution speeds of elements, local temporal distortions, etc, different sequences have different lengths and exhibit local temporal differences, so the i-th element in $\boldsymbol{X}$ and the i-th element in $\boldsymbol{Y}$ may not correspond to the same relative position. $\boldsymbol{T}^*$ is used to align the elements corresponding to the same temporal structure or position. Generally, the determination of $\boldsymbol{T}^*$ can be formulated as

$$\boldsymbol{T}^* = \underset{\boldsymbol{T} \in \boldsymbol{\Phi}}{arg\,min} \langle \boldsymbol{T}, \boldsymbol{D}(\boldsymbol{M}) \rangle + \mathscr{R}(\boldsymbol{T}), \tag{3}$$

where $\boldsymbol{\Phi}$ is the feasible set of $\boldsymbol{T}$, which is a subset of $\mathbb{R}^{L_X \times L_Y}$ with some constraints, and $\mathscr{R}(\boldsymbol{T})$ is a regularization term on $\boldsymbol{T}$. Different distance measures for sequences differ in the constraints imposed to the feasible set, the regularization term, and the optimization or inference method.

DTW (Sakoe & Chiba, 1978). DTW calculates an optimal alignment between two sequences with three constraints: boundary, continuity, and monotonicity. In the unified formulation, DTW restricts $\boldsymbol{T}$ to be a binary matrix, in which $t_{i,j} = 1$ if $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ are aligned and $t_{ij} = 0$ otherwise. DTW instantiates the formulation (3) by setting:

$$\begin{aligned} &\mathscr{R}(\boldsymbol{T}) = 0; \\ &\boldsymbol{\Phi} = \{\boldsymbol{T} \in \{0,1\}^{L_X \times L_Y} | \boldsymbol{T}_{1,1} = 1, \boldsymbol{T}_{L_X, L_Y} = 1; \\ &\quad \boldsymbol{T}\mathbf{1}_{L_Y} > \mathbf{0}_{L_X}, \boldsymbol{T}^T\mathbf{1}_{L_X} > \mathbf{0}_{L_Y}; \\ &\quad if\ t_{i,j} = 1, then\ t_{i-1,j+1} = 0, t_{i+1,j-1} = 0, \\ &\quad \forall 1 < i < L_X, 1 < j < L_Y\} \end{aligned} , \tag{4}$$

where $\mathbf{1}_b$ and $\mathbf{0}_b$ are the b-dimensional vectors with all one and zero elements ,respectively, and "$>$" should be understood as element-wise. DTW solves Eq. (3) with constraints (4) via dynamic programming.

Order-preserving Wasserstein Distance (OPW) (Su & Hua, 2017; 2018). OPW casts sequence alignment as the OT problem. It imposes two regularization terms to the original OT problem to preserve the global temporal information. The first regularization favors $\boldsymbol{T}$ with large *inverse difference moment* which is calculated as

$$I(\boldsymbol{T}) = \sum_{i=1}^{L_X} \sum_{j=1}^{L_Y} \frac{t_{ij}}{\left(\frac{i}{L_X} - \frac{j}{L_Y}\right)^2 + 1}, \tag{5}$$

The second regularization encourages the distribution of $\boldsymbol{T}$ to be similar to a prior distribution $\boldsymbol{P}$:

$$p_{ij} := \boldsymbol{P}(i, j) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{\ell^2(i,j)}{2\sigma^2}}, \tag{6}$$

where $\ell(i, j) = \frac{|i/L_X - j/L_Y|}{\sqrt{1/{L_X}^2 + 1/{L_Y}^2}}$. Both regularization terms encourage alignments between elements with similar relative temporal positions and restrict the matching between elements that are far away temporally. OPW instantiates the formulation (3) by setting:

$$\begin{aligned} &\mathscr{R}(\boldsymbol{T}) = \lambda_1 I(\boldsymbol{T}) + \lambda_2 KL(\boldsymbol{T}||\boldsymbol{P}); \\ &\boldsymbol{\Phi} = \{\boldsymbol{T} \in \mathbb{R}_+^{L_X \times L_Y} | \boldsymbol{T}\mathbf{1}_{L_Y} = \tfrac{1}{L_X}\mathbf{1}_{L_X}, \\ &\quad \boldsymbol{T}^T\mathbf{1}_{L_X} = \tfrac{1}{L_Y}\mathbf{1}_{L_Y}\} \end{aligned} , \tag{7}$$

where λ_1 and λ_2 are preset balancing coefficients, and $KL(\boldsymbol{T}||\boldsymbol{P})$ is the Kullback-Leibler divergence. OPW

solves Eq.(3) with constraints (7) by the Sinkhorn's matrix scaling algorithm. Each element t_{ij}^* in the learned $\boldsymbol{T}^*$ can be viewed as the probability of aligning $\boldsymbol{x}_i$ to $\boldsymbol{y}_j$.

We observe that these distances actually share the common formulation and can be considered as *meta-distances* built on $d(\boldsymbol{M})$, although they have different motivations. For these distances, the determination of $\boldsymbol{T}^*$ depends on the metric $d(\boldsymbol{M})$. In the literature (Rubner et al., 2000; Cuturi & Avis, 2014), the metric is called the *ground metric*. We follow this name to distinguish it with the meta-distance for sequences.

4. Methodology

4.1. Problem

With the unified formulation (1) and (3), we view the meta-distance as a function of the ground metric parameterized by $\boldsymbol{M}$. The goal of our method is to learn a ground metric $\boldsymbol{M}$ resulting in a meta-distance $g_{\boldsymbol{M}}(\boldsymbol{X}, \boldsymbol{Y})$ (1), such that the meta-distances between sequences from different classes are large, and those between sequences from the same class are small. We learn a squared Mahalanobis-like distance (Mahalanobis, 1936) as the ground metric, i.e.,

$$d(\boldsymbol{M}, \boldsymbol{x}_i, \boldsymbol{y}_j) = (\boldsymbol{x}_i - \boldsymbol{y}_j)^T \boldsymbol{M} (\boldsymbol{x}_i - \boldsymbol{y}_j), \tag{8}$$

where $\boldsymbol{M}$ is a positive semi-definite matrix and can be decomposed as $\boldsymbol{M} = \boldsymbol{W}\boldsymbol{W}^T$, $\boldsymbol{W} \in \mathbb{R}^{b \times b'}$ and b' is greater than or equal to the rank of $\boldsymbol{M}$. This is equivalent to transform all elements $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ with a linear projection $\boldsymbol{W}$.

Specially, let $\{\boldsymbol{X}^n, z^n\}_{n=1}^N$ be a set of N training sequences, where $\boldsymbol{X}^n = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{L^n}] \in \mathbb{R}^{b \times L^n}$ is the n-th sequence with length L^n. Different sequences may have different lengths. $\boldsymbol{x}_i, i = 1, \cdots, L^n$ are sampled in $\mathbb{R}^b$, and z^n is the class label of $\boldsymbol{X}^n$. We are interested in learning a meta-distance $g_{\boldsymbol{M}}(\boldsymbol{X}^n, \boldsymbol{X}^{n'})$ with the form of Eq. (1) by learning $\boldsymbol{W}$ from the training set, such that the resulting $g_{\boldsymbol{M}}(\boldsymbol{X}^n, \boldsymbol{X}^{n'}) = g_{\boldsymbol{I}}(\boldsymbol{W}^T\boldsymbol{X}^n, \boldsymbol{W}^T\boldsymbol{X}^{n'})$ captures the idiosyncrasy of sequence data and better separates sequences from different classes, where $g_{\boldsymbol{I}}$ means that $\boldsymbol{M} = \boldsymbol{I}$ when constructing Eq. (2): $\boldsymbol{D}_{\boldsymbol{I}}(\boldsymbol{W}) = [d(\boldsymbol{I}, \boldsymbol{W}^T\boldsymbol{x}_i, \boldsymbol{W}^T\boldsymbol{y}_j)]_{ij}$.

The difficulty largely lies in the fact that in Eq. (1), $\boldsymbol{T}^*$ is not fixed, but needs to be inferred by optimizing Eq.(3) for each sequence pair. The inference of $\boldsymbol{T}^*$ also heavily depends on $\boldsymbol{W}$. Once $\boldsymbol{W}$ changes, $\boldsymbol{T}^*$ for each sequence pair changes accordingly. Also, for any sequence pair, the corresponding optimal alignment $\boldsymbol{T}^*$ needs to be inferred individually. The cost of constructing a single must-link/cannot-link or relative constraint for sequence distance is much larger than for vector distance. Therefore, it can be computationally prohibitive to learn $\boldsymbol{W}$ with such constraints whose number is quadratic or cubic with the number of training sequences.

4.2. Objective and Optimization

RVML (Perrot & Habrard, 2015) introduces a new kind of constraints that moving each sample to its corresponding pre-defined virtual point. Compared with must-link/cannot-link and relative constraints, the number of such virtual point-based constraints is greatly reduced since it is linear with the number of samples. We extend RVML to sequence data by associating a virtual sequence instead of a virtual point for each sequence sample. Let $\boldsymbol{V}^n = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_{l^n}] \in \mathbb{R}^{b' \times l^n}$ be the virtual sequence related with $\boldsymbol{X}^n$. b' and l^n are the dimensionality and the number of elements in $\boldsymbol{V}^n$, respectively, which may not equal to those in $\boldsymbol{X}^n$. $\boldsymbol{V}^n$ is a function of $\boldsymbol{X}^n$ and z^n: $\boldsymbol{V}^n = f(\boldsymbol{X}^n, z^n)$. We first assume that the virtual sequences for all training sequences have been obtained. The goal is to learn a transformation $\boldsymbol{W}$ by minimizing the meta-distances between the training sequences and their associated virtual sequences, i.e.,

$$\begin{aligned} &\min_{\boldsymbol{W}} \tfrac{1}{N} \sum_{n=1}^{N} g_{\boldsymbol{I}}(\boldsymbol{W}^T\boldsymbol{X}^n, \boldsymbol{V}^n) + \beta\|\boldsymbol{W}\|_{\mathcal{F}}^2 \\ &= \tfrac{1}{N} \sum_{n=1}^{N} \langle \boldsymbol{T}^{n*}, \boldsymbol{D}_{\boldsymbol{I}}^n(\boldsymbol{W}) \rangle + \beta\|\boldsymbol{W}\|_{\mathcal{F}}^2 \\ &s.t.\ \boldsymbol{T}^{n*} = \underset{\boldsymbol{T} \in \boldsymbol{\Phi}}{arg\min} \langle \boldsymbol{T}^n, \boldsymbol{D}_{\boldsymbol{I}}^n(\boldsymbol{W}) \rangle + \mathscr{R}(\boldsymbol{T}^n) \end{aligned} \tag{9}$$

where $\|\cdot\|_{\mathcal{F}}$ is the Frobenius norm and β is a hyper-parameter that balances the two items.

The underlying $\boldsymbol{T}^{n*}, n = 1, \cdots, N$ for all training-virtual sequence pairs depend on the variable $\boldsymbol{W}$. We treat them as latent structures. In Eq.(9), if $\mathscr{R}(\boldsymbol{T})$ does not depend on $\boldsymbol{W}$, the inferences over $\boldsymbol{T}^{n*}, n = 1, \cdots, N$ in the constraints are actually minimizing the same objective as the optimization over $\boldsymbol{W}$. This allows us to jointly learn $\boldsymbol{W}$ and $\boldsymbol{T}^{n*}, n = 1, \cdots, N$ by optimizing the following objective:

$$\min_{\boldsymbol{W}, \boldsymbol{T}^n} \frac{1}{N} \sum_{n=1}^{N} \langle \boldsymbol{T}^n, \boldsymbol{D}_{\boldsymbol{I}}^n(\boldsymbol{W}) \rangle + \beta\|\boldsymbol{W}\|_{\mathcal{F}}^2 + \mathscr{R}(\boldsymbol{T}^n). \tag{10}$$

The objective function Eq. (10) is not jointly convex on $\boldsymbol{W}$ and $\boldsymbol{T}^n, n = 1, \cdots, N$. We minimize it by alternatively updating the metric and the latent alignments. We first fix $\boldsymbol{T}^n, n = 1, \cdots, N$ and update $\boldsymbol{W}$. In this case, the regularization term $\mathscr{R}(\boldsymbol{T})$ can be discarded and the objective can be reformulated as

$$\begin{aligned} &\tfrac{1}{N} \sum_{n=1}^{N} \langle \boldsymbol{T}^n, \boldsymbol{D}_{\boldsymbol{I}}^n(\boldsymbol{W}) \rangle + \beta\|\boldsymbol{W}\|_{\mathcal{F}}^2 \\ &= \tfrac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{L^n} \sum_{j=1}^{l^n} t_{ij}^n \|\boldsymbol{W}^T\boldsymbol{x}_i^n - \boldsymbol{v}_j^n\|_2^2 + \beta\|\boldsymbol{W}\|_{\mathcal{F}}^2 \end{aligned}. \tag{11}$$

Minimizing Eq.(11) is a weighted regression problem, which admits a closed form solution:

$$\boldsymbol{W}^* = \boldsymbol{A}^{-1} \Big(\sum_{n=1}^{N} \sum_{i=1}^{L^n} \sum_{j=1}^{l^n} t_{ij}^n \boldsymbol{x}_i^n \boldsymbol{v}_j^{nT} \Big), \tag{12}$$

Algorithm 1 RVSML

1: **Input:** A set of training sequences $\{\boldsymbol{X}^n\}_{n=1}^N$ and the associated virtual sequences $\{\boldsymbol{V}^n\}_{n=1}^N$
2: **Output:** the transformation $\mathbf{W}$
3: Initialize the alignment matrices $\boldsymbol{T}^n, n = 1, \cdots, N$ for all training-virtual sequence pairs.
4: **while** $\mathbf{W}$ has not converged **do**
5: Update $\mathbf{W}$ by Eq. (11)
6: **for** $n = 1, \cdots, N$ **do**
7: Update $\boldsymbol{T}^n$ by optimizing Eq. (14)
8: **end for**
9: **end while**

where

$$\boldsymbol{A} = \sum_{n=1}^{N} \sum_{i=1}^{L^n} \sum_{j=1}^{l^n} t_{ij}^n \boldsymbol{x}_i^n \boldsymbol{x}_i^{nT} + \beta N \boldsymbol{I}. \tag{13}$$

This solution can be simply derived by setting the derivative of Eq.(11) to 0.

We then update $\boldsymbol{T}^n, n = 1, \cdots, N$ by fixing $\boldsymbol{W}$. In this case, the matrix $\boldsymbol{D}_{\boldsymbol{I}}^n(\boldsymbol{W})$ consisting of all pairwise squared Euclidean distances between $\boldsymbol{W}\boldsymbol{x}_i^n$ and $\boldsymbol{v}_j^n$ is also fixed, and the irrelevant regularization term $\|\boldsymbol{W}\|_{\mathcal{F}}^2$ can be discarded. We further observe that the optimizations of $\boldsymbol{T}^n$ for $n = 1, \cdots, N$ are independent. Therefore, we can solve them separately by applying the inference Eq.(3) to each training-virtual sequence pair:

$$\boldsymbol{T}^{n*} = \underset{\boldsymbol{T}^n \in \boldsymbol{\Phi}}{arg\,min} \langle \boldsymbol{T}^n, \boldsymbol{D}_{\boldsymbol{I}}^n(\boldsymbol{W}) \rangle + \mathscr{R}(\boldsymbol{T}^n). \tag{14}$$

The two updating procedures are repeated until convergence or reaching a maximum number of iterations. We call this framework *Regressive Virtual Sequence Metric Learning (RVSML)* and summarize it in Alg. 1.

Convergence. Both updating procedures of Alg. 1 decrease the value of the objective (10). 0 is a trivial lower bound of the objective (10). Therefore, Alg. 1 ensures the convergence to a local solution.

Instantiation. Alg. 1 can be applied to learn any meta-distance with the form Eq. (1) as discussed in Sec. 3. A specific meta-distance instantiates step.7 in Alg. 1, i.e., the inference of $\boldsymbol{T}^n$. For instance, for DTW, step.7 is performed by dynamic programming; for OPW, step.7 is performed by Sinkhorn's matrix scaling. As long as sufficient inference or optimization method for an instantiation of Eq. (14) is available, Alg. 1 can be efficiently performed.

Non-linear extensions. Alg. 1 can be easily kernelized by kernelizing Eq. (11). It can also be extended to learn non-linear representations by replacing the linear transformation with a non-linear deep network such as RNN in step.3. We provide the kernelized version and the deep version in the supplementary file.

4.3. Links with Other Methods

Connection with RVML (Perrot & Habrard, 2015). RVML can be viewed as a special case of the proposed RVSML. By regarding vector data as sequences with only one element and setting the length of all virtual sequences to 1, the alignment between any training-virtual sequence pair by any meta-distance is unique. Therefore, RVSML degenerates into RVML.

Connection to must-link/cannot-link constraints. Most classical metric learning methods employ pair-based or triplet-based constraints so as to achieve a large margin between similar and dissimilar sample pairs, i.e., the distance between the samples from the same class is below a threshold θ_1, and the distance between those from different classes is above another threshold θ_{-1}.

$$\begin{aligned} &g_{\boldsymbol{M}}(\boldsymbol{X}^n, \boldsymbol{X}^{n'}) \leq \theta_1, for\ z^n = z^{n'} \\ &g_{\boldsymbol{M}}(\boldsymbol{X}^n, \boldsymbol{X}^{n'}) \geq \theta_{-1}, for\ z^n \neq z^{n'} \end{aligned}. \tag{15}$$

When the meta-distance $g_{\boldsymbol{W}}$ is a real metric, in the transformed space induced by RVSML, the distances between similar and dissimilar sequence pairs gain the following margins:

$$\begin{aligned} \theta_1 &= 2 \max_{(\boldsymbol{X}^n, \boldsymbol{V}^n)} g_{\boldsymbol{I}}(\boldsymbol{W}^T \boldsymbol{X}^n, \boldsymbol{V}^n) \\ \theta_{-1} &= \min_{\boldsymbol{V}^n, \boldsymbol{V}^{n'}, \boldsymbol{V}^n \neq \boldsymbol{V}^{n'}} g_{\boldsymbol{I}}(\boldsymbol{V}^n, \boldsymbol{V}^{n'}) - \theta_1 \end{aligned}. \tag{16}$$

Although some well-known meta-distances such as DTW do not satisfy the triangle inequality, intuitively, dissimilar sequences are still pushed relatively far away because they are moved to different distant virtual sequences.

4.4. Virtual Sequences Generation

In this section, we develop an approach to generate the virtual sequences. Another generation approach is presented in the supplementary file. Intuitively, the evolution of a sequence pattern can be segmented into several ordered stages and each stage corresponds to a temporal structure, e.g., an action can be identified by a series of ordered key poses. If the learned $\mathbf{W}$ is able to project the elements corresponding to different temporal structures to different clusters which are far away from each other, different sequence classes would become easier to distinguish.

Following this intuition, we construct a virtual sequence for each class, which consists of vectors w.r.t. the ordered basic temporal structures shared by this class. Let m be the number of temporal structures per class. There are Cm temporal structures for all C classes. We define the vector

Table 1. Comparison of the proposed RVSML variants instantiated by (a) DTW and (b) OPW with other metric learning methods using the NN classifier with the (a) DTW and (b) OPW distance on the MSR Action3D dataset.

(a) DTW

Method	MAP	Accuracy
Ori (Su & Hua, 2018)	58.95	81.32
ITML (Davis et al., 2007)	59.19	80.95
LMNN (Weinberger & Saul, 2009)	54.14	80.95
SCML (Shi et al., 2014)	42.79	63.00
RVML (Perrot & Habrard, 2015)	57.41	80.95
LDMLT (Mei et al., 2014)	**64.29**	**84.98**
SWMD (Huang et al., 2016)	59.65	80.95
RVSML	59.30	82.78

(b) OPW

Method	MAP	Accuracy
Ori (Su & Hua, 2018)	58.70	**84.25**
ITML (Davis et al., 2007)	**59.48**	83.52
LMNN (Weinberger & Saul, 2009)	32.73	82.42
SCML (Shi et al., 2014)	39.63	64.10
RVML (Perrot & Habrard, 2015)	44.58	73.63
LDMLT (Mei et al., 2014)	53.61	80.59
SWMD (Huang et al., 2016)	43.23	66.67
RVSML	47.54	76.56

Table 2. Comparison of the proposed RVSML variants instantiated by (a) DTW and (b) OPW with other metric learning methods using the NN classifier with the (a) DTW and (b) OPW distance on the MSR Activity3D dataset.

(a) DTW

Method	MAP	Accuracy
Ori (Su & Hua, 2018)	33.79	58.75
ITML (Davis et al., 2007)	33.80	58.75
LMNN (Weinberger & Saul, 2009)	32.24	55.63
SCML (Shi et al., 2014)	29.42	45.62
RVML (Perrot & Habrard, 2015)	41.55	60.62
LDMLT (Mei et al., 2014)	36.56	55.00
SWMD (Huang et al., 2016)	37.81	61.25
RVSML	**42.18**	**62.50**

(b) OPW

Method	MAP	Accuracy
Ori (Su & Hua, 2018)	34.62	**58.13**
ITML (Davis et al., 2007)	33.69	**58.13**
LMNN (Weinberger & Saul, 2009)	32.06	**58.13**
SCML (Shi et al., 2014)	28.50	45.00
RVML (Perrot & Habrard, 2015)	**38.73**	56.87
LDMLT (Mei et al., 2014)	34.84	54.37
SWMD (Huang et al., 2016)	35.62	55.00
RVSML	36.64	57.50

for the u-th temporal structure as a unit vector $\boldsymbol{e}_u \in \mathbb{R}^{Cm}$, in which only the u-th attribute is 1 and all other attributes are 0. Therefore, the virtual sequence for the c-th class is $\boldsymbol{V}_c^T = [\mathbf{0}^{m\times m}, \cdots, \mathbf{0}^{m\times m}, \boldsymbol{I}^{m\times m}, \mathbf{0}^{m\times m}, \cdots, \mathbf{0}^{m\times m}] \in \mathbb{R}^{Cm\times m}$, where only the c-th block square matrix is the identity matrix and all other $C-1$ blocks are the null matrices, i.e., $f(\boldsymbol{X}^n, z^n) = \boldsymbol{V}_{z^n} = [\boldsymbol{e}_{(z^n-1)m+1}, \cdots, \boldsymbol{e}_{(z^n-1)m+m}]$. In this way, we generate C virtual sequences each consists of m unit vectors. All unit vectors in all virtual sequences are orthogonal and the active attribute for each vector is attempted to be discriminant for one temporal structure.

5. Experimental Results

5.1. Experimental setup

Datasets. **MSR Action3D dataset (Li et al., 2010)** contains 567 depth video sequences from 20 action classes. We follow the splits in (Wang et al., 2012; Wang & Wu, 2013) to divide the dataset into training and testing sets. **MSR Daily Activity3D dataset (Wang et al., 2012)** consists of 320 Kinect daily activity sequences from 16 activity classes. We follow the splits in (Wang et al., 2012; Wang & Wu, 2013) to divide the dataset into training and test sets. **ChaLearn Gesture dataset (Escalera et al., 2013b;a)** consists of Kinect video sequences from 20 gesture types. The dataset is partitioned into training, validation and test sets. **"Spoken Arabic Digits (SAD)" dataset** from the UCI Machine Learning Repository (Bache & Lichman, 2013) contains 8,800 vector sequences from ten digit classes with 880 sequences per class. The dataset is partitioned into training and test sets. Sequences have different lengths in all datasets, e.g., the length varies from 4 to 93 on the SAD dataset. Results on an additional dataset are presented in the supplementary material.

Sequence representations. For the video sequences, we extract a feature vector from each frame, so as to represent each video as a sequence of frame-wide vectors. Specifically, for the MSR Action3D dataset, we adopt the 192-dimensional relative 3D joint angles based frame-wide vectors as in (Wang & Wu, 2013). For the MSR Activity3D dataset, we employ the 390-dimensional relative 3D joint positions based frame-wide features as in (Wang et al., 2012). For the ChaLearn dataset, we adopt the 100-dimensional 3D joint based frame-wide vectors as in (Fernando et al., 2015). For the SAD dataset, the sequences have already been represented as a series of 13-dimensional mel-frequency cepstrum coefficients features.

Classification and evaluation measures. We evaluate the performances of the proposed RVSML instantiated by the DTW distance and the OPW distance, respectively. After learning the ground metric, we employ the 1-nearest neighbor (NN) classifier in combination with the DTW distance and the OPW distance to perform sequence classification, respectively. The parameters λ_1, λ_2, and σ of OPW are fixed to 10, 0.1 and 12, respectively, on the MSR Activ-

Table 3. Comparison of RVSML instantiated by (a) DTW and (b) OPW with other methods using the NN classifier with the (a) DTW and (b) OPW distance on the ChaLearn dataset.

(a) DTW

Method	MAP	Accuracy
Ori (Su & Hua, 2018)	11.75	61.12
ITML (Davis et al., 2007)	13.46	52.17
LMNN (Weinberger & Saul, 2009)	11.67	63.78
RVML (Perrot & Habrard, 2015)	31.21	83.79
LDMLT (Mei et al., 2014)	21.30	84.37
SWMD (Huang et al., 2016)	14.39	64.45
RVSML	**33.83**	**87.38**

(b) OPW

Method	MAP	Accuracy
Ori (Su & Hua, 2018)	12.21	59.38
ITML (Davis et al., 2007)	13.92	64.71
LMNN (Weinberger & Saul, 2009)	12.07	62.83
RVML (Perrot & Habrard, 2015)	30.19	80.66
LDMLT (Mei et al., 2014)	21.56	82.74
SWMD (Huang et al., 2016)	15.36	60.31
RVSML	**33.07**	**83.82**

ity3D dataset, and 50, 0.1 and 1, respectively, on other datasets, as suggested in (Su & Hua, 2018). We report accuracy as the performance measure. Following (Su & Hua, 2017; 2018), we also regard each test sequence as a query to retrieval all training sequences and report the mean average precision (MAP). For RVSML, we select m in the range of 2 to 8 with an interval of 2 and set β to a small value via cross-validation. The influence of the two hyper-parameters are evaluated in the supplementary material.

5.2. Comparison with metric learning methods

We compare the proposed RVSML with the baseline NN classifier without metric learning (Ori) and several state-of-the-art conventional metric learning methods: ITML (Davis et al., 2007), LMNN (Weinberger & Saul, 2009), SCML (Shi et al., 2014), and RVML (Perrot & Habrard, 2015). These methods are originally developed for vector representations. We apply them to sequences by viewing all elements in the sequence from a class as independent samples of this class. On the ChaLearn dataset, SCML learned 0 LDA base and hence we remove it for comparison. For RVML, we employ the class-based virtual points.

We also compare with two metric learning methods for sequence data, including LDMLT (Mei et al., 2014) and SWMD (Huang et al., 2016). SWMD can not be directly applied to unconstrained sequences because it requires that the elements in sequences are from a finite set and learns the weights for all possible elements in this set. The weights determine the marginal constraints for the transport matrix. We modify SWMD by removing the weight learning procedures and setting the marginal constraints uniformly so that SWMD can be applied to unconstrained sequences. For different metric learning methods, the NN classifiers with DTW and OPW distances are used for classification by taking the learned metrics as ground metrics, respectively. RVSML is instantiated by the distances used by the corresponding NN classifiers, respectively.

Table 4. Comparison of RVSML instantiated by (a) DTW and (b) OPW with other metric learning methods using the NN classifier with the (a) DTW and (b) OPW distance on the SAD dataset.

(a) DTW

Method	MAP	Accuracy
Ori (Su & Hua, 2018)	56.58	96.36
ITML (Davis et al., 2007)	51.13	95.55
LMNN (Weinberger & Saul, 2009)	56.25	96.00
SCML (Shi et al., 2014)	47.98	93.27
RVML (Perrot & Habrard, 2015)	57.94	**96.59**
LDMLT (Mei et al., 2014)	59.54	96.50
SWMD (Huang et al., 2016)	52.44	93.95
RVSML	**60.24**	96.23

(b) OPW

Method	MAP	Accuracy
Ori (Su & Hua, 2018)	59.77	96.36
ITML (Davis et al., 2007)	54.51	96.36
LMNN (Weinberger & Saul, 2009)	59.33	96.27
SCML (Shi et al., 2014)	50.08	94.50
RVML (Perrot & Habrard, 2015)	60.71	95.77
LDMLT (Mei et al., 2014)	61.07	96.73
SWMD (Huang et al., 2016)	58.00	95.41
RVSML	**65.63**	**97.09**

Table 5. Comparison of the training times.

Dataset	Action3D	SAD	ChaLearn
LDMLT	1905.24	67329.29	213921.7
SWMD	970.70	7756.72	11489.10
RVSML(DTW)	115.72	662.41	2477.76
RVSML(OPW)	124.48	208.67	836.67

The comparisons on the four datasets are presented in Tab. 1, Tab. 2, Tab. 3, and Tab. 4, respectively. On the ChaLearn and SAD datasets, RVSMLs instantiated by both distances generally outperform the corresponding baseline classifiers without metric learning and other metric learning methods, respectively. Especially, on the ChaLearn dataset, RVSMLs outperform other methods by a margin of 3% on accuracies. RVSML is able to learn a discriminative ground metric that incorporates the holistic temporal dependencies of sequences and enhance different meta-distances consistently. In some cases, several conventional metric learning methods obtain worse results than the baseline classifiers. This may indicate that temporal information is inherent for sequence data and cannot be discarded.

On the Action3D dataset with the DTW distance, RVSML-

s perform inferior to LDMLT, but generally outperforms other metric learning methods. LDMLT is based on the dynamic triplet constraints, cannot ensure the convergence, and requires much more time for training. The training times of different metric learning methods for sequences on three datasets are shown in Tab. 5. We can observe that RVSML trains much faster compared with these sequence distance learning methods. Specifically, the training time of LDMLT is more than ten times the training time of RVSM-L instantiated by DTW on most datasets, the training of SWMD is also at least 5 times slower than RVSML.

The performances of the proposed RVSML depend on the choice of virtual sequences. Due to the different properties and real distributions of the sequence data, different virtual sequences can lead to different effects on the learned metric. In the supplementary file, we show that with a different virtual sequence generation method, RVSMLs instantiated by both distances achieves better results on this dataset.

5.3. Combination with state-of-the-art methods

The proposed RVSML learns a transformation that projects the sequences into another space. In the resulting space, we can use other advanced classification methods instead of the NN classifier. That is, we first apply the proposed RVSML to the original sequences and then employ state-of-the-art classification methods by taking the transformed sequences as input. In this way, the proposed RVSML can be combined with these methods.

We combine RVSML with kernelized-COV (Cavazza et al., 2016), which extracts the kernelized covariance representation from each sequence and applies SVM for classification. In (Cavazza et al., 2016), the 120-dimensional velocity and acceleration of the raw joint positions based frame-wide features (Zanfir et al., 2013) were employed. On the M-SR Activity3D dataset, the pre-computed features are provided and hence we directly apply RVSML to them. On the MSR Action3D dataset, we compute the features following (Zanfir et al., 2013), where the velocity and acceleration features are augmented by the raw joint positions. We perform Kernelized-COV to the transformed sequences. Tab. 6 and Tab. 7 show the results in comparison with the state-of-the-art methods on the two datasets, respectively. The combination of our method and Kernelized-COV achieves comparable results with other competitors.

On the MSR Action3D dataset, we also combine RVSML with the generalized temporal sliding LSTM (TS-LSTM) Network (Lee et al., 2017) denoted by TS-LSTM-GM. We apply RVSML to the 60-dimensional motion features used in (Lee et al., 2017), perform L_2 normalization to the transformed features, and input the resulting sequences to TS-LSTM-GM. The results are shown in Tab. 7. The proposed RVSML instantiated by DTW improves the accuracy of TS-LSTM-GM by 1.8%. RVSML instantiated by different meta-distances fits for different classification methods.

Table 6. Comparison with state-of-the-art methods on the MSR Activity3D dataset.

Method	Accuracy
Actionlet Ensemble (Wang et al., 2012)	85.8%
Moving Pose (Zanfir et al., 2013)	73.8%
COV-$J_{\mathcal{H}}$-SVM (Harandi et al., 2014)	75.5%
Ker-RP-POL (Wang et al., 2015)	96.9%
Ker-RP-RBF (Wang et al., 2015)	96.3%
Kernelized-COV (Cavazza et al., 2016)	96.3%
Luo et al. (Luo et al., 2017)	86.9%
Ji et al. (Ji et al., 2018)	81.3%
DSSCA SSLM (Shahroudy et al., 2018)	**97.5%**
RVSML-DTW+Kernelized-COV	96.9%
RVSML-OPW+Kernelized-COV	**97.5%**

Table 7. Comparison with state-of-the-art methods on the MSR Action3D dataset.

Method	Accuracy
Actionlet Ensemble (Wang et al., 2012)	88.2%
Moving Pose (Zanfir et al., 2013)	91.7%
COV-$J_{\mathcal{H}}$-SVM (Harandi et al., 2014)	80.4%
Ker-RP-POL (Wang et al., 2015)	96.2%
Ker-RP-RBF (Wang et al., 2015)	**96.9%**
Kernelized-COV (Cavazza et al., 2016)	96.2%
SCK+DCK (Koniusz et al., 2016)	91.45%
TS-LSTM-GM (Lee et al., 2017)	91.21%
FTP-SVM (Ben Tanfous et al., 2018)	90.01%
Bi-LSTM (Ben Tanfous et al., 2018)	86.18%
RVSML-DTW+Kernelized-COV	82.78%
RVSML-OPW+Kernelized-COV	**96.34%**
RVSML-DTW+TS-LSTM-GM	93.04%
RVSML-OPW+TS-LSTM-GM	90.48%

6. Conclusion

We present a metric learning framework for sequence data, which learns the meta-distance for sequences via learning the ground metric. The objective is to minimize the meta-distances between training sequences and their associated a prior defined virtual sequences. Constructing the meta-distance needs to infer the temporal alignments, but the inference also depends on the ground metric. We propose an efficient iterative solution to learn the ground metric and the latent alignments jointly. We unify a family of meta-distance measures for sequences into a common formulation and show that any meta-distance with such form can be employed to instantiate our framework. Additionally, we propose an approach to generate discriminative virtual sequences. We empirically show that our method is able to enhance different types of meta-distances and state-of-the-art sequence classification methods.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No.61603373, Youth Innovation Promotion Association CAS No. 2019110, National Science Foundation grant IIS-1619078, IIS-1815561, and the Army Research Office ARO W911NF-16-1-0138.

References

Bache, K. and Lichman, M. *UCI Machine Learning Repository*. http://archive.ics.uci.edu/ml, University of California, Irvine, School of Information and Computer Sciences, 2013.

Bayer, J., Osendorfer, C., and Smagt, P. V. D. Learning sequence neighbourhood metrics. In *Proceedings of the 22Nd International Conference on Artificial Neural Networks and Machine Learning*, pp. 2638–2644, 2012.

Bellet, A., Habrard, A., and Sebban, M. Learning good edit similarities with generalization guarantees. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 188–203. Springer, 2011.

Bellet, A., Habrard, A., and Sebban, M. Good edit similarity learning by loss minimization. *Machine Learning*, 89 (1-2):5–35, 2012.

Bellet, A., Habrard, A., and Sebban, M. A survey on metric learning for feature vectors and structured data. *arXiv preprint arXiv:1306.6709*, 2013.

Ben Tanfous, A., Drira, H., and Ben Amor, B. Coding kendall's shape trajectories for 3d action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2840–2849, 2018.

Cavazza, J., Zunino, A., San Biagio, M., and Murino, V. Kernelized covariance for action recognition. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pp. 408–413. IEEE, 2016.

Che, Z., He, X., Xu, K., and Liu, Y. Decade: A deep metric learning model for multivariate time series. In *3rd SIGKDD Workshop on mining and learning from time series*, 2017.

Cortes, C., Haffner, P., and Mohri, M. Rational kernels: Theory and algorithms. *Journal of Machine Learning Research*, 5(Aug):1035–1062, 2004.

Cortes, C., Mohri, M., and Rostamizadeh, A. Learning sequence kernels. In *2008 IEEE Workshop on Machine Learning for Signal Processing*, pp. 2–8. IEEE, 2008.

Cuturi, M. and Avis, D. Ground metric learning. *The Journal of Machine Learning Research*, 15(1):533–564, 2014.

Davis, J. V., Kulis, B., Jain, P., Sra, S., and Dhillon, I. S. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, pp. 209–216. ACM, 2007.

Escalera, S., Gonzàlez, J., Baró, X., Reyes, M., Guyon, I., Athitsos, V., Escalante, H., Sigal, L., Argyros, A., Sminchisescu, C., et al. Chalearn multi-modal gesture recognition 2013: grand challenge and workshop summary. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pp. 365–368. ACM, 2013a.

Escalera, S., Gonzàlez, J., Baró, X., Reyes, M., Lopes, O., Guyon, I., Athitsos, V., and Escalante, H. Multi-modal gesture recognition challenge 2013: Dataset and results. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pp. 445–452. ACM, 2013b.

Fernando, B., Gavves, E., M., J. O., Ghodrati, A., and Tuytelaars, T. Modeling video evolution for action recognition. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2015.

Garreau, D., Lajugie, R., Arlot, S., and Bach, F. Metric learning for temporal sequence alignment. In *Advances in neural information processing systems*, pp. 1817–1825, 2014.

Globerson, A. and Roweis, S. T. Metric learning by collapsing classes. In *Advances in neural information processing systems*, pp. 451–458, 2006.

Goldberger, J., Hinton, G. E., Roweis, S. T., and Salakhutdinov, R. R. Neighbourhood components analysis. In *Advances in neural information processing systems*, pp. 513–520, 2005.

Harandi, M., Salzmann, M., and Porikli, F. Bregman divergences for infinite dimensional covariance matrices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1003–1010, 2014.

Huang, G., Guo, C., Kusner, M. J., Sun, Y., Sha, F., and Weinberger, K. Q. Supervised word mover's distance. In *Advances in Neural Information Processing Systems*, pp. 4862–4870, 2016.

Ji, X., Cheng, J., Feng, W., and Tao, D. Skeleton embedded motion body partition for human action recognition using depth sequences. *Signal Processing*, 143:56–68, 2018.

Jin, R., Wang, S., and Zhou, Y. Regularized distance metric learning: Theory and algorithm. In *Advances in neural information processing systems*, pp. 862–870, 2009.

Koniusz, P., Cherian, A., and Porikli, F. Tensor representations via kernel linearization for action recognition from 3d skeletons. In *European Conference on Computer Vision*, pp. 37–53. Springer, 2016.

Lee, I., Kim, D., Kang, S., and Lee, S. Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 1012–1020. IEEE, 2017.

Li, W., Zhang, Z., and Liu, Z. Action recognition based on a bag of 3d points. In *IEEE Int'l Workshop on CVPR for Human Communicative Behavior Analysis*, 2010.

Luo, Z., Peng, B., Huang, D.-A., Alahi, A., and Fei-Fei, L. Unsupervised learning of long-term motion dynamics for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2203–2212, 2017.

Mahalanobis, P. C. On the generalized distance in statistics. National Institute of Science of India, 1936.

Mei, J., Liu, M., Karimi, H. R., and Gao, H. Logdet divergence-based metric learning with triplet constraints and its applications. *IEEE Transactions on Image Processing*, 23(11):4920–4931, 2014.

Mei, J., Liu, M., Wang, Y.-F., and Gao, H. Learning a mahalanobis distance-based dynamic time warping measure for multivariate time series classification. *IEEE transactions on Cybernetics*, 46(6):1363–1374, 2016.

Mokbela, B., Paassen, B., Schleif, F. M., and Hammer, B. Metric learning for sequences in relational lvq. *Neurocomputing*, 169:306–322, 2015.

Paaßen, B., Gallicchio, C., Micheli, A., and Hammer, B. Tree edit distance learning via adaptive symbol embeddings. In *International Conference on Machine Learning*, 2018.

Perrot, M. and Habrard, A. Regressive virtual metric learning. In *Advances in Neural Information Processing Systems*, pp. 1810–1818, 2015.

Qi, G.-J., Tang, J., Zha, Z.-J., Chua, T.-S., and Zhang, H.-J. An efficient sparse metric learning in high-dimensional space via l 1-penalized log-determinant regularization. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 841–848. ACM, 2009.

Rubner, Y., Tomasi, C., and Guibas, L. J. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

Sakoe, H. and Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustics, Speech and Signal Processing*, 26(1):43–49, 1978.

Schultz, M. and Joachims, T. Learning a distance metric from relative comparisons. In *Advances in neural information processing systems*, pp. 41–48, 2004.

Shahroudy, A., Ng, T.-T., Gong, Y., and Wang, G. Deep multimodal feature analysis for action recognition in rgb+ d videos. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1045–1058, 2018.

Shi, Y., Bellet, A., and Sha, F. Sparse compositional metric learning. In *AAAI*, pp. 2078–2084, 2014.

Song, H. O., Xiang, Y., Jegelka, S., and Savarese, S. Deep metric learning via lifted structured feature embedding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4004–4012, 2016.

Su, B. and Hua, G. Order-preserving wasserstein distance for sequence matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1049–1057, 2017.

Su, B. and Hua, G. Order-preserving optimal transport for distances between sequences. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Wang, J. and Wu, Y. Learning maximum margin temporal warping for action recognition. In *Proc. IEEE Int'l Conf. Computer Vision*, 2013.

Wang, J., Liu, Z., and Wu, Y. Mining actionlet ensemble for action recognition with depth cameras. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2012.

Wang, L., Zhang, J., Zhou, L., Tang, C., and Li, W. Beyond covariance: Feature representation with nonlinear kernel matrices. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4570–4578, 2015.

Weinberger, K. Q. and Saul, L. K. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(Feb):207–244, 2009.

Weinberger, K. Q., Blitzer, J., and Saul, L. K. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pp. 1473–1480, 2006.

Xing, E. P., Jordan, M. I., Russell, S. J., and Ng, A. Y. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pp. 521–528, 2003.

Yi, D., Lei, Z., Liao, S., and Li, S. Z. Deep metric learning for person re-identification. In *International Conference on Pattern Recognition*, pp. 34–39, 2014.

Zanfir, M., Leordeanu, M., and Sminchisescu, C. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2752–2759, 2013.

Zhao, J., Xi, Z., and Itti, L. metricdtw: local distance metric learning in dynamic time warping. *arXiv preprint arXiv:1606.03628*, 2016.