Auto-weighted Sequential Wasserstein Distance and Application to Sequence Matching

Mitsuhiko Horie
Dept. of Computer Science and Comm. Engineering
Waseda University
Tokyo, Japan
kidsgoldppp@akane.waseda.jp

Hiroyuki Kasai
Dept. of Computer Science and Comm. Engineering
Waseda University
Tokyo, Japan
hiroyuki.kasai@waseda.jp

***Abstract*—Sequence matching problems have been central to the field of data analysis for decades. Such problems arise in widely diverse areas including computer vision, speech processing, bioinformatics, and natural language processing. However, solving such problems efficiently is difficult because one must consider temporal consistency, neighborhood structure similarity, robustness to noise and outliers, and flexibility on start-end matching points. This paper presents a proposal of a shape-aware Wasserstein distance between sequences building upon optimal transport (OT) framework. The proposed distance considers similarity measures of the elements, their neighborhood structures, and temporal positions. We incorporate these similarity measures into three ground cost matrixes of the OT formulation. The noteworthy contribution is that we formulate these measures as independent OT distances with a single shared optimal transport matrix, and adjust those weights automatically according to their effects on the total OT distance. Numerical evaluations suggest that the sequence matching method using our proposed Wasserstein distance robustly outperforms state-of-the-art methods across different real-world datasets.**

***Index Terms*—optimal transport, sequence matching, dynamic time warping**

I. Introduction

Sequence data naturally appear in a wide variety of data domains including video, audio, sensor data, financial data, and event sequence data, where the order of elements included in data has important meaning [1]. Therefore, measuring distances between sequence data plays a crucially important role in many applications such as video classification, audio analysis, signal processing, and item recommendation. Nevertheless, the judgment of similarity of two sequence data must include consideration of many aspects including similarities of the elements, their orders, and the relations between adjacent elements. Therefore, efficient distance measurement between sequence data has attracted a surge of research interest. One approach to measure distances between sequences is *sequence matching*. Sequence matching finds correspondence among all the elements of two sequences, and defines the distance between the entire two sequences according to the total distance of the corresponding elements. At the matching process, the sequence data features must be considered in a comprehensive manner. Particularly, even if the same data are included in the two matching sequences, they are not necessarily matched when they are temporally distant or when their neighborhood structures are completely different.

One representative method of such sequence matching methods is Dynamic Time Warping (DTW) [2], which uses dynamic programming to find the optimal matching which minimizes the total distance between matched elements under some constraints. DTW can align sequences for which the lengths or frequencies are different [3]–[5]. In fact, DTW has been extended considerably in many ways to achieve, for example, higher robustness of noises and outliers, and higher flexibility to signal characteristics [3]–[7]. In another recent avenue of matching methods, several works leverage the optimal transport (OT) framework [8]. One salient benefit of such OT-based matching methods is the capability of accommodating *local inversion of the orders of elements* and the differences of starting point of periodic sequences. However, existing OT-based matching methods do not consider the neighborhood structures of each element. Therefore, matchings which are not intuitive, such as matching of elements on the rise and on the fall, often happen. More importantly, they provide no framework to accommodate multiple features of interests within the OT framework in a unified manner.

To overcome these issues, we propose a novel Wasserstein distance between sequence data that explicitly considers the similarity of relations between elements and their adjacent elements. More concretely, we define costs for three features, which are the similarity of elements, the similarity of neighborhood structure of each element, and the similarity of temporal positions. Then, we minimize the total transport loss. It is worth mentioning that we give weights for each cost and automatically calculate the optimal values for each pair of sequences by applying a self-weighting process. We designate our proposed distance as Auto-weighted Sequential Wasserstein Distance, called AWSWD.

In this paper, scalars, vectors, and matrices are represented by lower-case letters $(a, b, \ldots)$, bold lower-case letters $(\boldsymbol{a}, \boldsymbol{b}, \ldots)$, and bold-typeface capitals $(\mathbf{A}, \mathbf{B}, \ldots)$. The i-th elements of $\boldsymbol{a}$ are represented as a_i. The (i, j) position of $\mathbf{A}$ is represented as a_{ij} or $\mathbf{A}(i, j)$. $\mathbf{1}_d$ is used for the d-dimensional vector of ones. $\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{i,j} a_{ij} b_{ij}$ is the Frobenius dot product of matrices $\mathbf{A}$ and $\mathbf{B}$. When vector $\boldsymbol{a}$ lies in the probability simplex with d bins Δ_d, $\Delta_d := \{\boldsymbol{a} \in$

ISBN: 978-1-6654-6798-8

$\mathbb{R}^d \mid a_i \geq 0, \ \forall i = 1, \cdots d, \ \sum_{i=1}^n a_i = 1\}$. Operator ./ represents the element-wise division of vectors or matrices. Also, $\mathrm{diag}(\boldsymbol{a})$ represents a diagonal matrix of which diagonal and off-diagonal elements are $\boldsymbol{a}$ and zeros, respectively.

II. Related Work

A. Sequence Matching Methods

The most widely used way for solving sequence matching is DTW [2]. Many efforts have been undertaken to enhance the power of DTW. Earlier studies [9]–[11] have addressed reduction of computational costs. The Sakoe-Chiba band [2] and Itakura Parallelogram [12] restrict the warping area to prevent non-intuitive matching patterns called singularity, where one element of one sequence matches multiple elements of the other sequence. The authors of [6] also aim at reducing the singularity by assigning extra cost on temporally distant elements. Derivative DTW [3] exploits derivatives of elements; shapeDTW [4] and LSDTW [5] use feature vectors that capture the shape of sequence around each element instead of raw data. By conducting DTW with derivatives or feature vectors, they consider the relations between adjacent elements. As for OT-based approaches, the Order-Preserving Wasserstein distance (OPW) regards sequences and those elements as a *probability distribution*, and calculates the matching as the transport matrix of an OT problem [8]. To handle the order-structure of sequence, OPW considers two additional regularizers that are transformed into the entropic regularization. Therefore, this problem can be solved efficiently using the Sinkhorn algorithm [13]. In another line of methods, instead of calculating the element-wise matching, the authors of [14] generate groups of elements which represents the same events, and conducts group-wise matching. The group-generating process absorbs the difference of speed in each sequence. Therefore, this method can also be regarded as considering relations between adjacent elements.

B. Optimal Transport

Kantorovich relaxation formulation of the OT problem [15], [16] is explained briefly in this section. Let $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ respectively represent the probability or positive weight vectors as $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)^T \in \mathbb{R}_+^m$ and $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_n)^T \in \mathbb{R}_+^n$. Given two empirical distributions, i.e., discrete measures, $\boldsymbol{\nu} = \sum_{i=1}^m \alpha_i \boldsymbol{\delta}_{x_i}$, $\boldsymbol{\mu} = \sum_{j=1}^n \beta_j \boldsymbol{\delta}_{y_j}$ and the ground cost matrix $\mathbf{C} \in \mathbb{R}^{m \times n}$ between their supports, the problem minimizes the total transportation cost as

$$\min_{\mathbf{T} \in \mathbf{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})} \quad \langle \mathbf{T}, \mathbf{C} \rangle, \tag{1}$$

where $\mathbf{T} \in \mathbb{R}^{m \times n}$ is called the *transport matrix.* $\mathbf{U}(\boldsymbol{\alpha}, \boldsymbol{\beta}) := \{\mathbf{T} \in \mathbb{R}_+^{m \times n} | \mathbf{T}\mathbf{1}_n = \boldsymbol{\alpha}, \mathbf{T}^T \mathbf{1}_m = \boldsymbol{\beta}\}$ is the constraint for mass conservation in transportation. The obtained optimal transport matrix $\mathbf{T}^*$ brings powerful distances as $\mathcal{W}_p(\boldsymbol{\nu}, \boldsymbol{\mu}) = \langle \mathbf{T}^*, \mathbf{C} \rangle^{\frac{1}{p}}$, which is called the p-th order *OT distance* or *Wasserstein distance* [16]. This distance is used in many machine learning applications such as domain adaptation [17], clustering [18], barycenter problem [19], discriminant learning [20], color transfer [21], style transfer [22], and graph classification problem [23], [24]. The Sinkhorn algorithm solves the OT problem with an entropy regularizer $h(\mathbf{T}) = -\sum_{i,j} t_{i,j} \log t_{i,j}$ [13], [25], [26]. The entropy regularized OT problem and the Sinkhorn algorithm have some advantages such as having a unique optimal solution, faster computation, and differentiability of the objective function.

III. Proposed Auto-weighted Sequential Wasserstein distance: AWSWD

This section presents a proposal of a shape-aware Wasserstein distance between sequence data. The proposed distance is calculated with consideration of three features of sequence data, which are the similarity of elements, the similarity of neighborhood structure of each element, and the similarity of temporal positions. We incorporate these similarity measures into the three ground cost matrixes of the OT formulation. It is noteworthy that we formulate these measures as *independent* OT distances with one *shared* OT matrix, and adjust their weights *automatically* according to their distances.

A. OT-based Problem Formulation

Given two d-dimensional sequence data $\mathbf{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_m) \in \mathbb{R}^{d \times m}$, $\mathbf{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_n) \in \mathbb{R}^{d \times n}$ with length m and n, we define three ground cost matrices $\mathbf{D}_i \in \mathbb{R}^{m \times n}$ $(i = 1, 2, 3)$ of size $m \times n$.

We first denote the value similarity of elements as $\mathbf{D}_1$, which is the Euclidian distance matrix between $\mathbf{X}$ and $\mathbf{Y}$ as

$$\mathbf{D}_1(i, j) = \|\boldsymbol{x}_i - \boldsymbol{y}_j\|_2^2. \tag{2}$$

With respect to the similarity measures of the neighborhood structure of each element, denoted as $\mathbf{D}_2$, we address the derivative of each element as representing the relations between adjacent elements. We then define $\boldsymbol{x}'$ as

$$\boldsymbol{x}_i' = \frac{1}{2}\left((\boldsymbol{x}_i - \boldsymbol{x}_{i-1}) + \left(\frac{\boldsymbol{x}_{i+1} - \boldsymbol{x}_{i-1}}{2}\right)\right).$$

The first term is the differential of $\boldsymbol{x}_i$ and its previous element $\boldsymbol{x}_{i-1}$, and the second term is the differential of the two elements next to $\boldsymbol{x}_i$. The derivative value of $\boldsymbol{x}_i$ is defined as the average value of the two differentials. This assumption is the same as the one proposed in [3], [5]. Using this value, $\mathbf{D}_2(i, j)$ is defined as the sum of the distance of derivatives of adjacent l elements centered by $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$, which is given as

$$\mathbf{D}_2(i, j) = \sum_{k=-l}^{l} \|\boldsymbol{x}_{i+k}' - \boldsymbol{y}_{j+k}'\|_2^2. \tag{3}$$

In the experiment, we set $\|\boldsymbol{x}_{i+k}' - \boldsymbol{y}_{j+k}'\|_2^2 = 0$ if the indices $i + k$ or $j + k$ is outside of the length of the sequences

Finally, we define the cost matrix $\mathbf{D}_3(i, j)$, which stands for the similarity of sequential positions of $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$, as the logistic function of the order distance $|i - j|$, i.e.,

$$\mathbf{D}_3(i, j) = \frac{1}{1 + \exp(-k(|i - j| - t_0))}, \tag{4}$$

where k and t_0 respectively denote the sigmoid curve steepness and midpoint.

Now, putting all of the distances of $\mathbf{D}_i$ $(i = 1, 2, 3)$ and assigning weights $\boldsymbol{w} = (w_1, w_2, w_3)^T \in \mathbb{R}^3$ for each OT distance, we define the total Wasserstein distance between $\mathbf{X}$ and $\mathbf{Y}$ as

$$\mathrm{dist}(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{3} w_i \langle \mathbf{T}, \mathbf{D}_i \rangle.$$

We then obtain the following OT problem with entropic regularization of the optimal matching as

$$\mathbf{T}^* = \underset{\mathbf{T} \in \mathbf{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})}{\arg\min} \left\{ f(\mathbf{T}) := \sum_{i=1}^{3} w_i \langle \mathbf{T}, \mathbf{D}_i \rangle - \frac{1}{\lambda} \cdot h(\mathbf{T}) \right\}, \tag{5}$$

where $\lambda > 0$ is the regularization parameter. Finally, we obtain $\sum_{i=1}^{3} w_i \langle \mathbf{T}^*, \mathbf{D}_i \rangle$ as our proposed AWSWD.

B. Optimization Algorithm with Auto Calculation of Weights

The OT problem in (1) is a linear programming problem, i.e., a convex problem. It is true, however, that the problem in (5) is non-convex. Therefore, we consider a bi-convex alternative approach. For this purpose, we particularly address calculation of the weight vector $\boldsymbol{w}$ by borrowing the technique introduced in multi-view clustering [27]. More specifically, we introduce the auxiliary function below.

$$\min_{\mathbf{T} \in \mathbf{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})} \sum_{i=1}^{3} \sqrt{\langle \mathbf{T}, \mathbf{D}_i \rangle} - h(\mathbf{T}). \tag{6}$$

Setting the derivative of the this function of (6) with respect to $\mathbf{T}$ to zero, we obtain

$$\hat{w}_i \frac{\partial \langle \mathbf{T}, \mathbf{D}_i \rangle}{\partial \mathbf{T}} + \frac{\partial \Theta(\lambda_1, \lambda_2, \mathbf{T})}{\partial \mathbf{T}} = 0, \tag{7}$$

where $\Theta(\lambda_1, \lambda_2, \mathbf{T}) = h(\mathbf{T}) - \lambda_1(\mathbf{T}\mathbf{1}_n - \boldsymbol{\alpha}) - \lambda_2(\mathbf{T}^T\mathbf{1}_m - \boldsymbol{\beta})$ represents the terms derived from the entropy term and the constraints. λ_1 and λ_2 are the Lagrangian multipliers of the constraints. Consequently, this finally produces

$$\hat{w}_i = \frac{1}{2\sqrt{\langle \mathbf{T}, \mathbf{D}_i \rangle}}. \tag{8}$$

It is noteworthy that (7) is difficult to solve because $\hat{w}_i$ is dependent on $\mathbf{T}$. However, assuming that $\hat{w}_i$ is fixed, (7) is equal to the derivative of the function of (5). Subsequently, we can take the alternative optimization strategy of $\mathbf{T}$ by (5), and w_i by (8). This converges to a locally minimum solution $\mathbf{T}$ of (6). We summarize the optimization algorithm to calculate our proposed AWSWD in **Algorithm 1**.

Here, considering the relationships between w_i and $\mathbf{D}_i$, the weight w_i takes a large value when the loss $\langle \mathbf{T}, \mathbf{D}_i \rangle$ is small. Thus, the cost matrix $\mathbf{D}_i$ with a small loss can be regarded as important information. The weight calculation by (8) assigned a greater weight for the corresponding i. Therefore this weight calculation can be regarded as a process of learning optimal weights. These behaviors are confirmed in Section IV-A.

C. Convergence Analysis

This subsection presents a simple convergence analysis of **Algorithm 1**, which is inspired by [27]. For this purpose, we first give the following lemma.

Lemma III.1 (**Lemma 1** in [27]). *For any positive number a and b, the following inequality holds: $a - \frac{a^2}{2b} \leq b - \frac{b^2}{2b}$.*

Denoting $\mathbf{T}$ and w_i after the k-th iteration respectively as $\mathbf{T}^k$ and w_i^k, we now give a convergence analysis.

Theorem III.2 (Convergence of **Algorithm 1**). *Let $\{\mathbf{T}^k\}_{k \geq 0}$ be the transport matrix generated by* **Algorithm 1** *for solving the problem (5). Assume that $f(\mathbf{T})$ is bounded below over $\mathbf{U}(\boldsymbol{\alpha}, \boldsymbol{\beta})$. Then, the functional sequence $\{f(\mathbf{T}^k)\}_{k \geq 0}$ is non-increasing. Therefore, it converges.*

Proof. With the update of $\mathbf{T}^{k+1}$ from $\mathbf{T}^k$ in (5) under the fixed $w_i^k = 1/2\sqrt{\langle \mathbf{T}^k, \mathbf{D}_i \rangle}$, the following inequality holds:

$$\begin{aligned} &\sum_{i=1}^{3} \frac{\langle \mathbf{T}^{k+1}, \mathbf{D}_i \rangle}{2\sqrt{\langle \mathbf{T}^k, \mathbf{D}_i \rangle}} - \lambda h(\mathbf{T}^{k+1}) \\ &\leq \sum_{i=1}^{3} w_i \frac{\langle \mathbf{T}^k, \mathbf{D}_i \rangle}{2\sqrt{\langle \mathbf{T}^k, \mathbf{D}_i \rangle}} - \lambda h(\mathbf{T}^k). \end{aligned} \tag{9}$$

Then, the following is derived from **Lemma III.1**:

$$\begin{aligned} &\sum_{i=1}^{3} \sqrt{\langle \mathbf{T}^{k+1}, \mathbf{D}_i \rangle} - \sum_{i=1}^{3} \frac{\langle \mathbf{T}^{k+1}, \mathbf{D}_i \rangle}{2\sqrt{\langle \mathbf{T}^k, \mathbf{D}_i \rangle}} \\ &\leq \sum_{i=1}^{3} \sqrt{\langle \mathbf{T}^k, \mathbf{D}_i \rangle} - \sum_{i=1}^{3} \frac{\langle \mathbf{T}^k, \mathbf{D}_i \rangle}{2\sqrt{\langle \mathbf{T}^k, \mathbf{D}_i \rangle}}. \end{aligned} \tag{10}$$

Adding (9) and (10) yields

$$\sum_{i=1}^{3} \sqrt{\langle \mathbf{T}^{k+1}, \mathbf{D}_i \rangle} - h(\mathbf{T}^{k+1}) \leq \sum_{i=1}^{3} \sqrt{\langle \mathbf{T}^k, \mathbf{D}_i \rangle} - h(\mathbf{T}^k).$$

Therefore, the updated $\mathbf{T}$ decreases the value of (6), and **Algorithm 1** can obtain a local optimal solution. □

IV. Numerical evaluations

This section presents numerical evaluations of the proposed AWSWD using several real-world datasets. The datasets are Swedish-Leaf, FaceAll, ItalyPowerDemand, and SonyAIBORobotSurface1 from UCR time series datasets[1], which are one-dimensional data. We also use multi-dimensional datasets, such as Spoken Arabic Digit (SAD) dataset[2], High Quality Australian Sign Language Signs (HAS) dataset[2] [28], and MSR Sports Action3D dataset [29]. The experimental setup for MSR Action3D dataset is based on [30]. They are summarized in TABLE I. As for HAS and SAD, the length of sequence is not fixed, so the average length is shown in TABLE I. We compare the sequence matching method using our proposed distance AWSWD with state-of-the-art sequence matching methods which include DTW [2], Soft-DTW [7], DDTW [3], shapeDTW [4], and OPW [8].

[1] https://www.cs.ucr.edu/~eamonn/time_series_data_2018/

[2] https://archive.ics.uci.edu/ml/datasets.php

Algorithm 1 AWSWD Algorithm

```
**Require:** Data $\mathbf{X}$, $\mathbf{Y}$, parameter $\lambda$, subsequence length $l$
**Ensure:** Matching matrix $\mathbf{T}$, and distance between $\mathbf{X}$, $\mathbf{Y}$
  Calculate ground cost matrices $\mathbf{D}_i$ by (2), (3), (4), and initialize weights $w_i = \frac{1}{3}$ $(i = 1, 2, 3)$
  while $w_i$ has not converged do
    Calculate $\tilde{\mathbf{D}} = w_1\mathbf{D}_1 + w_2\mathbf{D}_2 + w_3\mathbf{D}_3$
    Calculate $\mathbf{K} = e^{-\tilde{\mathbf{D}}/\lambda}$
    Initialize $\boldsymbol{k}_1 = \mathbf{1}_m/m$
    while $\boldsymbol{k}_1$ has not converged do
      Update $\boldsymbol{k}_2 = \boldsymbol{\beta}./\mathbf{K}^{\mathsf{T}}\boldsymbol{k}_1$
      Update $\boldsymbol{k}_1 = \boldsymbol{\alpha}./\mathbf{K}\boldsymbol{k}_2$
    end while
    Calculate transport matrix $\mathbf{T} = \mathrm{diag}(\boldsymbol{k}_1)\mathbf{K}\mathrm{diag}(\boldsymbol{k}_2)$
    Update $w_i$ with (8) for $i = 1, 2, 3$
  end while
  Calculate distance $\sum_{i=1}^{3} w_i \langle \mathbf{T}^*, \mathbf{D}_i \rangle$
```

TABLE I
SPECIFICATION OF DATASETS.

dataset	dim	length	# class	# train	# test
SwedishLeaf	1	128	15	500	625
FaceAll	1	131	14	560	1690
ItalyPowerDemand	1	24	2	67	1029
SonyAIBORobotSurface1	1	70	2	20	601
HAS	22	57.3	95	1235	1330
SAD	13	39.8	10	6600	2200
MSRAction3D	192	24	20	284	273

The parameters for $\mathbf{D}_3$ are set to $k = 0.1$ and t_0 to one-third of the input sequence length. Our preliminary experiments revealed that smaller values of k, i.e., the linearly distributed $\mathbf{D}_3$ yield better classification results than larger values of k do, i.e., binary distribution with the boundary t_0.

A. Behaviors of Auto Calculation of Weights

We first evaluate the behaviors and impacts of the weight value $\boldsymbol{w}$ by using synthetic dataset. Concretely, we randomly chose one datum from SwedishLeaf dataset as the original sequence. We denote it as Type A. Then, we prepare two artificially modified sequences. We add a small positive offset along the y-axis direction, and denote it as Type B. The second sequence, denoted Type C, contains a small positive offset along the x-axis direction. These sequences are shown in Fig. 1. Now, for these three sequences, we calculate the proposed AWSWD, and see how the obtained weight value $\boldsymbol{w}$ changes for each type. The obtained weight value $\boldsymbol{w}$ is summarized in TABLE II, where the ratio of w_i to the total $\|\boldsymbol{w}\|_1$ is also summarized. Note that the parenthesized values in TABLE II indicate increase-decrease rates compared to the values in Type A. From this table, it can be seen that, when the y-axis offset is added as in Type B, the ratio of w_1 gets smaller than that of Type A. This behavior of w_1 can be regarded as the effect to reduce the additional transportation loss of $\mathbf{D}_1$ that is caused by adding the y-axis offset. When the x-axis offset is added, on the other hand, the ratio of w_3 gets smaller. Similarly to Type B, because the x-axis offset increases the transportation loss with $\mathbf{D}_3$, this can be considered to alleviate the increasing transportation loss of $\mathbf{D}_3$. Consequently, we conclude that the proposed weight calculation of $\boldsymbol{w}$ can adaptively reduce the deviations between two sequences by lowering those weights.

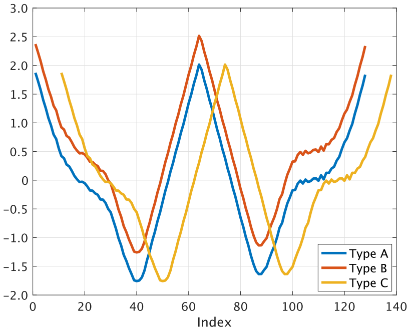

Fig. 1. Sample sequence and modified sequences

TABLE II
CALCULATED WEIGHT w_i (UPPER ROW) AND ITS RATIO TO $\|\boldsymbol{w}\|_1$ (LOWER ROW). PARENTHESIZED VALUES AT LOWER ROWS INDICATE INCREASE-DECREASE RATES COMPARED TO THE VALUES IN TYPE A.

sequence	w_1(similarity)	w_2 (derivative)	w_3 (position)
A	10.57	5.69	3.96
(original)	52.3%	28.1%	19.6%
B	3.82	4.93	3.80
(y-offset)	30.4% (−9.7%)	39.3% (+4.5%)	30.3% (+5.2%)
C	9.55	5.32	2.59
(x-offset)	54.7% (+2.4%)	30.5% (+2.3%)	14.8% (−4.8%)

B. Classification Performances

The classification performance is measured by 1-nearest neighbor (1-NN) algorithm. The parameters of AWSWD are set to $l = 5$ and $\lambda = 50$. As for SwedishLeaf and FaceAll, we observed in the preliminary experiment that the classification performance is significantly high when subsequence length l is small. Thereby, we set $l = 5$. γ is set to 0.001 for Soft-DTW, and $\lambda_1 = 0.1, \lambda_2 = 0.01, \sigma = 5$ for OPW. As for shapeDTW, we use HOG1D as the shape descriptor for single dimensional data and Raw-Subsequence for multi dimensional data [4], and set the subsequence length to 7. The accuracy (Acc) and mean Average Precision (mAP) are used for measuring the performances. The results are shown in TABLEs III and IV. The best and second performances are represented in underlined bold and just bold. The results demonstrated that the classification performances of the proposed AWSWD are comparable to or outperform other state-of-the-art methods. Especially, the proposed AWSWD outperforms the OPW distance, which is also based on the OT distance, on almost all datasets. This indicates the effectiveness of the consideration of neighborhood structure on the OT-based sequence matching.

V. CONCLUSION

We have presented a shape-aware Wasserstein distance between sequence data building upon optimal transport framework. The proposed AWSWD considers the shape of sequences around each elements in addition to distances of elements and their order position. Then it calculates their

TABLE III
CLASSIFICATION PERFORMANCES (ACCURACY: ACC).

method	DTW [2]	SoftDTW [7]	DDTW [3]	ShapeDTW [4]	OPW [8]	AWSWD
Swedish-Leaf	0.7920	0.7888	0.8784	**0.9040**	**0.9040**	**0.9408**
FaceAll	0.8077	0.8077	**0.8728**	0.7757	0.7929	**0.9444**
ItalyPowerDemand	0.9436	**0.9504**	0.8999	0.9201	0.8756	**0.9572**
SonyAIBORobotSurface1	0.5225	0.7255	0.7521	0.7388	**0.9118**	**0.9068**
SAD	**0.9636**	**0.9636**	0.9077	**0.9691**	0.8905	0.9482
HAS	0.8135	**0.8143**	0.7038	0.7887	**0.8173**	0.7489
MSRAction3D	**0.7692**	**0.7692**	0.7070	**0.7766**	0.6667	0.7619

TABLE IV
CLASSIFICATION PERFORMANCES (MEAN AVERAGE PRECISION: MAP).

method	DTW [2]	SoftDTW [7]	DDTW [3]	ShapeDTW [4]	OPW [8]	AWSWD
Swedish-Leaf	0.4928	0.4932	**0.6510**	0.6661	0.5643	**0.7210**
FaceAll	**0.5483**	**0.5483**	0.5185	0.5189	0.5269	**0.6142**
ItalyPowerDemand	0.7335	0.7335	0.7028	**0.7686**	0.6629	**0.7717**
SonyAIBORobotSurface1	0.6513	0.6512	0.7095	0.6652	**0.7418**	**0.7798**
SAD	0.5658	0.5658	0.4403	**0.5938**	0.3917	**0.5830**
HAS	**0.4461**	**0.4461**	0.4166	0.3642	0.4211	**0.4616**
MSRAction3D	**0.5343**	**0.5343**	0.4787	**0.5419**	0.4718	0.5070

weights automatically. As presented herein, numerical evaluation demonstrated the effectiveness of AWSWD for classification tasks. Our future work includes development of faster optimization algorithms, and applications to real-world problems that involve video alignments and purchase predictions.

ACKNOWLEDGMENT

H. Kasai was partially supported by JSPS KAKENHI Grant Numbers JP16K00031 and JP17H01732, and by Support Center for Advanced Telecomm. Technology Research (SCAT).

REFERENCES

[1] H. Narimatsu and H. Kasai, "State duration and interval modeling in hidden semi-Markov model for sequential data analysis," Annals of Mathematics and Artificial Intelligence, vol. 81, no. 3-4, pp. 377–403, 2017.

[2] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," IEEE transactions on acoustics, speech, and signal processing, vol. 26, no. 1, pp. 43–49, 1978.

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

[4] J. Zhao and L. Itti, "shapeDTW: shape dynamic time warping," Pattern Recognition, vol. 74, pp. 171–184, 2018.

[5] J. Yuan, Q. Lin, W. Zhang, and Z. Wang, "Locally slope-based dynamic time warping for time series classification," in CIKM, 2019.

[6] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," Pattern recognition, vol. 44, no. 9, pp. 2231–2240, 2011.

[7] M. Cuturi and M. Blondel, "Soft-DTW: a differentiable loss function for time-series," in International Conference on Machine Learning, 2017.

[8] B. Su and G. Hua, "Order-preserving Wasserstein distance for sequence matching," in CVPR, 2017.

[9] S. Salvador and P. Chan, "Toward accurate dynamic time warping in linear time and space," Intelligent Data Analysis, vol. 11, no. 5, pp. 561–580, 2007.

[10] G. Al-Naymat, S. Chawla, and J. Taheri, "SparseDTW: A novel approach to speed up dynamic time warping," arXiv preprint arXiv:1201.2969, 2012.

[11] D. F. Silva and G. E. A. P. A. Batista, "Speeding up all-pairwise dynamic time warping matrix calculation," in ICDM, 2016.

[12] F. Itakura, "Minimum prediction residual principle applied to speech recognition," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 23, no. 1, pp. 67–72, 1975.

[13] R. Sinkhorn, "Diagonal equivalence to matrices with prescribed row and column sums," Proc. Am. Math. Soc., vol. 45, no. 2, pp. 195–198, 1974.

[14] J. Qiu, X. Wang, P. Fua, and D. Tao, "Matching seqlets: An unsupervised approach for locality preserving sequence matching," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 2, pp. 745–752, 2021.

[15] L. Kantorovich, "On the transfer of masses," Dokl. Akad. Nauk, vol. 37, no. 2, pp. 227–229, 1942.

[16] C. Villani, Optimal Transport: Old And New, Springer, 2008.

[17] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 9, pp. 1853–1865, 2017.

[18] T. Fukunaga and H. Kasai, "Wasserstein k-means with sparse simplex projection," in ICPR, 2020.

[19] M. Cuturi and A. Doucet, "Fast computation of wasserstein barycenters," in ICML, 2014.

[20] H. Kasai, "Multi-view Wasserstein discriminant analysis with entropic regularized Wasserstein distance," in ICASSP, 2020.

[21] T. Fukunaga and H. Kasai, "Block-coordinate Frank–Wolfe algorithm and convergence analysis for semi-relaxed optimal transport problem," in ICASSP, 2022.

[22] N. Kolkin, J. Salavon, and G. Shakhnarovich, "Style transfer by relaxed optimal transport and self-similarity," in CVPR, 2019.

[23] J. Huang, Z. Fang, and H. Kasai, "LCS graph kernel based on Wasserstein distance in longest common subsequence metric space," Digital Signal Processing, vol. 189, pp. 108281, 2021.

[24] J. Huang and H. Kasai, "Graph embedding using multi-layer adjacent point merging model," in ICASSP, 2021.

[25] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in NIPS, 2013.

[26] G. Peyré and M. Cuturi, "Computational optimal transport," Foundations and Trends in Machine Learning, vol. 11, no. 5-6, pp. 355–602, 2019.

[27] F. Nie, J. Li, and X. Li, "Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification.," in IJCAI, 2016.

[28] M. W. Kadous, Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series, Ph.D. thesis, School of Computer Science and Engineering, University of New South Wales, 2002.

[29] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in CVPR - Workshops, 2010.

[30] J. Wang, Z.Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in CVPR, 2012.