Constrained AdaBoost for Totally-Ordered Global Features

Ryota Ogata*, Minoru Mori†, Volkmar Frinken* and Seiichi Uchida*
*Faculty of Information Science and Electrical Engineering, Kyushu University, Fukuoka, 819-0395 Japan
Email: {vfrinken, uchida}@ait.kyushu-u.ac.jp
†NTT Communication Science Laboratories, NTT Corporation, Kanagawa, 243-0198 Japan
Email: mori.minoru@lab.ntt.co.jp

***Abstract*—This paper proposes a constrained AdaBoost algorithm for utilizing global features in a dynamic time warping (DTW) framework. Global features are defined as a spatial relationship between temporally-distant points of a temporal pattern and are useful to represent global structure of the pattern. An example is the spatial relationship between the first and the last points of a handwritten pattern of the digit "0". Those temporally-distant points should be spatially close enough to form a closed circle, whereas those points of "6" should be distant enough. For a temporal pattern of an N-point sequence, it is possible to have $N(N-1)/2$ global features. One problem of using the global features is that they are not ordered as a one-dimensional sequence any more. Consequently, it is impossible to use them in a left-to-right Markovian model, such as DTW and HMM. The proposed constrained AdaBoost algorithm can select a totally-ordered subset from the set of $N(N-1)/2$ global features. Since the totally-ordered features can be arranged as a one-dimensional sequence, they can be incorporated into a DTW framework for compensating nonlinear temporal fluctuation. Since the selection is governed by the AdaBoost framework, the selected features can retain discriminative power.**

I. Introduction

In general, writing a character is a process with a high-order Markov property. By representing a writing process as a temporal sequence $P = p_1, \dots, p_n, \dots, p_N$, where p_n is the pen-tip position at time n, i.e., $p_n = (x_n, y_n)^T$, the high-order Markov property is defined as the mutual dependency between the pen-tip positions at temporally-distant points, n and n', $|n' - n| > 1$.

An intuitive example of the high-order Markov property is the writing process of the digit "0". In this case, the first and last points of P should be spatially close to each other, i.e., $p_1 \simeq p_N$, for forming a closed loop of "0". That is, the temporally-distant points depend on each other. We also can find another example of the high-order Markov property in the writing process "0"; for keeping the global shape of "0" circular, two temporally-distant points s and t of "0" in Fig. 1 should also be spatially distant, i.e., $p_s \not\simeq p_t$.

Typical online character recognition methods disregard the high-order Markovian property despite its importance. In fact, two very common methods, dynamic time warping (DTW) and hidden Markov model (HMM), assume just the first-order Markov property, that is, p_n depends only on p_{n-1}. This simple assumption is employed just for computational efficiency in the matching procedures of DTW and the decoding procedures of HMM. Specifically, under the first-order Markov property, a pattern P is represented only by local features, such as local direction feature $p_n - p_{n-1}$ and xy-coordinate feature p_n, and they can be arranged as a one-dimensional sequence along with n. Consequently, it is possible to employ the efficient dynamic programming (DP) optimization scheme into those procedures. The above examples on "0", however, suggest that this assumption is not accurate for modeling our character writing process.

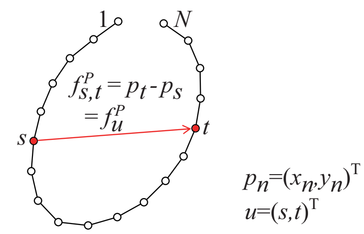

Fig. 1. Global feature defined between temporally-distant points s and t.

The global feature proposed by Mori et al. [1], [2] directly represents the spatial relationship between a pair of temporally-distant points. Specifically, the global feature between $n = s$ and t of Fig. 1 is defined as a two-dimensional vector $f^P_{s,t}$ showing their relative direction, i.e., $f^P_{s,t} = p_t - p_s = (x_t - x_s, y_t - y_s)^T$. For a pattern P of an N-point sequence, it is possible to have $N(N-1)/2$ global features. This simple feature has a significant potential to represent the high-order Markov property. For example, it is sufficient to require the global feature $p_N - p_1$ to be close to $(0, 0)^T$, for forming the "0" as a closed loop.

The main contribution of this paper is to propose a constrained AdaBoost algorithm for fully utilizing the global features in a DTW-based recognition framework. In general, for applying DTW, a temporal pattern P should be represented by a *totally-ordered set* of features. As detailed later, this total-ordering condition is equivalent to that the features can be arranged as a one-dimensional sequence and thus if this condition holds we can use DP as usual. The set of all $N(N-1)/2$ global features of P, however, is not totally-ordered, and thus it is not possible to use them directly in DTW. The proposed constrained AdaBoost algorithm extracts a totally-ordered subset of global features (i.e., selects global features which satisfy the total-ordering condition from $N(N-1)/2$ features) for making use of their potential in DTW.

The rest of this paper is organized as follows. After a brief review of related work in Section II, a total-ordering scheme of global features is detailed in Section III. Then in Section IV the selection of totally-ordered global features by the constrained AdaBoost is proposed. Section V describes a DP-based DTW algorithm which can deal with totally-ordered global features and Section VI gives a multi-start strategy for improving the recognition performance of the algorithm. Finally, in Section VII, the performance of the selected totally-ordered global features is evaluated through a DTW-based online digit recognition experiment.

II. Related Work

As noted in Section I, typical online character recognition methods have relied on features representing the 0th-order and/or 1st-order Markov property, such as xy-coordinate feature, $p_n = (x_n, y_n)^T$, and local direction features, $f^P_{n-1,n} = p_n - p_{n-1}$ (e.g., [3], [4]). The same local features have still been exploited in recent methods [5]–[7]. On the classification stage, DTW or HMM have been often used for online character recognition [8], [9] and they have utilized local features like xy-coordinate features and local directions (e.g., [10]), because these classification methods can handle features with the 0th- and/or 1st-order Markov property.

Alternatively, some authors have proposed the relative vector between an arbitrary point pair on the stroke as global features [2]. Although several researchers, e.g. [11], have had preliminary trials of global features, experimental evaluation in [2] showed the existence of the unique global structure on each character class and its efficiency on the recognition accuracy. Moreover, some authors have proposed a preliminary trial to directly handle global features for DTW [1]. This method utilizes the F-ratio values for selecting ordered features to ensure the consistency with the Markov property of the DP matching and has enhanced the recognition accuracy. Unfortunately the improvement of recognition accuracy was limited because of the insufficiency of feature selection.

Recently, long-distance dependencies are introduced into recurrent neural network (RNN), i.e., artificial neural networks, for processing sequential patterns. It is called long short-term memory (LSTM) neural network [12]. In this network architecture, memory cells are simulated to store information over arbitrarily long time step. As a result, LSTM NN have been shown to perform very well for various sequence-based recognition tasks [13], [14]. The advantages of LSTM NN, however, come at the cost of added complexity which, in turn, requires large amounts of training data to perform well.

III. Global Features and Their Total-Ordering

A. Definition [1], [2]

As noted in Section I and shown in Fig. 1, our global feature is a two-dimensional vector $f^P_{s,t} = p_t - p_s = (x_t - x_s, y_t - y_s)^T$ showing the relative direction from s to t. We can assume $s < t$ because $f^P_{s,t}$ is identical to $f^P_{t,s}$ in a pattern classification task. Then, for a N-point sequence pattern P, it is possible to have a set of $N(N-1)/2$ global features. Note that this set includes "local" directional features, i.e., $f^P_{n-1,n}$, which have been widely used in online character recognition.

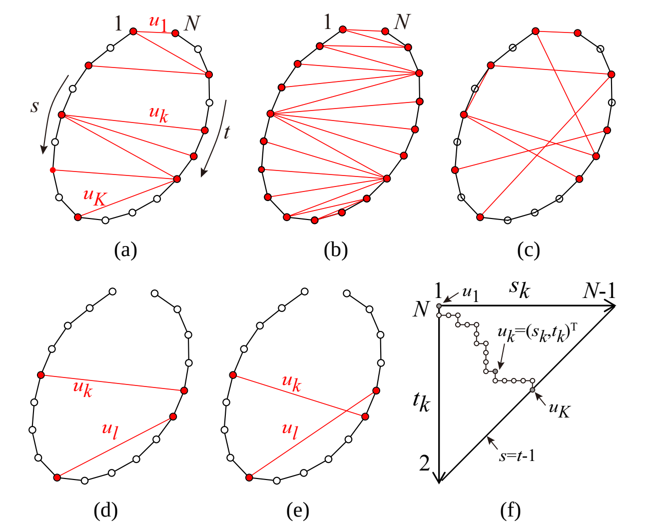

Fig. 2. (a) A totally-ordered set of global features. (b) Another totally-ordered set with more global features. (c) A set of global features without total order. (d) A pair of global features satisfying $u_k \prec u_l$. (e) A pair of global features satisfying neither $u_k \prec u_l$ nor $u_l \prec u_k$. (f) Representation of total-order by a monotonic path in the $s-t$ plane.

Hereafter, for notational simplicity, we introduce a variable $u = (s,t)^T \in \mathcal{U}$, where $\mathcal{U} = \{(s,t)^T \mid s \in [1, N-1],\ t \in [s+1, N]\}$. Since each $u \in \mathcal{U}$ corresponds one global feature $f^P_u = f^P_{s,t} = p_t - p_s$, we also refer to u as global feature unless otherwise mentioned. Also for simplicity, we assume the number of points, N, is fixed for all patterns by using some resampling procedure.

B. Totally-ordered set of global features

1) Illustrative explanation: Now we define the *totally-ordered set* of global features for using the global features in a DTW framework. As noted before, the elements of a totally-ordered set can be arranged as a one-dimensional sequence and thus can be considered as an sequential pattern. Consequently, if we have a totally-ordered set of global features, we can use those global features in the DTW framework.

In advance to a formal and detailed description, we shall start from an illustrative example of Fig. 2 (a), which shows a totally-ordered set of global features for "0". In fact, the global features in (a) can be ordered from top ($u_1 = (1, N)^T$) to bottom. This kind of ordering is obtained by increasing s and decreasing t monotonically, as indicated by arrows in (a). Figure 2 (b) shows a more dense set of $N-1$ global features with a similar ordering as (a). In contrast, it is impossible to introduce any order into global features in Fig. 2 (c). Note that since "0" is a convex pattern, the total-ordering of its global features is understandable clearly by the absence of intersecting features, as shown in Figs. 2 (a) and (b). For other (non-convex) digit patterns, the global features may show intersections even though they are totally-ordered (as we will see in Fig. 8).

For deriving a totally-ordered set of global features, we need to extract a special (i.e., totally-ordered) *subset* from $\mathcal{U}$. In other words, the entire set $\mathcal{U}$ itself is not a totally-ordered set, since $\mathcal{U}$ contains all possible global features and they are mutually intersecting like Fig. 2 (c). An extraction algorithm of a totally-ordered subset will be proposed in Section IV.

2) Detailed explanation: Formally, a set A is a totally-ordered set if its two different elements a_i and a_j are *comparable*; that is, $a_i \prec a_j$ or $a_j \prec a_i$ will hold always for a binary relation "$\prec$". Note that "$a_i \prec a_j$" means that a_i precedes a_j by a certain ordering rule. The simplest totally-ordered set is a set of integers because any two different integers i and j satisfy one of the binary relations, $i < j$ or $j < i$.

As an ordering rule of global features, consider the following binary relation:

$$u_k \prec u_l \Leftrightarrow (s_k \leq s_l) \wedge (t_k \geq t_l), \tag{1}$$

where $\wedge$ is the logical conjunction operator. Note that we restrict that the two equalities in (1) do not hold simultaneously and thus $u_k \neq u_l$. Then, a set of global features $\widetilde{\mathcal{U}} = \{u_1, \ldots, u_k, \ldots, u_K\}$ $(u_k = (s_k, t_k)^T \in \mathcal{U})$ is defined as a totally-ordered set, if its any two elements u_k and u_l are always comparable, that is,

$$u_k \prec u_l \text{ or } u_k \succ u_l, \ \forall u_k, u_l \in \widetilde{\mathcal{U}}, \ k \neq l. \tag{2}$$

Figure 2 (d) is a comparable pair of global features and (e) is not. Accordingly, if a set includes two global features like (e), the set is not a totally-ordered set.

An important property of a totally-ordered set $\widetilde{\mathcal{U}}$ is that its all elements can be arranged as a one-dimensional sequence $u_1, \ldots, u_k, u_{k+1}, \ldots, u_K$, which satisfies

$$u_1 \prec \cdots \prec u_k \prec u_{k+1} \prec \cdots \prec u_K, \tag{3}$$

The above conditions (1) and (3) mean that the sequence $u_1, \ldots, u_k, \ldots, u_K$ forms a monotonic path on $s-t$ plane, as shown in Fig. 2 (f). Since $s_k < t_k$, the path ends at some position on the diagonal line $s = t - 1$. The number of the points $\{(s_k, t_k)\}$ in this triangular plane of (f) equals to the number of all possible global features, $|\mathcal{U}| = N(N-1)/2$. Since the maximum length of a monotonic path in the plane is $N - 1$, we can say $K \leq \max |\widetilde{\mathcal{U}}| = N - 1$. Accordingly, Fig. 2 (b) shows a case where the maximum number of totally-ordered features are given.

IV. Constrained AdaBoost for Selecting Totally-Ordered Global Features

As discussed in the previous section, we need to extract a totally-ordered sequence $u_1, \ldots, u_k, \ldots, u_K$ satisfying (3) from $\mathcal{U}$. Generally, the subset extraction problem can be considered as a feature selection problem and thus it can be solved by machine learning algorithm with a feature selection function, such as AdaBoost, decision tree, and random forest. In this paper, we employ AdaBoost with a modification discussed below.

Initialize W (as the general AdaBoost).
$\mathcal{U}^1 := \mathcal{U}$ and $\widetilde{\mathcal{U}} := \emptyset$.
for $l = 1$ **to** K **do**
  Find the best weak learner $h^l(u^l), \ u^l \in \mathcal{U}^l$, under the current weight W.
  Update W and calculate the reliability α^l of $h^l(u^l)$ (as the general AdaBoost).
  $\widetilde{\mathcal{U}} := \widetilde{\mathcal{U}} \cup u^l$.
  Generate the subset $\mathcal{U}^{l+1}$ whose elements satisfy the condition (2) with $\widetilde{\mathcal{U}}$.
endfor
Arrange the K elements of $\widetilde{\mathcal{U}}$ to be a sequence $u_1, \ldots, u_K$ which satisfies (3).

Fig. 3. Pseudo-code of the constrained AdaBoost algorithm.

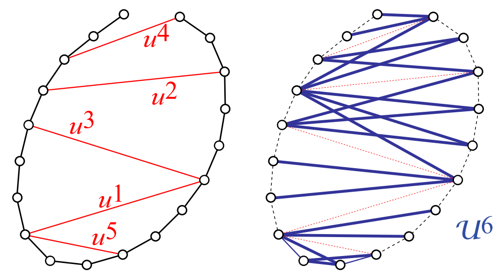

Fig. 4. A set $\mathcal{U}^l$ of selectable features (left), for given $\{u^1, \ldots, u^{l-1}\}$ (right), where $l = 6$.

A. Overview of the general AdaBoost

AdaBoost is a well-known machine learning method for classification. The basic idea of AdaBoost is to repeat selection of a binary classifier, called weak learner, until convergence. By repeating the selection K times, we have a sequence of weak learners, $h^1(\boldsymbol{u}), \ldots, h^l(\boldsymbol{u}), \ldots, h^K(\boldsymbol{u})$, along with their reliability $\alpha^1, \ldots, \alpha^l, \ldots, \alpha^K$, where $\boldsymbol{u}$ is a $|\mathcal{U}|$-dimensional feature vector comprised of the element of a set $\mathcal{U}$ of all possible features representing a pattern P and the output of h^l is 1 or -1[1].

Those weak learners are selected to be complementary; the selection of the l-th weak learner h^l tries to recognize the training samples mis-recognized by the previous weak learners $h^1(\boldsymbol{u}), \ldots, h^l(\boldsymbol{u}), \ldots, h^{l-1}(\boldsymbol{u})$. This procedure is realized by using a weight vector W whose elements are the current weights of individual training samples. (A misrecognized sample will have a larger weight.) By penalizing the individual misrecognitions by W, $h^l(u^l)$ with the minimum penalty will be selected. For more details of the general AdaBoost, see [15].

AdaBoost can also be used for feature selection by using only one element of $\boldsymbol{u}$ as the input of each weak learner. Consequently, at the l-th iteration, among $|\mathcal{U}|$ weak learner candidates, $\{h^l(u^1), \ldots, h^l(u^{|\mathcal{U}|})\}$, the most discriminative

[1]In our notation, when the element of a set is indexed by a superscript, such as u^l, h^l, α^l, the set is not totally-ordered. For example,$u^1, \ldots, u^l, \ldots, u^K$ is not ordered and $u_1, \ldots, u_l, \ldots, u_K$ is totally-ordered.

```
1:  G_1[v_1] := d(u_1, v_1) where v_1 = (1, N)^T.
2:  for k = 2 to K do
3:      for ∀v_k ∈ V do
4:          Calculate G_k[v_k] by (5) using u_k and u_{k-1}.
5:      endfor
6:  endfor
7:  min J := min_{v_K} G_K[v_K].
```

Fig. 5. Pseudo-code of DTW for totally-ordered global features.

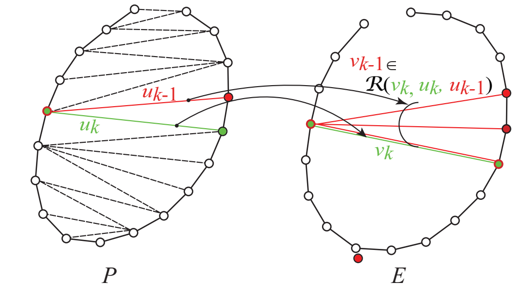

Fig. 6. Correspondence of global features between a pair of patterns P and E and a set $\mathcal{R}(v_k, u_k, uk-1)$ of v_{k-1} for realizing the continuous and monotonic DTW.

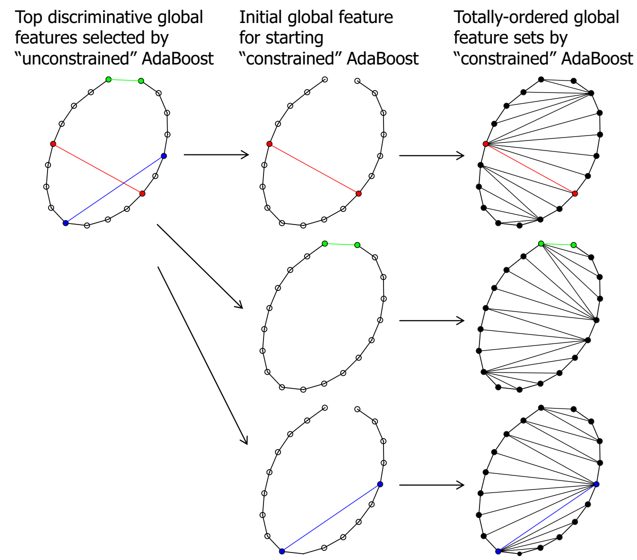

Fig. 7. Multi-start strategy ($M = 3$).

candidate $h^l(u^l)$ (u^l is an element of $\boldsymbol{u}$) is selected as the l-th weak leaner. This can be interpreted that the feature u^l is selected by the l-th weak learner. Accordingly, it is possible to select K discriminative features $u^1, \ldots, u^K$ by K weak learners. One of the most famous examples of using AdaBoost as feature selection is Viola and Jones [16], where features for face detection are selected by AdaBoost.

B. Constrained AdaBoost

We now propose a *constrained AdaBoost* method for providing a set of weak learners of $h^1(u^1), \ldots, h^K(u^K)$ whose features $u^1, \ldots, u^K$ can be an ordered sequence by satisfying the condition (2). Figure 3 is the pseudo-code of the constrained AdaBoost, where Steps 1, 3, 5 and 8 are the same as the general AdaBoost. The set $\widetilde{\mathcal{U}}$ is the set of the ordered global features already selected until the l-th iteration.

The main difference from the general AdaBoost is Step 4, where selectable global features for the l-th weak learner is limited to $\mathcal{U}^l$, instead of all features $\mathcal{U}$ for the general AdaBoost. The set $\mathcal{U}^l$ is a subset of $\mathcal{U}$ and comprised of the global features being consistent with $\widetilde{\mathcal{U}}$ in the sense of total-ordering condition (1). Formally the set $\mathcal{U}^l$ is expressed as follows:

$$\mathcal{U}^l = \{u \in \mathcal{U} \mid u \prec u' \text{ or } u \succ u', \forall u' \in \widetilde{\mathcal{U}} = \{u^1, \ldots, u^l\}\}. \tag{4}$$

Step 7 in Fig. 3 is the procedure for generating $\mathcal{U}^l$ and Fig. 4 illustrates an example of generating $\mathcal{U}^l$ for given $\widetilde{\mathcal{U}} = \{u^1, \ldots, u^{l-1}\}$. No global feature of $\mathcal{U}^l$ will cause the situation like Fig. 2 (e) with $u^1, \ldots, u^5$. Step 9 creates the ordered sequence of global features from the totally-ordered set $\widetilde{\mathcal{U}}$; this is possible because of the property of the totally-ordered set as noted in III-B2.

V. DTW WITH TOTALLY-ORDERED GLOBAL FEATURES

Representing P by an ordered sequence of global features $u_1, \ldots, u_k, \ldots, u_K$, we can apply the DP algorithm for establishing the optimal DTW between P and an input pattern E with their global feature representation. The application of DTW will be useful to compensate nonlinear temporal fluctuation between P and E and thus contribute to achieve a better recognition accuracy.

Let $v_k = (S_k, T_k)^T \in \mathcal{V}$ denote the global feature on E corresponding to u_k on P, where $\mathcal{V}$ is the set of all possible global features on E. Our task is to determine the optimal totally-ordered sequence $v_1, \ldots, v_k, \ldots, v_K$ so that the objective function $J = \sum_k d(u_k, v_k)$ is minimized, where $d(u_k, v_k)$ is a distance function between two global features, $f^P_{u_k}$ and $f^E_{v_k}$. In this paper, we assume the Euclidean distance, i.e., $d(u_k, v_k) = \|f^P_{u_k} - f^E_{v_k}\|$.

Figure 5 shows a pseudo-code of DP algorithm for solving the DTW problem with totally-ordered global features. The main step of the algorithm is the following DP-recursion:

$$G_k[v_k] = d(u_k, v_k) + \min_{v_{k-1} \in \mathcal{R}(v_k, u_k, u_{k-1})} G_{k-1}[v_{k-1}], \tag{5}$$

where $G_k[v_k]$ is the minimum accumulated distance cost from v_1 to v_k. Thus, $\min_{v_K} G_K[v_K]$ at Step 7 provides the minimum value of the objective function J achieved by the optimal sequence of $v_1, \ldots, v_K$. The set $\mathcal{R}(v_k, u_k, u_{k-1})$ contains all v_{k-1} which can precede v_k while keeping continuity and monotonicity between them. Figure 6 illustrates $\mathcal{R}(v_k, u_k, u_{k-1})$ in a specific case[2].

Although the input pattern E is not initially represented as a totally-ordered global feature sequence, the optimal $v_1, \ldots, v_K$ becomes a totally-ordered global feature sequence automatically. In other words, a totally-ordered sequence is extracted from the input pattern E as the result of DTW

[2] In this case, $S_k = S_{k-1}$ because $s_k = s_{k-1}$.

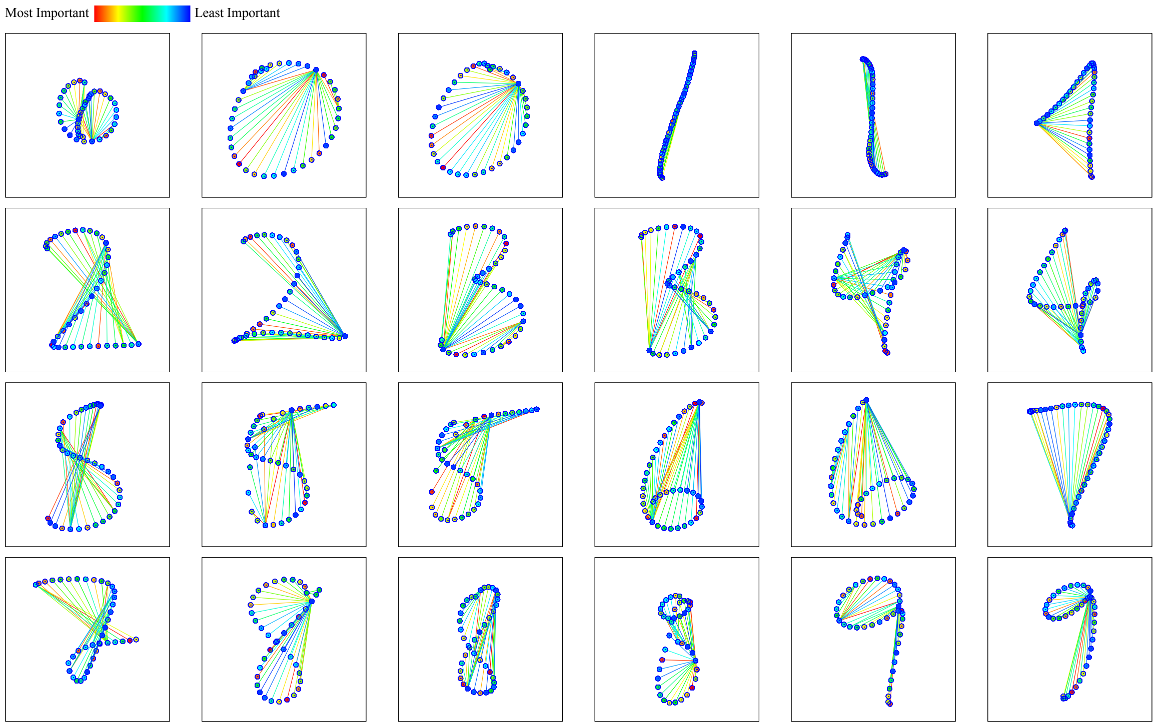

Fig. 8. Selection result of totally-ordered global features at each of 24 sub-classes.

with P. This is realized by the continuity and monotonicity constraints of DTW, given by the set $\mathcal{R}(v_k, u_k, u_{k-1})$.

VI. Multi-start Strategy

Since AdaBoost is based on a greedy optimization strategy to minimize recognition error, its performance depends on the initial condition, which is given as u^1 in Fig. 3. In the constrained AdaBoost, this dependency becomes even stronger because u^1 restricts the selection of the later global features $u^2, u^3, \ldots,$ tightly. An extreme example is the case of $u^1 = (1,2)^T$, which restricts u^l as $(1,\cdot)^T$ for all $l > 1$. This means that all of the selected global features will have its one end always at $n = 1$, i.e., $s_k \equiv 1$.

In order to avoid this severe dependency on the first global feature, we will employ a multi-start strategy as a simple remedy. Figure 7 illustrates the multi-start strategy. First, like [2], top-M discriminative global features are selected without the total-ordering constraints, i.e., by the general AdaBoost. Then, constrained AdaBoost is launched while using each of them as the initial feature u^1. Consequently, it is possible to have M different global feature representations for each pattern.

VII. Experimental Result

A. Dataset

We conducted an experiment of totally-ordered global feature selection and DTW-based recognition with the selected features by using digit samples (“0”- “9”) in the UNIPEN database [17]. The numbers of training and test samples for 10 digit classes were 14 124 and 1 569 in total. As preprocessing, each sample was linearly scaled to fit to the 128×128 bounding box while keeping the original aspect ratio and then resampled to be a $N = 40$-point temporal pattern. To deal with variations in stroke-order, stroke-number, and writing-direction, the training samples were clustered into 24 sub-classes by k-means (each of 10 digit has 2∼3 sub-classes) and then global feature selection was done at each sub-class independently using the constrained AdaBoost in the one-vs-others manner.

The DTW-based recognition experiment was performed to match the totally-ordered global features sequence representing a reference pattern to the input pattern E. The class of the reference pattern giving the minimum DTW cost was treated as the recognition result of the input pattern. Note that using the multi-start strategy, we have M different representations for each training patterns, and thus DTW of Fig. 5 was performed $14\,124 \times M$ times for each input pattern.

B. Result of totally-ordered global feature selection

Figure 8 shows the result of totally-ordered global feature selection for each sub-class. It is possible to observe that the selected features for “0” are totally-ordered. Although it is difficult to confirm the order of the selected features for the other classes (e.g., “2”), a closer inspection proves their total ordering.

Color in Fig. 8 indicates the importance of the global features. Specifically, the red-colored global feature is the feature selected at first by the constrained AdaBoost and thus the most important (i.e., discriminative) global feature. For the digit “0”, important global features are selected between pairs of diagonal points. Again, selecting these features is reasonable for forming the circular shape of “0”. In contrast,

(a) The most important feature u^1 (b) The second most important feature u^2

Fig. 9. Selected global features of the class "6".

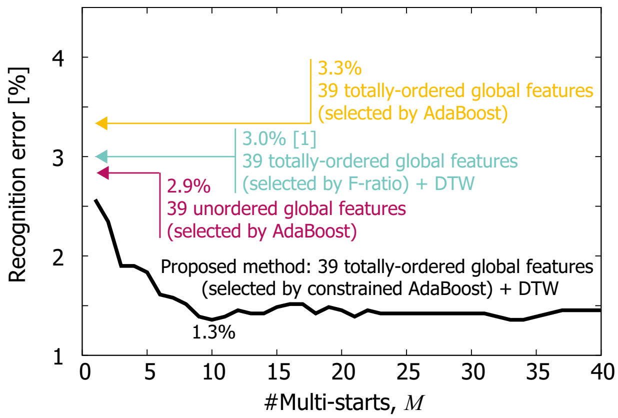

Fig. 10. Recognition error rates by different methods using global features.

the blue-colored global feature is the feature selected at last and thus the least important.

Figures 9(a) and (b) show the first and the second global features, respectively, for "0" and "6". The first global features, u^1, of both classes are selected similarly between a beginning point and a middle point, and thus less discriminative for these class pairs. In contrast, the second features, u^2, are selected from the very different time points and thus helpful for discriminating those classes.

C. Result of DTW- based recognition

Figure 10 shows the recognition error rates by different methods using global features and proves the following facts:

- The proposed method achieved the lowest error 1.3% with the multi-start strategy. Even without the multi-start strategy, the proposed feature selection by the constrained AdaBoost achieved lower rate 2.5% than other methods.
- If DTW function is removed, that is, if $s_k = S_k$ and $t_k = T_k$ for all k, the accuracy degrades to 3.3%. This result indicates that the introduction of DTW for matching global features is meaningful.
- If we do not introduce DTW, we do not need to introduce the total-ordering constraint. In this "unordered" case (like [2]), the error rate became 2.9%. This is higher than 3.3% of the constrained case because the unordered case has more flexibility in feature selection. However, since the error rate of the constrained case is improved from 3.3% to 2.5% by introducing DTW, it is possible to conclude the ordering constraint is important.
- The method of [1], where the totally-ordered features are selected in a different criterion (of F-ratio), achieved 3.0% and thus it is shown that the proposed AdaBoost framework could provide more reasonable global features.

VIII. Conclusion

In this paper, we have proposed a constrained AdaBoost for utilizing global features in a DTW framework. Although global features are promising for online character recognition, introducing them into the DTW framework is not straightforward. This is because the set of all possible global features is not ordered and thus difficult to be dealt in the one-dimensional optimization process of DTW. The proposed constrained AdaBoost is the algorithm to select a totally-ordered subset of global features while keeping the discrimination ability of global features. The totally-ordered global features are arranged as a one-dimensional sequence, and thus now DTW can deal with them. A DTW-based online character recognition experiment has showed that the proposed method is promising for utilizing the global features.

Acknowledgment

This work is supported in part by the CREST project from Japan Society for the Promotion of Science (JSPS).

References

[1] M. Mori, S. Uchida and H. Sakano, "Dynamic programming matching with global features for online character recognition," Proc. ICFHR, 348–353, 2012.

[2] M. Mori, S. Uchida and H. Sakano, "Global feature for online character recognition," Pattern Recognition Letters, 35, 142–148, 2014.

[3] C. C. Tappert, C. Y. Suen, and T. Wakahara, "The state of the art in online handwriting recognition," IEEE Trans. PAMI, 12(8), 787–808, 1990.

[4] C. Bahlmann, "Directional feature in online handwriting recognition," Pattern Recognition 39 (1), 115–125, 2006.

[5] T. V. Phan, J. Gao, B. Zhu and M. Nakagawa, Effects of Line Densities on Nonlinear Normalization for Online Handwritten Japanese Character Recognition, Proc. ICDAR, 834–838, 2011.

[6] B. Zhu and M. Nakagawa, "On-line handwritten Japanese characters recognition using a MRF model with parameter optimization by CRF," Proc. ICDAR, 603–607, 2011.

[7] N. Bhattacharya, U. Pal, and F. Kimura, "A system for Bangla online handwritten text," Proc. ICDAR, 1335–1339, 2013.

[8] M. Cheriet, N. Kharma, C. L. Liu, and C. Y. Suen, *Character recognition systems: A Guide for Students and Practioners*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2007.

[9] C. Biswas, U. Bhattacharya, S. K. Parui, "HMM based online handwritten Bangla character recognition using Dirichlet distributions," Proc. ICFHR, 600–605. 2012.

[10] M. Kobayashi, S. Masaki, O. Miyamoto, Y. Nakagawa, Y. Komiya, and T. Matsumoto, "RAV (reparameterized angle variations) algorithm for online handwriting recognition," IJDAR, 3(3), 181–191, 2001.

[11] S. Izadi and C. Y. Suen, "Integration of contextual information in online handwriting representation," Proc. ICIAP, 132–142, 2009.

[12] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," Neural Computation, 9(8), 1735–1780, 1997.

[13] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks," Proc. IJCNN, 2047–2052, 2005.

[14] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," IEEE Trans. Pat. Anal. Mach. Intell., 31(5), 855–868, 2009.

[15] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," J. Comput. Syst. Sci. vol. 55, no. 1, pp.119–139, 1997.

[16] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proc. CVPR, pp.511–518, 2001.

[17] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet, "Unipen project of on-line data exchange and recognizer benchmarks," Proc. ICPR, 29–33, 1994.