Support Vector Machine with Dynamic Time-Alignment Kernel for Speech Recognition

Hiroshi Shimodaira†, Ken-ichi Noma‡, Mitsuru Nakai†, Shigeki Sagayama††

† School of Information Science, Japan Advanced Institute of Science and Technology
‡ Graduate School of Engineering, University of Tokyo
E-mail: {sim, knoma, mit, sagayama}@jaist.ac.jp
URL: http://www-ks.jaist.ac.jp/

Abstract
A new class of Support Vector Machine (SVM) which is applicable to sequential-pattern recognition is developed by incorporating an idea of non-linear time alignment into the kernel. Since time-alignment operation of sequential pattern is embedded in the kernel evaluation, same algorithms with the original SVM for training and classification can be employed without modifications. Furthermore, frame-wise evaluation of kernel in the proposed SVM (DTAK-SVM) enables frame-synchronous recognition of sequential pattern, which is suitable for continuous speech recognition. Preliminary experiments of speaker-dependent 6 voiced-consonants recognition demonstrated excellent recognition performance of more than 98% in correct classification rate, whereas 93% by hidden Markov models (HMMs).

1. Introduction
Support Vector Machine (SVM) is one of the latest and most successful statistical pattern classifiers [1] that utilizes a kernel function which classifies an input vector \( x \in \mathcal{R}^n \) as

\[
g(x) = w \cdot \phi(x) + b,
\]

where \( w \) is a normal vector of a separating hyper-plane in a “feature space” where the input vector \( x \) is mapped with a non-linear mapping function \( \phi(x) : \mathcal{R}^n \rightarrow \mathcal{R}^{n'}\), \( n \ll n' \), the operator “\( \cdot \)” denotes the inner product between the vectors, and \( b \) is a bias. Though this formulation is nothing but the one for linear discriminant function of non-linearly mapped input vector, what makes SVM more effective than other discriminant functions is its learning criterion. Unlike other classifiers which are trained to minimize the classification errors on training dataset, i.e. the empirical error, SVM is trained in the fashion of “structural risk minimization” to find a compromise between the empirical error and a confidence measure corresponding to generalization. The explicit formulation of learning problem is defined as a constrained optimization problem:

\[
\begin{align*}
\min_{w,b,\xi_i} \quad & \frac{1}{2} w \cdot w + C \sum_{i=1}^{N} \xi_i, \\
\text{subject to} \quad & y_i (w \cdot \phi(x_i) + b) \geq 1 - \xi_i, \\
& \xi_i \geq 0, \quad \text{for } i = 1, \cdots, N,
\end{align*}
\]

where \( x_i \in \mathcal{R}^n, \ i = 1, \cdots, N \) is a set of training vectors with corresponding targets (class labels) \( y_i \in \{-1, +1\} \), and the parameter \( C \) controls the trade-off between the two measures. Since this optimization problem is quadratic and even convex, global solutions can be found easily compared to the neural networks which always suffer from the local minima problems.

Solving the optimization problem yields

\[
\begin{align*}
w &= \sum_{i=1}^{N} \alpha_i y_i \phi(x_i), \\
b &= \sum_{i=1}^{N} \alpha_i y_i \phi(x_i) \cdot \phi(x_i) + y_i, \quad \forall i,
\end{align*}
\]

where \( \alpha_i, \ i = 1, \cdots N \) are the Lagrange multipliers corresponding to each training vector \( x_i \). Only the training vectors with non-zero coefficients \( \alpha_i \) can contribute to determine the weight vector \( w \) and called as support vectors. Utilizing what is called kernel trick: \( K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \), the final form of SVM is represented by

\[
g(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b.
\]

As a kernel function, Gaussian radial basis function is widely used:

\[
K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\gamma^2}\right).
\]

Despite the success of SVM in the field of pattern recognition such as character recognition and text classification, SVM has not been applied to speech recognition that much. This is because that SVM assumes that each sample is a vector with fixed dimension, and hence it can not deal with the variable length sequences directly. Because of this, most of the efforts that have been made so far to apply SVM to speech recognition employ linear time normalization where input feature vector sequences with different lengths are aligned to have same length [4]. Another approach is the hybrid of SVM and HMM (hidden Markov model). There have been two methods proposed for this approach. One is to use HMM as a segmentation tool or time-alignment tool [5]. Once segmentation is done for the target input speech, standard SVM without modification is applicable. The second approach is to utilize the Fisher kernel and HMM [6, 7].

In contrast to the conventional approaches, our approach is a direct extension of the original SVM for variable length sequences. The idea is to incorporate the operation of dynamic time alignment into the kernel function itself. Because of this the proposed new SVM is called “Dynamic Time-Alignment Kernel SVM (DTAK-SVM)".
2. Dynamic Time-Alignment Kernel

This section describes how the time alignment operation can be embedded into the SVM’s kernel.

Let denote a sequence of vectors as \( X = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_L) \), where \( \mathbf{x}_i \in \mathbb{R}^n \). \( L \) be the length of the sequence, and the notation \( |X| \) be sometimes used to represent the length of the sequence instead. For simplification, we at first assume the so-called linear SVM that does not employ non-linear mapping, in which the kernel operation in (7) is identical to the inner product operation.

2.1. Formulation for linear kernel

Assume we have two vector sequences \( X \) and \( V \). In case that these two patterns are equal in length, i.e., \( |X| = |V| \), the inner product between \( X \) and \( V \) can be obtained straightforward, and therefore SVM classifier is defined as the same manner in (7). On the other hand in case where the sequence lengths are different each other, the inner product can not be calculated directly. Even in such case, however, some sort of inner product can be defined if we align the frame lengths of the two patterns. To that end, let \( \psi_T(k), \psi_J(k) \) be the time-warping functions of normalized time frame for the pattern \( X \) and \( V \), respectively, and let \( "\cdot" \) be the new inner product operator instead of the original inner product \( \cdot \). Then the new inner product between the two vector sequences \( X \) and \( V \) can be given by

\[
X \cdot V = \frac{1}{L} \sum_{k=1}^{L} \mathbf{x}_{\psi_T(k)} \cdot \mathbf{v}_{\psi_J(k)}. \tag{9}
\]

There would be two possible types of time warping functions. One is a linear time-warping function and the other is a non-linear time-warping function. The linear time-warping function takes the form as

\[
\begin{align*}
\psi_T(k) &= \lfloor |X|/L \rfloor k, \tag{10} \\
\psi_J(k) &= \lfloor |V|/L \rfloor k, \tag{11}
\end{align*}
\]

where \( L \) can be either \( |X| \) or \( |V| \). As is seen from the definition above, the linear warping function is not suitable for continuous speech recognition, i.e. frame-synchronous processing, because the normalized length \( L \) should be known beforehand. On the other hand, non-linear time warping, or dynamic time warping (DTW) in other word, enables frame-synchronous processing.

Though the original DTW uses a distance measure and finds the optimal path that minimizes the accumulated distance, the normalized length \( L \) is restricted to the one having the same term as the above restriction \( \Phi \) preserves the original length of sequence at the cost of losing long-term correlations such as the one between \( x_i \) and \( x_{l_i} \). As a result, a new class of kernel can be defined by using the extended inner product introduced in the previous section:

\[
K_\phi (X, V) = \Phi(X) \circ \Phi(V) \tag{17}
\]

where \( \phi(\cdot) \) is a non-linear mapping function that is applied to each frame vector \( \mathbf{x}_i \), same as in (1). It should be noted that under the above restriction \( \Phi \) preserves the original length of sequence at the cost of losing long-term correlations such as the one between \( x_i \) and \( x_{l_i} \). As a result, a new class of kernel can be defined by using the extended inner product introduced in the previous section:

\[
K_\phi (X, V) = \Phi(X) \circ \Phi(V) \tag{17}
\]

(Proof) Generally if \( K \) is a kernel then it should be symmetric, and it satisfied the Mercer’s condition:

\[
\int_{x \in X} K(x, y) f(x) f(y) dx dy \geq 0, \quad \forall f \in L_2(X). \tag{20}
\]

It is trivial that \( K_\phi \) is symmetric as long as symmetric path is used in (15). In order to check the Mercer’s condition, it is sufficient to show that the kernel matrix \( K_\phi \) calculated from \( N \) data: \( \{X^{(1)}, \ldots, X^{(N)}\} \) is positive semi-definite [8]. In other words, for an arbitrary vector \( \mathbf{u} \in \mathbb{R}^N \), the kernel matrix \( K_\phi \) should satisfy the following inequality:

\[
\mathbf{u}^T K_\phi \mathbf{u} \geq 0, \tag{21}
\]

where

\[
(K_{\phi})_{ij} = K_\phi (X^{(i)}, X^{(j)}). \tag{22}
\]

For simplification, we here assume that all the \( N \) data are equal in its length, and omit \( m(k) \) and \( M_\phi \) in (19). Let denote the optimal warping function as \( \psi_{T}^*(k), \psi_{J}^*(k) \), then the \( (i, j) \) element of \( K_\phi \)

\[
(K_{\phi})_{ij} = K_\phi (X^{(i)}, X^{(j)}) = \sum_{k=1}^{L} K(x_{\psi_{T}^*(k)}^{(i)}) x_{\psi_{J}^*(k)}^{(j)} = K_{ij}(1) + \cdots + K_{ij}(L), \tag{23}
\]

where

\[
K_{ij}(k) = K(x_{\psi_{T}^*(k)}^{(i)}) x_{\psi_{J}^*(k)}^{(j)}. \tag{24}
\]
Using the above notation, \( K_s \) is expressed as
\[
K_s = K^{(1)} + \cdots + K^{(L)}.
\] (25)
As a result,
\[
u^T K_s u = u^T (K^{(1)} + \cdots + K^{(L)}) u
\] (26)
\[
= u^T K^{(1)} u + \cdots + u^T K^{(L)} u
\] (27)
\[
\geq 0.
\]

3. DTAK-SVM
Using the dynamic time alignment kernel (DTAK) introduced in the last section, the discriminant function of SVM for a sequential pattern is expressed as
\[
g(X) = \sum_{i=1}^{N} \alpha_i y_i \Phi(X^{(i)}) \circ \Phi(X) + b
\] (28)
\[
= \sum_{i=1}^{N} \alpha_i y_i K_s(X^{(i)}, X) + b
\] (29)
As it can be seen from these expressions, the SVM discriminant function for time sequence has the same form with the original SVM except for the difference in kernels. It is straightforward to deduce the learning problem which is given as
\[
\min_{W, \xi} \frac{1}{2} W \circ W + C \sum_{i=1}^{N} \xi_i
\] (30)
subject to
\[
y_i(W \circ \Phi(X^{(i)}) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, N.
\] (31)
\[
\xi_i \geq 0, \quad i = 1, \ldots, N.
\] (32)
Again, since the formulation of learning problem defined above is almost the same with that for the original SVM, same training algorithms for the original SVM can be used to solve the problem.

4. Experiments
Although the proposed method can be applied to continuous speech recognition with some modifications, it is our major concern to evaluate the basic performance of the method under the condition that the positions of target patterns in the utterance are known. Therefore the performance of the proposed DTAK-SVM was evaluated by a task of speaker-dependent hand-segmented phoneme recognition.

4.1. Experimental conditions
Since the training of each SVM classifier requires not only the data belonging to the target class but also the whole data set of all the classes and hence it is time consuming, phoneme recognition experiments for the six voiced-consonants \( /b,d,g,m,n,N/ \) were conducted in the present study. The classification task of these 6 phonemes without using contextual information is considered as a relatively difficult task.
To apply the SVM which is basically formulated as a two-class classifier to the multi-class problem, what is called “one against the others” type of strategy was chosen. This type of classifier has been more successful than others in the SVM literature. The proposed method has been implemented with the publicly available toolkit, SVMTorch[9].

Other experimental conditions are shown in Table 1.

4.2. Experimental results
Phoneme classification experiments were conducted for each speaker, and the results are summarized in Fig. 1 for male speakers and Fig. 2 for female speakers. It can be seen from the figures, the best performance of 98.7% for male speakers, 97.8% for female speakers, is achieved at \( \gamma = 0.01 \) and \( C = 10 \).

Fig. 3 plots the total number of support vectors. Because 200 samples were used to train each SVM and six SVMs exist in all, the maximum total number of support vector is 1200 in this case. About a half of the training samples were chosen as the support vectors when \( \gamma = 0.01 \). Next, to compare the classification performance with HMMs, comparison experiments were carried out with varying the size of training samples (50, 100, 200), and the size of mixture (1, 4, 8, 16) for each state of HMM. Each HMM used in this experiment is a 3-states, continuous density, Gaussian mixture with diagonal covariances, context-independent model. The parameter of SVM were fixed to \( C = 10 \), \( \gamma = 0.01 \). The results for the male speaker (mau) and female speaker (ffs) are shown in Table 2 and Table 3, respectively. It can be said from those results that DTAK-SVM outperforms HMM in every cases, es-

| Phoneme classes | 6 voiced consonants: /b, d, g, m, n, N/ |
| Database       | ATR A-set: (5240 words vocabulary) |
| Speakers       | 5 males, 5 females |
| Training samples | 200 samples per phoneme per speaker, segmented out from even-numbered words |
| Testing samples | 2035 samples in all per speaker, segmented out from odd-numbered words |

| Signal sampling | 12kHz, 10ms frame-shift |
| Feature values  | 13-MFCCs and 13-AMFCCs |
| Kernel type     | RBF (radial basis function) |
| Multi-class classification strategy | one against the others type |

![Figure 1: Average correct classification rate as a function of RBF’s γ (5 male speakers)](image)
especially in the case where the size of training samples is small.

5. Conclusions

A novel approach to extend the SVM framework for the sequential-pattern classification problem has been proposed by embedding a dynamic time alignment operation into the kernel. Though long-term correlations between the feature vectors are omitted at the cost of achieving frame-synchronous processing for speech recognition, the proposed DTAK-SVMs demonstrated an excellent performance in hand-segmented voiced-consonants recognition than HMMs. Further recognition experiments using all the phoneme classes are in progress and very promising results will be presented in the next opportunity.

6. Acknowledgment

The authors would like to thank the author of SVMTorch for providing the fast training software of SVM.

Table 2: Performance comparison of DTAK-SVM with HMM for a male speaker: mau (numbers show percent correct classification rate)

<table>
<thead>
<tr>
<th># training samples/phoneme</th>
<th>50</th>
<th>100</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM (1 mix.)</td>
<td>75.0</td>
<td>69.1</td>
<td>77.1</td>
</tr>
<tr>
<td>HMM (4 mix.)</td>
<td>83.3</td>
<td>84.7</td>
<td>90.9</td>
</tr>
<tr>
<td>HMM (8 mix.)</td>
<td>82.8</td>
<td>87.0</td>
<td>92.4</td>
</tr>
<tr>
<td>HMM (16 mix.)</td>
<td>79.9</td>
<td>85.0</td>
<td>93.2</td>
</tr>
<tr>
<td>DTAK-SVM</td>
<td>98.8</td>
<td>99.0</td>
<td>99.5</td>
</tr>
</tbody>
</table>

Table 3: Performance comparison of DTAK-SVM with HMM for a female speaker: ffs (numbers show percent correct classification rate)

<table>
<thead>
<tr>
<th># training samples/phoneme</th>
<th>50</th>
<th>100</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM (1 mix.)</td>
<td>72.2</td>
<td>65.5</td>
<td>76.6</td>
</tr>
<tr>
<td>HMM (4 mix.)</td>
<td>77.3</td>
<td>76.4</td>
<td>86.4</td>
</tr>
<tr>
<td>HMM (8 mix.)</td>
<td>74.6</td>
<td>79.3</td>
<td>88.5</td>
</tr>
<tr>
<td>HMM (16 mix.)</td>
<td>72.9</td>
<td>78.7</td>
<td>89.8</td>
</tr>
<tr>
<td>DTAK-SVM</td>
<td>94.6</td>
<td>94.9</td>
<td>98.2</td>
</tr>
</tbody>
</table>

7. References