ACOUSTIC MODELING OF SUBWORD UNITS USING SUPPORT VECTOR MACHINES

C. Chandra Sekhar
Dept. of Computer Science and Engineering
Indian Institute of Technology, Madras,
Chennai-600036, India

W.F. Lee, K. Takeda and F. Itakura
CIAIR, Itakura Laboratory,
Dept. of Information Electronics,
Nagoya University, Nagoya, Japan

ABSTRACT
This paper addresses the issues in recognition of subword units of speech using support vector machines. Discriminative training and good generalization capability of support vector machines are useful in developing acoustic models for subword units when the confusability among the units is high. We compare the performance of support vector machine based systems with that of hidden Markov models in recognition of subword units. We demonstrate the better performance of support vector machines in recognition of monophone units in a large corpus of Japanese speech and recognition of consonant-vowel units in a broadcast news corpus of an Indian language.

1. INTRODUCTION
Approaches to large vocabulary continuous speech recognition are based on acoustic modeling of subword units of speech. Hidden Markov models and neural network models (such as multilayer perceptrons and recurrent neural networks) are commonly used for acoustic modeling of subword units. The focus of learning in these models is on minimizing the classification error on the training data. Support vector machines (SVMs) use a discriminative training method and the focus of learning in these models is on the generalization performance, i.e., classification error on the test data [1]. Recently, the SVM based approaches have been explored for tasks such as recognition of isolated utterances of vowels [2] and recognition of a small set of consonants in continuous speech [3]. In this paper, we focus on SVM based approaches for acoustic modeling of monophone units in Japanese speech and Consonant-Vowel (CV) units in speech for Indian languages. We first study the performance of SVM based approaches on recognition of monophone units in a large corpus of Japanese speech. Then we focus on recognition of CV units in a broadcast news corpus of an Indian language. The CV units are considered as subword units for Indian languages mainly because of the large number of stop consonants in Indian languages and the significant coarticulation effects between consonants and the following vowels [4].

The paper is organized as follows: In the next section, we describe the method for the design of a support vector machine for two-class pattern recognition. In Section 3, we describe the approaches for multi-class pattern recognition using SVMs. In Section 4, we present our studies on recognition of monophone units in Japanese speech. Our studies on recognition of CV units in Indian languages are presented in Section 5.

2. TWO-CLASS PATTERN RECOGNITION USING SVMS
A support vector machine for pattern classification is built by mapping the input pattern \( x \) into a high-dimensional feature vector \( \text{v} \) using a non-linear transformation \( g(x) \), and then constructing an optimal hyperplane in the feature space. Nonlinear transformation \( g(x) \) should be such that the pattern classes are linearly separable in the feature space, i.e., the training patterns can be classified.
without any error. We first discuss how an optimal hyperplane, with minimum generalization error, can be constructed for linearly separable patterns. Then we extend the discussion to construction of an optimal hyperplane for linearly nonseparable patterns, i.e., some training patterns cannot be classified correctly.

Suppose that the training data set consists of \( n \) examples, \( \{(x_i, y_i), i = 1, 2, ..., n\} \), where \( x_i \) is a \( d \)-dimensional pattern vector and \( y_i \in \{+1, -1\} \) is the corresponding desired output. The corresponding set of non-linearly transformed vectors, \( \{v_i\} \), are assumed to be linearly separable in the feature space. The discriminant function of a separating hyperplane that is capable of separating the training data without any error and specified by the parameters \( (w, b) \) is given by:

\[
D(v) = w^T v + b
\]  

A separating hyperplane satisfies the constraints that define the separation of the training data set:

\[
y_i(w^T v_i + b) \geq 1 \quad \text{for} \quad i = 1, 2, ..., n
\]  

The distance between a separating hyperplane and a data point \( v \) is \( D(v) / ||w|| \). The data points \( v_i \) for which the constraint in Eq.(2) is satisfied with the equality sign are called support vectors, as illustrated in Fig.1. The support vectors are the data points that lie closest to the decision surface and are therefore the most difficult to classify [5]. The distance between a support vector and the hyperplane, called the margin, is \( 1 / ||w|| \). It is intuitive that a larger margin corresponds to better generalization. The problem of finding the optimal hyperplane that maximizes the margin is equivalent to minimizing \( ||w|| \).

The learning problem of finding the optimal hyperplane is a quadratic optimization problem with linear constraints. The constrained optimization problem can be stated as follows: Given the training data set \( \{(x_i, y_i), i = 1, 2, ..., n\} \), find the values of \( w \) and \( b \) such that they satisfy the constraints

\[
y_i(w^T v_i + b) \geq 1 \quad \text{for} \quad i = 1, 2, ..., n
\]

and the parameter vector \( w \) minimizes the cost function:

\[
\Psi(w) = \frac{1}{2}w^T w
\]  

The constrained optimization problem may be solved using the method of Lagrange multipliers. The Lagrangian function is given by:

\[
J(w, b, \alpha) = \frac{1}{2}w^T w - \sum_{i=1}^{n} \alpha_i[y_i(w^T v_i + b) - 1]
\]  

where the nonnegative variables \( \alpha_i \) are called Lagrange multipliers. The saddle point of the Lagrangian function provides the solution for the optimization problem. The solution is determined by first minimizing the Lagrangian function with respect to \( w \) and \( b \), and then maximizing with respect to \( \alpha \). The two conditions of optimality due to minimization are:

\[
\frac{\delta J(w, b, \alpha)}{\delta w} = 0
\]

\[
\frac{\delta J(w, b, \alpha)}{\delta b} = 0
\]

Application of optimality conditions to the Lagrangian function gives:

\[
w = \sum_{i=1}^{n} \alpha_i y_i v_i
\]

\[
\sum_{i=1}^{n} \alpha_i y_i = 0
\]
Substituting the expression for \( w \) from Eq.(8) in Eq.(5) and using the condition of Eq.(9), the Lagrangian function can be written as an objective function of Lagrangian multipliers \( \alpha \):

\[
Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j v_i^T v_j
\]  

(10)

The optimum Lagrangian multipliers are determined by maximizing the objective function \( Q(\alpha) \) subject to the constraints:

\[
\sum_{i=1}^{n} \alpha_i y_i = 0
\]  

(11)

\[
\alpha_i \geq 0 \quad for \quad i = 1, 2, \ldots, n
\]  

(12)

This optimization problem is solved using quadratic programming methods. For data points corresponding to the support vectors only, the optimum Lagrange multipliers will take non-zero values. For optimum Lagrange multipliers \( \alpha_{o,i} \), the optimum parameter vector \( w_o \) is given by:

\[
w_o = \sum_{i=1}^{N_s} \alpha_{o,i} y_i v_i
\]  

(13)

where \( N_s \) is the number of support vectors. The optimum bias value, \( b_o \), is computed as follows:

\[
b_o = 1 - w_o^T v_s
\]  

(14)

where \( v_s \) is a positive support vector with \( y_s = +1 \).

Now we consider a method to construct an optimal hyperplane for linearly nonseparable patterns. For training patterns that cannot be separated without classification errors, it would be desirable to find an optimal hyperplane that minimizes the probability of classification error averaged over the training data set. A data point is nonseparable when it does not satisfy the constraint in Eq.(3). This corresponds to a data point that falls within the margin or on the wrong side of the decision boundary as illustrated in Fig.2.

The nonnegative slack variable \( \beta_i \) is introduced into the definition of the separating hyperplane that should satisfy the following constraints for the training data set:

\[
y_i (w^T v_i + b) \geq 1 - \beta_i, \quad for \quad i = 1, 2, \ldots, n
\]  

(15)

The slack variable \( \beta_i \) is a measure of the deviation of a data point \( v_i \) from the ideal condition of separability. For \( 0 \leq \beta_i \leq 1 \), the data point falls inside the region of separation but on the correct side of the separating hyperplane. For \( \beta_i > 1 \), the data point falls on the wrong side of the separating hyperplane. The support vectors are those particular data points that satisfy the constraint in Eq.(15) with equality sign. The cost function for linearly nonseparable patterns is:

\[
\Psi(w) = \frac{1}{2} w^T w + C \sum_{i=1}^{n} \beta_i
\]  

(16)

where \( C \) is a user-specified positive parameter that controls the tradeoff between complexity of the learning machine and the number of nonseparable data points. Using the method of Lagrange multipliers to solve the constrained optimization problem, the objective function can be obtained as follows [5]:

\[
Q(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j v_i^T v_j
\]  

subject to the constraints:

\[
\sum_{i=1}^{n} \alpha_i y_i = 0
\]  

(18)

\[
0 \leq \alpha_i \leq C \quad for \quad i = 1, 2, \ldots, n
\]  

(19)

For optimum Lagrange multipliers \( \alpha_{o,i} \), the optimum parameter vector \( w_o \) is given by:
\[ \mathbf{w}_o = \sum_{i=1}^{N_s} a_{o,i} y_i \mathbf{v}_i \]  
(20)

where \( N_s \) is the number of support vectors. The discriminant function of the optimal hyperplane is defined in terms of support vectors:

\[ D(\mathbf{v}) = \mathbf{w}_o^T \mathbf{v} + b_o = \sum_{i=1}^{N_s} a_{o,i} y_i \mathbf{v}_i^T \mathbf{v} + b_o \]  
(21)

The vectors \( \mathbf{v}_i \) and \( \mathbf{v} \) correspond to the high-dimensional feature vectors obtained from the non-linear transformation \( g \) on the pattern vectors \( \mathbf{x}_i \) and \( \mathbf{x} \) in the input space. Computation of the discriminant function in Eq.(21) involves only the inner product operation \( \mathbf{v}_i^T \mathbf{v} \). Evaluation of inner products in a high-dimensional feature space is avoided by using the inner product kernel, \( K(\mathbf{x}, \mathbf{x}_i) \), defined as follows [6]:

\[ K(\mathbf{x}, \mathbf{x}_i) = g(\mathbf{x}_i)^T g(\mathbf{x}) = \mathbf{v}_i^T \mathbf{v} \]  
(22)

The objective function of Eq.(17) and the discriminant function of the optimal separating hyperplane in Eq.(21) can now be specified using the inner product kernel function of pattern vectors in the input space as follows:

\[ Q(\alpha) = \sum_{i=1}^{n} a_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j y_i y_j K(\mathbf{x}, \mathbf{x}_i) \]  
(23)

\[ D(\mathbf{x}) = \sum_{i=1}^{N_s} a_{o,i} y_i K(\mathbf{x}, \mathbf{x}_i) + b_o \]  
(24)

The architecture of a binary support vector machine for two-class pattern recognition that implements the discriminant function of the hyperplane in Eq.(24) is given in Fig.3. The number of hidden nodes corresponding to the number of support vectors, and the training examples corresponding to the support vectors are determined by maximizing the objective function in Eq.(23) using a given training data set for a chosen kernel function. In the next section, we present the commonly used approaches to multi-class pattern recognition using support vector machines.

3. MULTI-CLASS PATTERN RECOGNITION USING SVMS

Support vector machines are originally designed for two-class pattern classification. Multi-class pattern recognition problems are commonly solved using a combination of binary support vector machines (SVMs) and a decision strategy to decide the class of the input pattern [7]. Each SVM has the architecture given in Fig.3 and is independently trained. Now we present the approaches for decomposition of the learning problem in multi-class pattern recognition into several two-class learning problems so that a combination of SVMs can be used. The training data set \( \{(\mathbf{x}_i, c_i)\} \) consists of \( N \) examples belonging to \( M \) classes. The class label \( c_i \in \{1, 2, \ldots, M\} \). We assume that the number of examples for each class is the same, i.e., \( N/M \).

3.1. One-against-the-rest Approach

In this approach, an SVM is constructed for each class by discriminating that class against the remaining \((M-1)\) classes. The recognition system based on this approach consists of \( M \) SVMs. A
test pattern $x$ is classified by using the winner-takes-all decision strategy, i.e., the class with the maximum value of the discriminant function $D(x)$ is assigned to it. All the $N$ training examples are used in constructing an SVM for a class. The SVM for class $l$ is constructed using the set of training examples and their desired outputs, $\{(x_i, y_i)\}$. The desired output $y_i$ for a training example $x_i$ is defined as follows:
\[
y_i = +1, \quad \text{if } c_i = l
\]
\[
y_i = -1, \quad \text{if } c_i \neq l
\]
The examples with the desired output $y_i = +1$ are called positive examples. The examples with the desired output $y_i = -1$ are called negative examples. An optimal hyperplane is constructed to separate $N/M$ positive examples from $N(M - 1)/M$ negative examples.

3.2. One-against-one Approach

In this approach, an SVM is constructed for every pair of classes by training it to discriminate the two classes. The number of SVMs used in this approach is $M(M - 1)/2$. An SVM for a pair of classes $(l, m)$ is constructed using $2N/M$ training examples belonging to the two classes only. The desired output $y_i$ for a training example $x_i$ is defined as follows:
\[
y_i = +1, \quad \text{if } c_i = l
\]
\[
y_i = -1, \quad \text{if } c_i = m
\]
The maxwins strategy is commonly used to determine the class of a test pattern $x$ in this approach. In this strategy, a majority voting scheme is used. If $D_{lm}(x)$, the value of the discriminant function of the SVM for a pair of classes $(l, m)$, is positive, then class $l$ wins a vote. Otherwise, class $m$ wins a vote. Outputs of SVMs are used to determine the number of votes won by each class. The class with maximum number of votes is assigned to the test pattern. When there are multiple classes with the maximum number of votes, the class with maximum value of the total magnitude of discriminant functions (TMDF) is assigned. The total magnitude of discriminant functions for class $l$ is defined as follows:
\[
TMDF_l = \sum_m |D_{lm}(x)|
\]
where the summation is over all $m$ with which class $l$ is paired.

In the following sections, we study the performance of different approaches for multi-class pattern recognition using SVMs in recognition of subword units of speech. First we study recognition of monophone units in Japanese speech. Then we study recognition of a subset of Consonant-Vowel (CV) units of Indian languages.

4. STUDIES ON RECOGNITION OF MONOPHONE UNITS

In our first study, we use the manually marked segments of monophone units in the large vocabulary continuous speech corpus of Japanese newspaper article sentences [8]. This corpus consists of speech data for recording of a total of 45,000 sentences from 150 male speakers and 148 female speakers. We consider 10 sentences from each speaker for our studies. The number of monophone units is 41. A total of 103,813 monophone segments in the data for 100 male speakers and 100 female speakers are used for training. The test data set includes 41,135 monophone segments in the speech data of the remaining 98 speakers.

For recognition using SVMs, patterns extracted from monophone segments should be of a fixed length. A monophone segment is analyzed frame by frame, with a frame size of 25 milliseconds and a frame shift of 10 milliseconds. Each frame is represented by a parametric vector consisting of 12 mel-frequency cepstral coefficients, energy and their first order derivatives. The fixed length pattern is extracted by dividing a monophone segment into a fixed number of parts of the same duration and then obtaining an averaged frame from the frames in each part [9]. The pattern length $PL$ is equal to the number of parts. We consider pattern lengths of 3 and 5. The duration of a monophone segment is also included in its pattern as an additional parameter.

The performance of systems using HMMs is compared with the performance of Gaussian kernel SVM based systems. A 5-state left-to-right continuous density HMM using 16 Gaussian mixtures with diagonal covariance matrix is trained for each
class. For the SVM based systems, the width of the Gaussian kernel is 70. The SVMs are trained using the one-against-the-rest approach. The performance of the HMM based system and the SVM based systems trained using different values of $PL$ is given in Table 1. The performance is given as the correct classification accuracy on the test data of monophone segments. The average number of support vectors per class is given as $NSV$.

Table 1: Performance of monophone recognition systems built using the one-against-the-rest approach.

<table>
<thead>
<tr>
<th>Models</th>
<th>$PL$</th>
<th>$NSV$</th>
<th>Accuracy (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMMs</td>
<td>—</td>
<td>—</td>
<td>79.44</td>
</tr>
<tr>
<td>SVMs</td>
<td>3</td>
<td>3019</td>
<td>82.92</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3262</td>
<td>84.27</td>
</tr>
</tbody>
</table>

It is seen that the SVM based systems give a better classification accuracy than the HMM based systems. The performance of the one-against-the-rest approach based system is compared with that of the one-against-one approach based system in Table 2. It is seen that both the approaches give approximately the same classification accuracy. However, the number of SVMs in a recognition system and the average number of support vectors per SVM indicate that the complexity for the one-against-the-rest approach based system is significantly less than that of the one-against-one approach based system.

Table 2: Performance of monophone recognition for different approaches to multi-class pattern recognition.

<table>
<thead>
<tr>
<th>Approach</th>
<th>No. of SVMs</th>
<th>$NSV$</th>
<th>Accuracy (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-against-the-rest</td>
<td>41</td>
<td>3262</td>
<td>84.27</td>
</tr>
<tr>
<td>One-against-one</td>
<td>820</td>
<td>303</td>
<td>83.90</td>
</tr>
</tbody>
</table>

In the next section, we focus on recognition of subword units of speech for Indian languages. Since the confusability among the large number of stop consonants is high and the coarticulation between consonants and their following vowels in continuous speech is high [4], we focus on Consonant-Vowel (CV) units of Indian languages.

5. STUDIES ON RECOGNITION OF CONSONANT-VOWEL UNITS

Automatic speech recognition of broadcast news data has become a challenging research topic in recent years [10]. Because of the channel distortion, the broadcast news speech data is noisy and degraded. Discriminative training based approaches are important for acoustic modeling of subword units occurring in noisy and degraded speech data. In this section, we present our studies on recognition of CV segments excised from continuous speech in a broadcast news corpus for Indian languages built at Speech and Vision Laboratory, Indian Institute of Technology, Madras, India. Speech data of television broadcast news in Telugu, a south Indian language, is collected for about five hours over 20 sessions of news reading by 11 male readers and 9 female readers. The speech data is digitized at a sampling frequency of 16 kHz. Boundaries of CV syllables in continuous speech are manually marked. In our studies, the data in 15 sessions of news reading by 8 male speakers and 7 female speakers is used for training. The data in the remaining 5 sessions by 3 male speakers and 2 female speakers is used for testing. The CV units have a varying frequency of occurrence in the database. There are 33 consonants and 10 vowels in Telugu leading to a total of 330 CV units. We consider 86 CV classes that have a frequency of occurrence greater than 100 in our studies. The training data set has about 35,450 segments belonging to these 86 CV classes. The frequency of occurrence for these classes in the training data varies in the range of 105 to 1892. The test data includes about 12,800 CV segments belonging to the 86 CV classes.

The method used for extraction of fixed length patterns from segments of monophone units cannot be extended for CV units. It is mainly because the characteristics of the acoustic signal vary significantly in the consonant and vowel regions.
Table 3: List of 86 Consonant-Vowel (CV) classes. Short vowels are denoted by lower case letters and long vowels are denoted by upper case letters.

<table>
<thead>
<tr>
<th>Category of consonant</th>
<th>CV classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stop consonants</td>
<td>ka, kA, ki, ku, kO, khA</td>
</tr>
<tr>
<td></td>
<td>ga, gA, gi, gu</td>
</tr>
<tr>
<td></td>
<td>tA, tA, i, ti, tu</td>
</tr>
<tr>
<td></td>
<td>da, dA, di, du</td>
</tr>
<tr>
<td></td>
<td>ta, tA, ti, tu, tU, te, tO</td>
</tr>
<tr>
<td></td>
<td>da, dA, di, du, dE, dhA, dh</td>
</tr>
<tr>
<td></td>
<td>pa, pA, pi, pu, pO</td>
</tr>
<tr>
<td></td>
<td>ba, bA, bha, bhA</td>
</tr>
<tr>
<td>Affricates</td>
<td>ca, cA, ci, cE, ja, jA, ju</td>
</tr>
<tr>
<td>Nasals</td>
<td>na, nA, ni, nu, nE</td>
</tr>
<tr>
<td>Semivowels</td>
<td>ya, yA, yi, yu</td>
</tr>
<tr>
<td></td>
<td>ra, rA, ri, ru, rO</td>
</tr>
<tr>
<td></td>
<td>la, lA, li, lu, lE, lO</td>
</tr>
<tr>
<td></td>
<td>wa, wA, wi, wE</td>
</tr>
<tr>
<td>Fricatives</td>
<td>shA, sa, sA, si, su, ha, hA</td>
</tr>
</tbody>
</table>

The method for extraction of fixed length patterns from CV segments is as follows: A CV segment is analyzed frame by frame, with a frame size of 25 milliseconds and a frame shift of 10 milliseconds. The length of a segment, SL, is the number of frames in it. For a chosen pattern length PL specified as the number of frames, the fixed length patterns are obtained by linear compaction or elongation of CV segments. If the segment length SL is greater than PL, a few frames of the segment are omitted. If the segment length SL is smaller than PL, a few frames of the segment are repeated. The linear relationship between the index s of a frame in the CV segment and the index p of a frame in the CV pattern is as follows:

\[ s = \frac{p \times SL}{PL} \]  

(28)

The length of the CV segment is also included in the pattern as an additional parameter. We study the performance of CV recognition systems for different values of PL.

The performance of 86-class CV recognition systems using HMMs is compared with the performance of Gaussian kernel SVM based systems. A 5-state left-to-right continuous density HMM using multiple mixtures with diagonal covariance matrix is trained for each class. The number of mixtures is 4 for the CV classes with a frequency of occurrence less than 500 in the training data. For the other classes, the number of mixtures is 8. For the SVM based systems, the width of the Gaussian kernel is 50 and the trade-off parameter C is 100. The SVMs are trained using the one-against-the-rest approach. We consider the pattern lengths of 10 and 14 frames. The length of 10 frames corresponds to the average length of all the CV segments in the training data. The length of 10 frames corresponds to the maximum of the average segment length for 86 CV classes. The 86-class CV recognition performance of the HMM based system and the SVM based systems is given in Table 4. The SVM based system gives the higher accuracy when PL is 10.

Table 4: Performance of CV recognition systems built using the one-against-the-rest approach.

<table>
<thead>
<tr>
<th>Models</th>
<th>PL</th>
<th>NSV</th>
<th>Accuracy (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMMs</td>
<td>10</td>
<td>2472</td>
<td>62.00</td>
</tr>
<tr>
<td>SVMs</td>
<td>14</td>
<td>3707</td>
<td>61.64</td>
</tr>
</tbody>
</table>

The performance of the one-against-the-rest approach based system is compared with that of the one-against-one approach based system in Table 5. The accuracy of the one-against-one approach based system is less by about 3.5% compared to the one-against-the-rest approach based system. Complexity of the recognition system is also significantly less for the one-against-the-rest approach based system.

6. SUMMARY AND CONCLUSIONS

In this paper, we have studied the performance of different classification models for recognition of subword units of speech. The results of our studies show that the SVM based classifiers give a better performance compared to the classifiers based on
Table 5: Performance of CV recognition for different approaches to multi-class pattern recognition.

<table>
<thead>
<tr>
<th>Approach</th>
<th>No. of SVMs</th>
<th>NSV</th>
<th>Accuracy (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-against-the-rest</td>
<td>86</td>
<td>2472</td>
<td>62.00</td>
</tr>
<tr>
<td>One-against-one</td>
<td>3655</td>
<td>372</td>
<td>58.40</td>
</tr>
</tbody>
</table>

hidden Markov models. In our studies, we have used the inner product kernel functions for pattern vectors of a fixed dimension. The loss of information in extraction of fixed length patterns from the varying duration segments of units can be avoided by using the dynamic alignment kernels [11] or score-space kernels [12]. Hybrid architectures based on a combination of support vector machines and HMMs [9] can be explored for continuous speech recognition.

Acknowledgment

Authors thank Prof.B.Yegnamarayana, Indian Institute of Technology, Madras, India, for making the broadcast news data available for our studies.

REFERENCES


