Automatic Recognition of Speech Patterns of Numeric Digits Using Support Vector Machines: A New Approach

G. Batista¹, and W. Silva²

¹Student of Electrical Engineering, Federal Institute of Maranhao, Sao Luis, Maranhao, Brazil
²Department of Electrical and Electronics, Federal Institute of Maranhao, Sao Luis, Maranhao, Brazil

Abstract—This paper proposes the implementation of a Support Vector Machine (SVM) for automatic recognition of numerical speech commands. Besides the pre-processing of the speech signal with mel-cepstral coefficients, is used to Discrete Cosine Transform (DCT) to generate a two-dimensional matrix used as input to SVM algorithm for generating the pattern of words to be recognized. The Support Vector Machines represent a new approach to pattern classification. SVM is used to recognize speech patterns from the mean and variance of the speech signal input through the two-dimensional array aforementioned, the algorithm trains and tests those data showing the best response. Finally it shows the experimental results in speech recognition applied to Brazilian Portuguese language process.

Keywords: Support Vector Machines; Classification; Pattern Recognition; Statistical Learning Theory; Application in Speech Recognition.

1. Introduction

1.1 Digital Processing of the Speech Signal

Digital speech processing is a specialty in full expansion. There are numerous applications of this research area, we can refer to automatic speech recognition for purposes of interpretation of commands by machines or robots, automatic speech recognition for the purpose of biometric authentication, recognition of pathology in the mechanism of speech production for biometric and or medicinal purposes. The speech processing systems are divided basically into three sub-areas: speech coding, speech synthesis and speech automatic recognition. Regardless of the specific purpose, the initial stages of a system for processing digital speech is sampling followed by segmentation of words or phonemes [1] for short-term analysis by Fourier transform or by spectral analysis. The speech signal processing first involves obtaining a parametric representation based on a certain model and then applying a transformation to represent the signal in a more convenient form for recognition. The last step in the process is the extraction of important characteristics for a given application. This step can be performed either by human listeners or automatically by machines [14]. Among the techniques that have been developed for segmentation of speech, those based on Hidden Markov Models (HMM) are quite traditional. Hybrid methods based on artificial neural networks and criteria such as average energy, selection of voiced phonemes and non voiced, Mel Frequency Cepstral Coefficients (MFCCs), spectral metrics, and others, are also used. Speech coding systems include those cases in which the purpose is to obtain a parametric representation of the speech signal, based on the analysis of the frequency, average power and other characteristics of the spectrum of the signals. The techniques of encoding the speech signal are used both for transmission and for compact storage of speech signals. One of the main applications of speech coding is to transmit the speech signal efficiently [2]. Systems for automatic speech recognition or Speech Recognition Systems (SRS) are focused on the recognition of the human voice by intelligent machines.

1.2 Methodology Proposed

This article uses as a recognition default locations from Brazilian Portuguese of the digits 0′, 1′, 2′, 3′, 4′, 5′, 6′, 7′, 8′, 9′. The speech signal is sampled and encoded in mel-cepstral coefficients and coefficients of Discrete Cosine Transform (DCT) in order to parameterize the signal with a reduced number of parameters. Then, it generates two dimensional matrices referring to the mean and variance of each digit. The elements of these matrices representing two-dimensional temporal patterns will be used to classify by machines (Support Vector Machine).

Fig. 1: Flowchart Blocks of Training System.

1.3 Pre-processing of Speech Signal

Initially, the speech signal is sample and segmented into frames, after segmentation of the speech signal passing through a process of windowing and it is encoded in a set of mel-cepstral parameters. The number of parameters obtained is determined by the order of mel-cepstral coefficients. The
obtained coefficients are then encoded by Discrete Cosine Transform (DCT) in a two dimensional matrix that will represent the speech signal that will be recognized. The process of windowing in a given signal, aims to select a small portion of this signal, which will be analysed, named frame. A short-term Fourier analysis performed on these frames is called signal analysis frame by frame. The length of the frame \( T_f \) is defined as the length of time upon which a parameter set is valid. The term frame is used to determine the length of time between successive calculations of parameters. For speech processing, normally, the time frame is between 10ms and 30ms [13].

1.4 Generation of two-dimensional DCT-temporal matrix

After being properly parameterized in mel-cepstral coefficients, the signal is encoded by DCT performed in a sequence of observation vectors of mel-cepstral coefficients on the time axis. Thus, a two-dimensional temporal array DCT is generated for each \( m (m=1,2,3,...,10 \) number of samples to generate each pattern) example of model \( P \), represented by \( CV^m \). Finally, arrays of mean \( CM^j_{kn} \) (1) and variance \( CV^j_{kn} \) (2) are generated. The parameters of \( CM^j_{kn} \) and \( CV^j_{kn} \) are used as data of input in SVM algorithm.

\[
CM^j_{kn} = \frac{1}{M} \sum_{m=0}^{M-1} C^j_{km} \tag{1}
\]

\[
CV^j_{kn} = \frac{1}{M-1} \sum_{m=0}^{M-1} C^j_{kn} - \left( \frac{1}{M} \sum_{m=0}^{M-1} C^j_{km} \right)^2 \tag{2}
\]

where \( k, 1 \leq k \leq K \), refers to the \( k \)-th line (number of Mel frequency cepstral coefficients) of \( t \)-th segment of the matrix, \( n, 1 \leq n \leq N \) component refers to the \( n \)-th column (order of DCT) and \( j=0,1,2,...,9 \) is the number of patterns to be recognized.

1.5 Generation of machines

In the technical literature about SVMs, the standards are called classes. The mean and variance matrices are transformed in two column vectors, \( CMe \) (vector with means) and \( CVar \) (vector with variances).

\[
CMe^j = \left\{ CM^j_{11}, CM^j_{12},...,CM^j_{K1},CM^j_{11},CM^j_{22},...,CM^j_{KN},\ldots,CM^j_{KN} \right\} \tag{3}
\]

\[
CVar^j = \left\{ CV^j_{11}, CV^j_{12},...,CV^j_{K1},CV^j_{22},...,CV^j_{KN} \right\} \tag{4}
\]

For example, in the case of a matrix \( CM^j_{22} \), that is, where \( K=2 \) e \( N=2 \), the matrices \( CMe \) and \( CVar \) take the following form:

\[
CMe^j = \{ CM^j_{11},CM^j_{12},CM^j_{21},CM^j_{22} \} \tag{5}
\]

\[
CVar^j = \{ CV^j_{11},CV^j_{12},CV^j_{21},CV^j_{22} \} \tag{6}
\]

Each class in this example is represented by 4 elements in the vector of mean and 4 elements in vector of variance according to (5) and (6), that is, the first 4 elements of the vector of mean and of the vector of variance refer into class 0, the following 4 elements of each vector to the class 1, and so on. Figure 2 shows data of the peers of mean and variance of the speech signals from the examples of (5) and (6). The set of functions mapping of type input-output is given by Equation 7:

\[
\Omega = f\left( [CMe^j; CVar^j], w \right) \tag{7}
\]

where \( \Omega \) is the real response produced by the learning machine associated with the entry of pairs of means and variances, and \( w \) is a set of free parameters, called weights for weighting, selected from the parameter space related to patterns. Figure 3 shows a general model of the supervised learning from the examples, having three components:

The Environment is the fixed input system, this yields \( x_i \) (points that come from the pairs of coordinates \( (CMe, CVar) \)) from the response of the DCT matrix of speech signals. The Supervisor returns a value of the desired output \( d_i \) for each input vector \( x_i \) in accordance with a conditional distribution function \( F (d_i | x_i) \), also set. Machine of Learning (ML), or algorithm capable of implementing a set of functions \( f\left( [CMe^j; CVar^j], w \right), \) where \( w \in W \), where \( W \) is a set of parameters belonging to the set of desired
responses. In this context, the learning problem can be interpreted as a problem of approximation, which involves finding a function $f \left( CMe^{\epsilon_1}; CV_{r_1} \right)$ that generates the best approximation to the $\Omega$ output of the supervisor. The selection is based on a set of independent training examples $I$ and identically distributed ($iid$), generated according to:

$$ F(x, d) = F(x)F(d|x) : (x_i, d_i) $$

(8)

where $(x_i, d_i)$ are peers with desired input and output with $d_i \in R^n$ and $i = 1, ..., I$.

### 1.5.1 SVM (Support Vector Machine)

Based on Statistical Learning Theory, a Support Vector Machine was developed by Vapnik [3], in order to solve problems of pattern classification, from studies initiated in the work “On the uniform convergence of relative frequencies of their probabilities to events” [5]. The Theory of Statistical Learning aims to establish mathematical conditions that allow the selection of a classifier with good performance for the data set available for training and testing. In other words this theory seeks to find a good classifier with good generalization regarding the entire data set. The desired performance of a classifier $f$ is that it gets the smallest mistake during training, with the error being measured by the number of incorrect predictions of $f$. Therefore it’s defined as Empirical Risk $R_{emp}(f)$ the extent of loss between the desired response and the actual response and restrictions on Risk Functional use the concept of VC dimension [5]. Theory of uniform convergence of functional of empirical risk to functional of real risk includes limits on the rate of convergence, which are based on an important parameter called the Vapnick-Chervonenkis dimension, or simply VC dimension, named after its creators, Vapnik and Chervonenkis. The VC dimension is a measure of the capacity or power of expression of the family of classification functions performed by the learning machine [9]. Haykin [6] furthers more details about the Functional of Risk and VC dimension.

SVM is a classifier that separate linearly the data through a hyperplane. And to determine the optimal hyperplane separability, as it was assumed that the training set is linearly separable. The separating hyperplane the follows equation of a decision surface below:

$$ \omega^T x + b = 0 $$

(9)

where $x$ is an input vector, $\omega$ is a vector of adjustable weight (maximum separation possible between true and false examples) and $b$ is a bias [16].

For the case of a non-linear set, the SVM’s create another feature space from the original space, and the concepts and calculations of linear optimal hyperplane are applied in this new space [6].

The SVM is a dichotomic algorithm, that is, for pattern classification based on two classes [6]. However, it is possible to obtain a classifier for multiple classes using the SVM algorithm. Scholkopf et al. the proposed classifier model of type “one vs. all” [10]. Clarkson and Brown have proposed classifier model of the “one vs. one” [4]. However, both models are indeed classifiers of only two classes: Class +1 and Class -1 [6]. On system “one vs. all” it is used one machine for each group, in which each group is trained separately from the rest of the set. In the system “one vs. one” it is used only three machines, in which a group is classified against another, then this one is rated against another group and so on, until the whole set is trained.

The decision surface of the SVM, which in the feature space is always linear, usually is nonlinear in the input space. As seen earlier, the idea of Support Vector Machine depends on two mathematical operations:

1) Nonlinear mapping of an input vector into a feature space of high dimensionality, which is hidden from the entry and exit [16];

2) It’s necessary to build an optimal hyperplane to separate the features discovered in the first step. To design the optimal hyperplane it is needed a kernel function, or core of the inner product. A Kernel function is a function that receives two points of the input space and calculates the scalar product of the data in the feature space [16].

To ensure the convexity of the optimization problem and introduce the Kernel mapping in which the calculation of scalar products is possible, you must use a kernel function that follows the conditions set by Mercer’s Theorem [11], [12]. In general, the three most important of the kernel functions are Polynomial, RBF kernel and Perceptron (MLP) [6].

### 2. Experimental Results

After performing the pre-processing of the speech signal coding and generation of temporal matrices $CM_{k_n}$ and $CV_{l_n}$, the models were trained by SVM machines $CM_{2_2}$ and $CV_{2_2}$, that is, $K=2$ and $N=2$ as shown in Figure 4, for $CM_{3_3}$ and $CV_{3_3}$, that is, $K=3$ and $N=3$ as shown in Figure 5, and $CM_{4_4}$ e $CV_{4_4}$, i.e., $K=4$ e $N=4$ as shown in Fig 6.

With the result of the best machines from training, the tests were made from voice banks where the speakers
are independent and classified with the best machines of training. The speakers 1 and 2 are male and the speaker 3 is female. The Tables 1, 2 and 3 show the rates of successes. The Figures 4, 5 and 6 are from the best results that were generated by RBF function of sigma 0.03. Besides the best results, good results also were obtained from Polynomial of order 2, as show the tables 4, 5 and 6. To improve the tests results, training was made from 20 examples of each pattern.

The generated hyperplane during classification with RBF funtion with sigma 0.03 is very small. This is because as smaller the sigma, smaller the coverage area of the hyperplane is. This explain why it is impossible to observe the data that were classified as true in its classes in the Figures 4, 5 and 6.

![Figure 4: Machine generated for class 4 from matrices CM\(_{422}^4\) and CV\(_{422}^4\).](image1)

![Figure 5: Machine generated for class 4 from matrices CM\(_{333}^4\) and CV\(_{333}^4\).](image2)

![Figure 6: Machine generated for class 4 from matrices CM\(_{444}^4\) and CV\(_{444}^4\).](image3)

<p>| Table 1: Test performed from matrices CM(<em>{222}^j) and CV(</em>{222}^j) and RBF of sigma 0.03 |
|---------------------------------|-----------------|-----------------|-----------------|</p>
<table>
<thead>
<tr>
<th>Machines</th>
<th>Training</th>
<th>Class 0</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
<th>Class 5</th>
<th>Class 6</th>
<th>Class 7</th>
<th>Class 8</th>
<th>Class 9</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speaker 1</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>76</td>
</tr>
<tr>
<td>Speaker 2</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>70</td>
</tr>
<tr>
<td>Speaker 3</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>74</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>160</strong></td>
<td><strong>76</strong></td>
<td><strong>70</strong></td>
<td><strong>74</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| Table 2: Test performed from matrices CM(<em>{333}^j) and CV(</em>{333}^j) and RBF of sigma 0.03 |
|---------------------------------|-----------------|-----------------|-----------------|</p>
<table>
<thead>
<tr>
<th>Machines</th>
<th>Training</th>
<th>Class 0</th>
<th>Class 1</th>
<th>Class 2</th>
<th>Class 3</th>
<th>Class 4</th>
<th>Class 5</th>
<th>Class 6</th>
<th>Class 7</th>
<th>Class 8</th>
<th>Class 9</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speaker 1</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>76</td>
</tr>
<tr>
<td>Speaker 2</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>70</td>
</tr>
<tr>
<td>Speaker 3</td>
<td>16</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>6</td>
<td>8</td>
<td>8</td>
<td>74</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>164</strong></td>
<td><strong>80</strong></td>
<td><strong>78</strong></td>
<td><strong>76</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 3. Conclusions

Analysing the methodology and applications of SVM, one realises that it is a technique with excellent response time of computational execution. Despite being a dichotomic method of classification, this also has possible means to work with a larger number of classes of different data types to be separated. In the standards classification proposed in this work, the SVM presented problems to correctly classify points very close among each other, because of the form of generalization one versus all. However, as it has a very wide scope in relation to the classification functions during learning process of the machines, the SVM ends up compensating the problem of generalization with the use of more points for classification. That is, the greater the number of points to represent the class the higher the amount of hits. Regarding the recognition of patterns in general were well classified, except with the digit ‘9’. The digits ‘1’ and ‘8’ obtained the highest classifications. And between Polynomial and RBF functions, the second one presented the best results. The use of mean and variance chosen as characteristics of the data to be generated patterns was the most appropriate way to find a better separability between points and therefore a better classification.

### References

2. A. A. Bresolin, Reconhecimento de voz através de unidades menores do que a palavra, utilizando Wavelet Packet e SVM, in uma nova Estrutura Hierarquica de Decisao, UFRN, 2008.


