Classification of Pattern using Support Vector Machines: An Application for Automatic Speech Recognition

Gracieth Batista*, Washington Silva† and Orlando Filho†
*Student of Electrical Engineering
Federal Institute of Maranhão, São Luis, Maranhão, Brazil 65053-155
Email: gracieth.cavalcanti@hotmail.com
†Department of Electrical and Electronics
Federal Institute of Maranhão
Email: washington.wlss@ifma.edu.br and orlando.rocha@ifma.edu.br

***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-ceptral coefficients. Also, this paper 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, the experimental results are presented for the speech recognition applied to Brazilian Portuguese language process.**

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

I. INTRODUCTION

A. 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 automatic speech 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 [2] or by spectral analysis [2]. 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 [2]. Among the techniques that have been developed for segmentation of speech, those based on Hidden Markov Models (HMM) [2] are quite traditional. Hybrid methods based on Artificial Neural Networks (ANN) [3] and criteria, such as average energy, selection of voiced phonemes and non voiced, Mel Frequency Cepstral Coefficients (MFCCs) [2], spectral metrics [2], 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 [4]. Systems for automatic speech recognition or Speech Recognition Systems (SRS) are focused on the recognition of the human voice by intelligent machines.

B. Methodology Proposed

This article uses as a recognition default locutions from Brazillian 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) [2] 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 classified by Support Vector Machines (SVMs) [3]. The innovation of this work is in the reduced number of parameters lies in the SVM classifier and in the reduction of computational load caused by this reduction of parameters.

C. SVM (Support Vector Machine)

Based on Statistical Learning Theory, SVM classifier is another category of feed-forward, whose outputs of neurons from a layer feed neurons from the next layer where feedback doesn't occur [3]. This technique originally developed for binary classification, seeks to build hyperplanes as decision surfaces, in such a way so that the separation between classes is maximum, assuming that the patterns are linearly separable. As for non-linearly separable patterns, the SVM seeks an appropriate mapping function to make the mapped set linearly separable. Due to its efficiency in working with high-dimensional data, it is cited in the literature as a highly robust technique [5]. The results of applying this technique are comparable and often superior to those obtained by other learning algorithms, such as ANN.

 ISBN: 978-1-61208-354-4

1) Theory of Statistical Learning: 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. But, this classifier abstains from particular cases, which defines the capability to correctly predict the class of new data from the same domain in which the learning occured. **Machines Learning (ML)** [3] employs an inference principle called induction, in which general conclusions are obtained from a particular set of examples. A model of supervised learning based on Theory of Statistical Learning is given in Figure 1 [3].

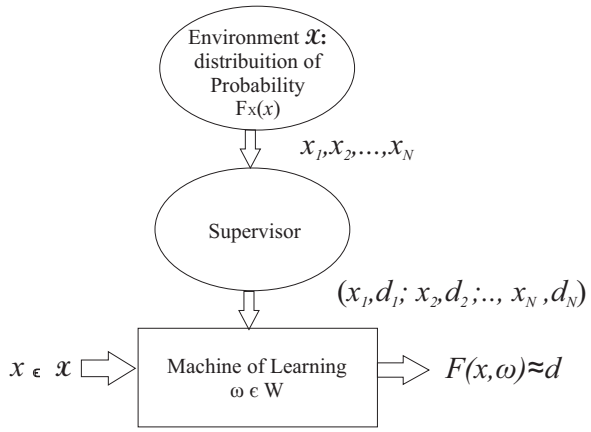

Figure 1. Flowchart of a model of supervised learning.

Environment: It is stationary. It provides an input vector x with a function of distribution of cumulative probability fixed, but unknown $F_x(x)$.

Supervisor: It shows a desired response d for each input vector x is provided by the environment accordance to a conditional cumulative distribution function $F_x(x|d)$ which is also fixed but unknown. The desired response due to input vector x is related by (1):

$$d = f(x, v) \tag{1}$$

where v is noise that allows the supervisor to be noisy. The kind of learning discussed in this work is supervised, but not noisy.

2) Functional of Risk: 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, its defined as Empirical Risk $Remp(f)$ the extent of loss between the desired response and the actual response. In (2), it is shown the definition of the Empirical Risk.

$$Remp(w) = \frac{1}{N}\sum_{i=1}^{N} c(f_i(x_i, y_i)) \tag{2}$$

where $y_i = F(x_i, w_i)$, w_i is a vector of adjustable weights, c is the cost function related to the prediction $f(x_i)$, with desired output $f(y_i)$, where one type of cost function is the "loss 0/1" defined by (3). The process of search by an equation $f(x)$ that represents a smaller value of $Remp(f)$ is called Empirical Risk Minimization.

$$c(f_i(x_i, y_i) = \begin{cases} 1, & \text{if} \quad y_i f(x_i) < 0 \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Assuming that the patterns used for training (x_i, y_i) are generated by an independent and identically distributed distribution (iid) of probability $P(x, y)$. The probability of incorrect Classification from classifier f is called Functional Risk, which quantifies the capability of generalization, according to (4) [6].

$$R(f) = \int c(f(x_i, y_i))dP(x_i, y_i) \tag{4}$$

During the training process, $Remp(f)$ can be easily obtained, while $R(f)$ cannot, since probability P is unknown. Given a set of training data (xi, yi) with $x_i \in \Re^N$ and $y_i \in \{\pm 1\}, i = 1, 2, ..., n$, i = 1, 2, ..., n, the input vector x_i and y_i is the output related to class x_i, then the goal is to estimate a function $f : \Re^N \rightarrow \{\pm 1\}$ and if no restriction is imposed on the class of functions in which one chooses to estimate f, it may happen that the function obtains a good performance in the training set, but not having the same performance in unknown patterns. This phenomenon is called the error "*overfitting*". Thus, the minimization of the empirical risk does not guarantee a good generalization capability, and being a great classifier is desired f^* such that $R(f^*) = min_{f \in F} R(f)$, where F is the set of possible functions f . The Theory of Statistical Learning provides ways to limit the class of functions (hyperplanes), in order to exclude bad models, that is, those leading to the error of overfitting, implementing a function with an adequate capacity to correctly classify the set of training data. Restrictions on Risk Functional use the concept of VC dimension [7].

3) SVM (Mathematical Modeling): Classifiers that separate the data through a hyperplane are called Linear and SVM fits this definition, therefore, we must pay attention to all that there is to train and classify, for as a SVM must also deal with non-linearly separable sets, this will resort to techniques. In the application of Techniques of Statistics Learning (TSL), the classifier must be chosen the classifier with the lowest possible empirical risk and which also satisfies the constraint of belonging to a family F with a small VC dimension. Also, to determine the separability of the optimal hyperplane, as it was assumed that the training set is linearly separable. The equation of a decision surface folows below:

$$\omega^T x + b = 0 \tag{5}$$

where x is an input vector, ω is a vector of adjustable weight (maximum separation possible between true and false examples) and b is a *bias*. And from this consideration follows a sequence of calculations in order to find the hyperplane with higher separability between classes. Under these conditions, the surface found is called optimal. In Figure 2, the geometry of an optimal hyperplane for two-dimensional space is illustrated.

For the case of a non-linear set, SVM creates another feature space from the original space, and the concepts and

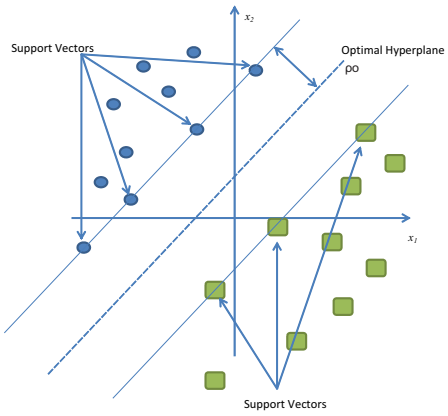

Figure 2. Optimal hyperplane for linearly separable patterns.

calculations of linear optimal hyperplane are applied in this new space [3].

4) SVM for multiple classes: The SVM is a dichotomic algorithm, that is, for pattern classfication based on two classes [3]. However, it is possible to obtain a classifier for multiple classes using the SVM algorithm. Scholkopf et al. proposed a classifier model of type "one vs. all" [8]. Clarkson and Brown have proposed a classifier model of the "one vs. one" [9]. However, both models are indeed classifiers of only two classes: Class +1 and Class -1 [3]. On system "one vs. all", one machine for each group is used, in which each group is trained separately from the rest of the set. In the system "one vs. one", only three machines are used, 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.

5) Functions Kernel: 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 SVM 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;
2) Build an optimal hyperplane to separate the features discovered in the first step. To design the optimal hyperplane, a kernel function is needed, or a core of the inner product. A Kernel function is a function that receives two points x_i and x_j of the input space and calculates the scalar product of the data in the feature space, given by (6).

$$k(x_i, x_j) = \Phi^T(x_i) \cdot \Phi(x_j) \quad (6)$$

To ensure the convexity of the optimization problem and introduce the Kernel mapping in which the calculation of scalar products is possible, a kernel function that follows the conditions set by Mercers Theorem [10][11]must be used. The kernels that satisfy Mercers conditions are characterized for giving origin to semi-definite positive matrices k, in which each element k_{ij} is defined by $k_{ij} = k(x_i, x_j)$, $\forall i, j = 1, 2, ..., n$. Once the mapping is performed by a SVM kernel function, and not directly by $\Phi(x)$, it is not always possible to know exactly which mapping is actually performed, because the kernel functions perform an implicit mapping. Table I shows the main features commonly used as kernel functions. The expansion of the inner product core $K(x_i, x_j)$ in (6) makes it possible to find a decision surface that is non-linear in the input space, but whose image in the feature space is linear [3].

TABLE I. APPLICATION OF SVM

Name of Kernel	Function
Polinomial	$\left(x^T x_i + 1\right)^p$
RBF Kernel	$exp\left(-\frac{1}{2\sigma^2}\|x - x_i\|^2\right)$
Perceptron	$tanh\left(\beta_0 x^T x_i + \beta_1\right)$

The Kernel functions used have the following restrictions:

- In the Polynomial kernel, the parameter p is first specified by the user;
- In the kernel RBF, the parameter σ^2 is common to all cores;
- In the perceptron, Mercer's theorem is satisfied only for some values of β_0, β_1;

6) Automatic Systems for Speech Recognition with SVM: Hidden Markov Models (HMMs) have become the most employed technique for Automatic Speech Recognition (ASR). However, the HMM-based ASR systems may reach their limit of performance. Hybrid systems based on a combination of artificial intelligence techniques provide significant improvements of performance. However, the progress in this paradigm has been hindered by their training computational requireents, which were excessive when these systems were proposed. Recently, several methods of Speech Recognition have been proposed using mel-frequency cepstral coefficients and Neural Networks Classifiers [12][13][14], Sparse Systems for Speech Recognition [15], Hybrid Robust Voice Activity Detection System [16], Wolof Speech Recognition with Limited vocabulary Based HMM and Toolkit [17], Real-Time Robust Speech Recognition using Compact Support Vector Machines [6].

Thus, the SVM has many functions; it is a binary algorithm, based in the Theory of Statistical Learning and in the Functional of Risk. And, finally, it has many functions for classification, such as in the case of multiple classes.

II. SYSTEM OF SPEECH RECOGNITION

A. Pre-processing of Speech Signal

Initially, after the segmentation of the speech is passed through the process of windowing, the speech signal is sampled and segmented into frames and is encoded in a set of melcepstral 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) [2] in a two dimensional matrix that will represent the speech signal that to be recognized. The process of windowing in a given signal, aims to select a small portion of this signal, which will be analysed and 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

 ISBN: 978-1-61208-354-4

valid. The term frame is used to determine the length of time between successive calculations of parameters. Normally, for speech processing, the time frame is between 10ms and 30ms [18].

B. 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 T observation vectors of mel-cepstral coefficients on the time axis. The coding by DCT is given by the equation following:

$$C_k(n,T) = \frac{1}{N}\sum_{t=1}^{T} mfcc_k(t) cos\frac{(2t+1)n\pi}{2T} \quad (7)$$

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), $mfcc_k(t)$ represents the mel-cepstral coefficients. Thus, one obtains the two-dimensional matrix that encode the long term variations of the spectral envelope of the speech signal [19]. This procedure is performed for each spoken word. Thus, there is a two-dimensional matrix $C_k(n,T) \equiv C_{kn}$ for each input signal. The matrix elements are obtained as the following:

1) For a given model of spoken word ***P*** (digit), ten examples of this model are pronounced. Each example is properly divided into T frames distributed along the time axis. Thus, we have: $P_0^0, P_1^0, ..., P_9^0, P_0^1, P_1^1 ..., P_9^1, P_0^2, P_1^2, ..., P_9^2$,..., P_m^j, where j=0,1,2,...,9 is the number of patterns to be recognized and m=1,2,3,...,10, is the number of samples to generate each pattern.
2) Each frame of a given example of model ***P*** generates a total of K mel-frequency cepstral coefficients, and then, significant characteristics are obtained within each frame over this time. The DCT of order N is then calculated for each mel-cepstral coefficient of the same order within the frame, that is, c_1 in the frame t_1, c_1 in the frame $t_2, ...,$, c_1 in the frame t_T, c_2 in the frame t_1, c_2 in the frame $t_2, ...,$, c_2 in the frame t_T, and so on, generating elements $\{c_{11}, c_{12}, c_{13}, ..., c_{1N}\}$, $\{c_{21}, c_{22}, c_{23}, ..., c_{2N}\}$, $\{c_{K1}, c_{K2}, c_{K3}, ..., c_{KN}\}$ in the matrix given in (7). Thus, a two-dimensional temporal array DCT is generated for each m example of model ***P***, represented by C_{kn}^{jm}. Finally, arrays of mean CM_{kn}^j (8) e variance CV_{kn}^j (9) are generated. The parameters of CM_{kn}^j and CV_{kn}^j are used as datas of input in SVM algorithm.

$$CM_{kn}^j = \frac{1}{M}\sum_{m=0}^{M-1} C_{kn}^{jm} \quad (8)$$

$$CV_{kn}^j = \frac{1}{M-1}\sum_{m=0}^{M-1}\left[C_{kn}^{jm} - \left(\frac{1}{M}\sum_{m=0}^{M-1} C_{kn}^{jm}\right)\right]^2 \quad (9)$$

C. 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).

$$CMe_i^j = \langle CM_{11}^0, CM_{12}^0, ..., CM_{1N}^0, CM_{21}^0, CM_{22}^0, ..., CM_{2N}^0, CM_{KN}^J \rangle \quad (10)$$

$$CVar_i^j = \langle CV_{11}^0, CV_{12}^0, ..., CV_{1N}^0, CV_{21}^0, CV_{22}^0, ..., CV_{2N}^0, CV_{KN}^J \rangle \quad (11)$$

For example, in the case of a matrix CM_{22}^j, that is, where K=2 e N=2, the matrices CMe and $CVar$ take the following form:

$$CMe_i^j = \langle CM_{11}^0, CM_{12}^0, CM_{21}^0, CM_{22}^0, CM_{11}^1, CM_{12}^1, CM_{21}^1, CM_{22}^1, ..., CM_{22}^J \rangle \quad (12)$$

$$CVar_i^j = \langle CV_{11}^0, CV_{12}^0, CV_{21}^0, CV_{22}^0, CV_{11}^1, CV_{12}^1, CV_{21}^1, CV_{22}^1, ..., CV_{22}^J \rangle \quad (13)$$

Each class in this example is represented by 4 elements in the vector of mean and 4 elements in vector of variance according to (12) and (13), 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 3 shows data of the peers of mean and variance of the speech signals from the examples of (12) and (13).

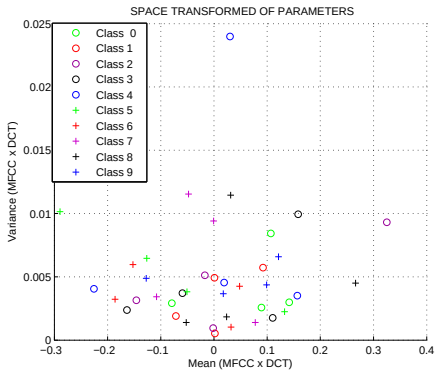

Figure 3. Classes and their different points.

The set of functions mapping of type input-output is given by (14):

$$\Omega = f\Big([CMe_i^j; CVar_i^j], w\Big) \quad (14)$$

where Ω is the real response produced by the learning machine associated with the entry 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 4 shows a general model of the supervised learning from the examples, having three components:

 ISBN: 978-1-61208-354-4

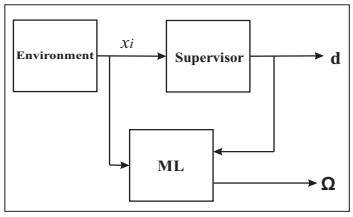

Figure 4. Model of Learning.

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)**, is an algorithm capable of implementing a set of functions $f\left([CMe_i^j; CVar_i^j], w\right)$, where $\omega \in W$, 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 $f\left([CMe_i^j; CVar_i^j], w\right)$ that generates the best approximation to the Ω output of the supervisor. The selection is based on a set of independent training examples I and identically distributed (iid), generated according to:

$$F(x, d) = F(x)F(d|x) : (x_i, d_i) \quad (15)$$

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

III. Experimental Results

A. Training

After performing the pre-processing of the speech signal coding and generation of temporal matrices CM_{kn}^j and CV_{kn}^j, the models were trained by SVM machines CM_{22}^j and CV_{22}^j, that is, *K*=2 and *N*=2, as shown in Figure 5, for CM_{33}^j and CV_{33}^j, that is, *K*=3 and *N*=3, as shown in Figure 6 , and CM_{44}^j e CV_{44}^j,i.e., *K*=4 e *N*=4, as shown in Figure 7. The best results for matrices with *K*=2, *N*=2 , *K*=3 and *N*=3 were generated by polynomial function of order 3. However, the best results for matrices with *K*=4 e *N*=4 were generated by *Kernelcachelimit* function, because as each class is represented by 16 points and there are 10 classes to be classified (separated), there are 160 points separated and the *Polynomial* function obeys an order P as shown in Table I and $1 \leq P \leq 3$, $P \in Z$ resulting in a very limited hyperplane relative to the curvature that the function line can make with a limit P equal to 3. The *Kernelcachelimit* function provides a value that specifies the size of the cache memory of the kernel matrix, while the algorithm maintains a matrix with up to 5000×5000 of double precision floating-point numbers in memory.

In Bresolin [20], the use of SVM with wavelet digital voice recognition in Brazilian Portuguese, obtained an average of 97.76% using 26 MFCC's in the pre-processing of voice and SVM machine's with the following characteristics: MLP as Kernel functions, ten machines (one for each class) and "one vs. all" as method of multiple classes. In comparison to this work, the results of this remain more effective, because the amount of MFCC's is smaller and, also, the input of parameters in the machines are lower. Consequently, the computational load is lower.

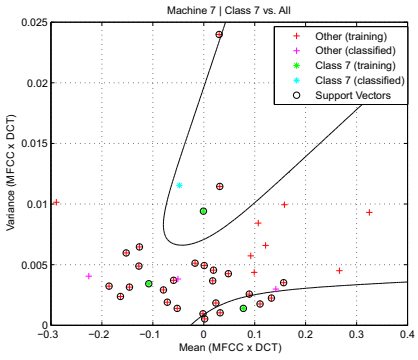

Figure 5. Machine generated for class 7 from matrices CM_{22}^7 and CV_{22}^7.

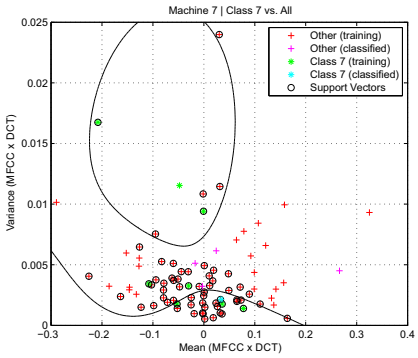

Figure 6. Machine generated for class 7 from matrices CM_{33}^7 and CV_{33}^7.

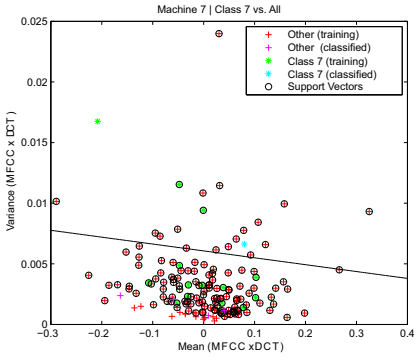

Figure 7. Machine generated for class 7 from matrices CM_{44}^7 and CV_{44}^7.

B. Test

With the result of the best function from training, the tests were made from voice banks where the speakers are independent and classified with the best function of training: *Polynomial* of order 3, except the matrices with *K*=4 e *N*=4 were tested (classified) with the same function of the training: *Kernelcachelimit*. The speakers 1 and 2 are male and the speaker 3 is female. The Tables II, III and IV show the rates of successes.

 ISBN: 978-1-61208-354-4

TABLE II. TEST PERFORMED FROM MATRICES CM_{22}^{j} AND CV_{22}^{j}

Machines	Training	Test		
		Speaker 1	Speaker 2	Speaker 3
Class 0	10	10	10	10
Class 1	10	10	7	10
Class 2	8	5	7	5
Class 3	8	5	5	5
Class 4	10	5	5	5
Class 5	10	3	5	7
Class 6	8	5	7	5
Class 7	10	7	5	5
Class 8	10	10	10	10
Class 9	10	3	0	0
TOTAL	94	63	61	62

TABLE III. TEST PERFORMED FROM MATRICES CM_{33}^{j} AND CV_{33}^{j}

Machines	Training	Test		
		Speaker 1	Speaker 2	Speaker 3
Class 0	10	9	8	9
Class 1	10	10	9	10
Class 2	8	8	7	7
Class 3	10	9	7	10
Class 4	10	3	4	2
Class 5	10	3	4	3
Class 6	10	6	7	7
Class 7	8	9	8	9
Class 8	10	9	10	7
Class 9	10	6	7	7
TOTAL	96	72	71	71

TABLE IV. TEST PERFORMED FROM MATRICES CM_{44}^{j} AND CV_{44}^{j}

Machines	Training	Test		
		Speaker 1	Speaker 2	Speaker 3
Class 0	8	8	8	8
Class 1	10	10	10	10
Class 2	10	8	9	8
Class 3	10	9	7	8
Class 4	10	6	7	8
Class 5	10	8	9	7
Class 6	8	8	7	8
Class 7	10	10	10	10
Class 8	10	10	10	10
Class 9	10	6	6	5
TOTAL	96	82	83	82

IV. CONCLUSION

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 to each other, because of the form generalization of one versus all. However, as it has a very wide scope in relation to the classification functions during the learning process of the machines, the SVM ends up compensating for 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. In general, the patterns were classified very well, except with the digit '9'. The digits '1' and '8' obtained the highest classifications. 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.

ACKNOWLEDGMENT

The authors thank the Scientific Initiation Program of the Federal Institute of Maranhao for financial support through grant aid, and availability of Digital Systems Laboratory, and the Research Group for Electronic Instrumentation Technology Applied to the IFMA.

REFERENCES

[1] P. Fantinato, Segmentacao de Voz baseada na Analise Fractal e na Transformada Wavelet. Prentice Hall, Outubro 2008.

[2] L. Rabiner and R. Schafer, Digital Processing of Speech Signals. Prentice Hall, 1978.

[3] S. Haykin, Redes Neurais:Principio e pratica. Bookman, 2002.

[4] A. Bresolin, Reconhecimento de voz atraves de unidades menores do que a palavra, utilizando Wavelet Packet e SVM, em uma nova Estrutura Hierarquica de Decisao. Tese de Doutorado, Natal 2008.

[5] C. Ding and I. Dubchak, Multi-class protein fold recognition using support vector machines and neural networks. Bioinformatics, 2001.

[6] R. Urena, A. Moral, C. Moreno, M. Ramon, and F. Maria, "Real-time robust automatic speech recognition using compact support vector machines." IEEE Transactions on Audio, Speech, and Language Processing, May 2012, pp. 1347–1362.

[7] V. Vapnik and A. Chervonenkis, On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. Dokl, 1968.

[8] B. Scholkopf, O. Simard, A. Smola, and V. Vapnik, Prior knowledge in support vector kernels. The MIT Press, 1998.

[9] P. C. Clarkson and P. Moreno, "Acoustics, speech and signal processing." IEEE International Conference, March 1999, pp. 585–588.

[10] J. Mercer, Functions of positive and negative type, and their connections with theory of integral equations. Transactions of the London Philosophical Society, 1909.

[11] C. De-Gang, Y. Heng, and E. Tsang, Generalized Mercer theorem and its application to feature space related to indefinite kernels. International Conference Machine Learning and Cybernetics, 2008.

[12] D. Hanchate, M. Nalawade, M. Pawar, V. Pohale, and P. Maurya, "vocal digit recognition using artificial neural network"." 2nd International Conference on Coumputer Engineering and Technology, April 2010, pp. 88–91.

[13] R. Aggarwal and M. Dave, "application of genetically optimized neural networks for hindi speech recognition system"." World Congress on Information and Communication Technologies (WICT), December 2011, pp. 512–517.

[14] S. Azam, Z. Mansor, M. Mughal, and S. Moshin, "urdu spoken digits recognition using classifield mfcc and backpropagation neural network"." 4th International Conference on Computer Graphics, Imaging and Visualization (CGIV), August 2007, pp. 414–418.

[15] M. Mohammed, E. Bijov, C. Xavier, A. Yasif, and V. Supriya, "robust automatic speech recognition systems:hmm vesus sparse"." Third International Conference on Intelligent Systems modelling and Simulation, February 2012, pp. 339–342.

[16] C. Ganesh, H. Kumar, and P. Vanathi, "performance analysis of hybrid robust automatic speech recognition system"." IEEE International Conference on Signal Processing, Computing and Control (ISPCC), March 2012, pp. 1–4.

[17] J. Tamgo, E. Barnard, C. Lishou, and M. Richome, "wolof speech recognition model of digits and limited-vocabulary based on hmm and toolkit"." 14th International Conference on Computer Modelling and Simulation (UKSim), March 2012, pp. 389–395.

[18] J. Picone, "Signal modeling techniques in speech recognition." IEEE Transactions on Computer, April 1991, pp. 1215–1247.

[19] P. Fissore and E. Rivera, "Using word temporal structure in hmm speech recongnition." ICASSP 97, April 1997, pp. 975–978.

[20] A. Brasolin, A. Neto, and P. Alsin, ""digit recognition using wavelet and svm in brazilian portuguese"." ICASSP 2008, April 2008, pp. 1–4.

 ISBN: 978-1-61208-354-4