Automatic Speech Recognition Using Support Vector Machine and Particle Swarm Optimization

Gracieth Cavalcanti Batista
Federal Institute of Maranhao
Student of Electrical Engineering
Sao Luis, Maranhao, Brazil
Email: gracieth.cavalcanti@gmail.com

Washington Luis Santos Silva
Federal Institute of Maranhao
Department of Electrical and Electronics
Sao Luis, Maranhao, Brazil
Email: washington.wlss@ifma.edu.br

Angelo Garangau Menezes
Tiradentes University
Student of Mechatronics Engineering
Aracaju, SE, Brazil
Email: agaranga@lakeheadu.ca

***Abstract*—Support Vector Machine (SVM) is an algorithm that trains and classifies different types of data through of an optimal hyperplane of decision. On the other hand, Particle Swarm Optimization (PSO) is, in general, an algorithm that finds the best point to represent a dataset. In this paper, PSO is used to find the best data of each class (pattern) to be trained by SVM and there is a comparison of the difference between using or not this optimization. The digits of zero to nine in Brazilian Portuguese language are recognized automatically by SVM. Those digits are pre-processed using mel-cepstral coefficients and Discrete Cosine Transform (DCT) to generate a two-dimensional matrix used as input to the PSO algorithm for generating the optimal data.**

***Index Terms*—Support Vector Machines, Particle Swarm Optimization, Pattern Recognition, Statistical Learning Theory, Automatic Speech Recognition.**

1. Introduction

Techniques of pre-processing signals based on Hidden Markov models (HMM) have been treated as conventional within the domain of those developed for speech segmentation. Hybrid techniques that take into consideration Mel Frequency Cespstral Coefficients (MFCCSs), selection of voiced phonemes and non voiced, artificial neural networks are also applied to the same operation. 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 signals spectrum. 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 [1]. The speech processing usually takes much time and computational load and in order to minimize these characteristics, optimization algorithms can be applied efficiently as PSO, in this case. Due to its high performance and flexibility, PSO has become a great option to optimization applications. The PSO technique was developed based on the social behavior of flocking birds and schooling fish when searching for food [7]. The PSO technique simulates the behavior of individuals in a group to maximize the species survival. Each particle "flies" in a direction that is based on its experience and that of the whole group. Individual particles move stochastically toward the position affected by the present velocity, previous best performance, and the best previous performance of the group [8]. The main purpose of PSO algorithm is to find a solution function (optimal solution) that finds the best data positions and in this case, also reduces the quantity of data to be trained without loss of information. The Support Vector Machine (SVM) was initially developed by Cortes and Vapnik [4], and its concept involves subjects such as calculus, vector geometry and Lagrange multipliers. In details, SVM is based on the Statistical Learning Theory that provides a classifier ability of generalizing the data set as good as possible, in order to find the best response of separating through training and testing [14]. In brief, the innovation in this paper is the use of PSO algorithm (optimization technique) to reduce the processing time and computational load during the training.

2. Methodology Proposed

This article uses as a recognition default locutions 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 the Discrete Cosine Transform (DCT) is applied in order to parametrize the signal with a reduced number of parameters. Those coefficients are organized in a two-dimensional matrix and introduced on the PSO algorithm. The PSO algorithm finds the optimal data to represent the speech signals and those data representing the two-dimensional temporal patterns will be used in the classification by machines (Support Vector Machine).

978-1-5090-4240-1/16/$31.00 ©2016 IEEE

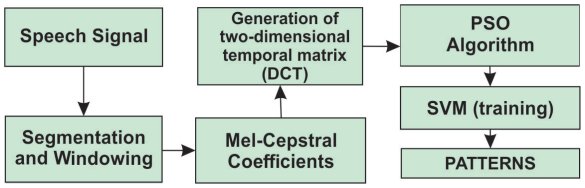

Figure 1. Flowchart Blocks of Training System.

3. Pre-processing of Speech Signal

Firstly, the voice signals are divided in small parts reorganized into frames (with a time frame between 10ms and 30ms). Secondly, a process of windowing is used and this data (properly windowed) is transformed in mel-cepstral parameters [10]. The quantity of parameters is set by the order of mel-cepstral coefficients. Then, the Discrete Cosine Transform (DCT) is applied to the coefficients and a two dimensional matrix is generated to be recognized.

3.1. 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. Then, a array is originated from the application of the DCT for each m (m=1,2,3,...,20 number of samples to generate each pattern) example of model ***P***, represented by C_{kn}^{jm}, 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. Finally, C_{kn}^{jm} is reorganized in a matrix $CT_{i\times 2}$, where $i = (j \times K \times N \times m)/2$.

4. Particle Swarm Optimization (PSO)

As described by the inventers James Kennedy and Russell Eberhart, particle swarm algorithm imitates human (or insects) social behaviour. Individuals interact with one another while learning from their own experience, and gradually the population members move into better regions of the problem space. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles [16]. With the two-dimensional DCT-temporal matrix, the PSO algorithm was applied to find the best pair of points to represent each class (pattern). Equation 1 is the mathematical function that finds the best solution for the optimization [7].

$$g(\alpha, \beta) = (\alpha - a)^2 + (\beta - b)^2 \quad (1)$$

where α and β form pairs of columns originated of the matrix $CT_{i\times 2}$ that it will be transformed in only one α and β for each class at the end of the PSO program processing, a and b are the points which are localized at the center of each class, a in relation to x axis and b in relation to y axis. Finally, the g function will find the best pairs of points to represent the classes, a matrix of 2 columns and 10 lines, where each line represents each spoken digit (each class).

5. Generation of the machines - SVM (Support Vector Machine)

The set of functions mapping of type input-output is given by 2:

$$\Omega = f([CT_{10\times 2}], w) \quad (2)$$

where Ω is the real response produced by the learning machine associated with the entry of pairs of observation vectors of mel-cepstral coefficients on the time axis (matrix from the output of the PSO algorithm), and w is a set of free parameters, called weights for weighting, selected from the parameter space related to patterns.

The function f is the classifier that finds the best response for the training with the smallest error possible, then this error is obtained from the number of incorrect predictions of the classifier f. The Empirical Risk $Remp(f)$ is the average loss of an estimator for a finite set of data and the VC dimension [14] is used on the Risk Functional to calculate that $Remp(f)$ function. Vapnik and Chervonenkis are the creators of the VC dimension concept that measures the capacity of classification by the learning machine [6]. The Risk Functional and VC dimension are more detailed in "Redes Neurais:Principio e Pratica" [13]. SVM is a dichotomic method that finds the best optimal hyperplane to separate two classes (patterns) of each other, and it obeys the following equation:

$$\omega^T x + b = 0 \quad (3)$$

where x is an input vector, ω is a where x is an input vector, is a vector of adjustable weight (maximum separation possible between true and false examples) and b is a bias [2].

As in the most of cases, in this approach, the data set is non-linear, then, SVM creates a different feature space in order to linearize as much as possible so that the optimal hyperplane could be developed in this new space [13]. Techniques for multi-classes classification are required to do this kind of classification since it is known that SVM is a dichotomic algorithm. "One vs. all" and "one vs. one" are the techniques created by Scholkopf et al. [3], Clarkson and Brown [12], respectively. During the first one ("one vs. all" technique), one group is trained against the rest of the data set (two or more groups together) until the best solution with the minimum rate is found. On the other hand, the second one ("one vs. one" technique), is about training one group against the other, and then the other against another one until there is no more groups (data) left to be trained. In brief, SVM works in two general steps:

1) Transformation of the non-linear data set (input space) into a linear data set (feature space) that can be trained normally by the algorithm [11];
2) The building of the optimal hyperplane that it is used the Kernel function concept where the hyperplane is originated from calculations of scalar products [11]. The kernel function follows determinations developed by Mercers Theorem [9], [5]. Polynomial, Radial Basis Function (RBF) and Perceptron (MLP) are the most commonly used kernel functions with the most promising results in most applications of the SVM algorithm [13].

6. Experimental Results

6.1. Recognition using SVM and PSO

After performing the pre-processing of the speech signal coding and generation of temporal matrix and application of the PSO algorithm, the models were trained by SVM machines with *K*=2 and *N*=2, with *K*=3 and *N*=3 and with *K*=4 and *N*=4. All voice banks of training were composed by 20 samples for each class (pattern), that is, $m = 1, 2, ..., 20$ in the array DCT (C_{kn}^{jm}). With the result of the machines from training, the tests were made from voice banks where the speakers are independent and classified with the machines of training. The speakers 1 and 2 are male and the speakers 3 and 4 are female. The table 1 shows the rates of success for *K*=2 and *N*=2, the table 2 shows the rates of success for *K*=3 and *N*=3, and the table 3 shows the rates of success for *K*=4 and *N*=4. The best results were generated by *RBF* function of sigma 0.03. The generated hyperplane during classification with *RBF* function with sigma 0.03 is very small. This is because as smaller the sigma, smaller the coverage area of the hyperplane is. This explains why the results from *K*=2 and *N*=2 and results from *K*=3 and *N*=3 are very similar.

6.2. Recognition using only SVM

The recognition with SVM without using PSO was made to compare the results with and without the PSO application. This recognition was made using the same techniques, in other words, the same pre-processing of speech signal, the same training and testing, and the same respective voice banks. Also, the speakers 1 and 2 are male and the speakers 3 and 4 are female. However, in this application, it was necessary to apply the same training 100 times, because it was necessary to find the best machine of each class to be used during testing. In addition, the input of the training algorithm was bigger than in the first recognition (using PSO) because it was used all the data from the pre-processing, that is, it was taken more time on this recognition than on the other one and the input matrix CT that had 10 lines and two columns, now, it has i lines and 2 columns.

Then, after performing the pre-processing of the speech signal coding and generation of the temporal matrix $CT_{i\times 2}$, the models were trained by SVM machines with *K*=2 and *N*=2 , with *K*=3 and *N*=3 , and with *K*=4 and *N*=4 . All voice banks of training were composed by 20 samples for each class (pattern), that is, $m = 1, 2, ..., 20$ in the array DCT (C_{kn}^{jm}).

The table 4 shows the rates of success for *K*=2 and *N*=2, the table5 shows the rates of success for *K*=3 and *N*=3, and the table 6 show the rates of success for *K*=4 and *N*=4. The best results were generated by *RBF* function of sigma 0.03 as well. To improve the tests results, training was made from 20 examples of each pattern and the tests were made from 10 examples of each pattern

6.3. Comparison between the process with optimization and without optimization

The Figure 4 is a block diagram that shows the process without the using of PSO algorithm which is shorter but it took more time to be finished and the Figure 5 is another block diagram that shows the other process, with PSO algorithm, which is larger but more efficient in relation to the results and faster than without the PSO application. The Figure 3 shows the training result of the $Class4$ for the process of recognition using the optimization, where the numbers of data points were reduced to matrices of 10 rows by 2 columns. In contrast, the Figure 2 shows the training result of the $Class4$ for the process of recognition without the optimization, where the numbers of data points were matrices of n rows by 2 columns. The time of process using SVM and PSO was 68.0481 seconds and on other hand, the processing time using just SVM (without optimization) was 1,760.231329 seconds.

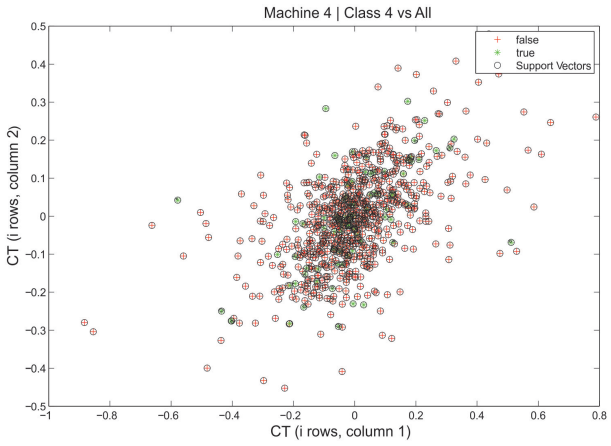

Figure 2. Machine generated for class 4 with *K*=3 and *N*=3 without the Optimization.

TABLE 1. TEST PERFORMED FROM MATRICES WITH K=2 AND N=2 AND *RBF* OF SIGMA 0.03, USING PSO.

Machines	Training	Test			
		Speaker 1	Speaker 2	Speaker 3	Speaker 4
Class 0	100%	9	9	9	9
Class 1	100%	9	9	9	9
Class 2	100%	9	9	9	8
Class 3	100%	9	9	9	9
Class 4	100%	9	9	9	8
Class 5	100%	9	9	9	9
Class 6	100%	9	9	9	9
Class 7	100%	9	9	9	9
Class 8	100%	9	9	9	10
Class 9	100%	9	9	9	9
TOTAL	100%	90	90	90	89

TABLE 2. TEST PERFORMED FROM MATRICES WITH K=3 AND N=3 AND *RBF* OF SIGMA 0.03, USING PSO.

Machines	Training	Test			
		Speaker 1	Speaker 2	Speaker 3	Speaker 4
Class 0	100%	10	10	10	10
Class 1	100%	10	10	10	10
Class 2	100%	10	9	9	9
Class 3	100%	10	10	10	9
Class 4	100%	10	9	9	9
Class 5	100%	10	9	10	10
Class 6	100%	10	9	10	9
Class 7	100%	9	9	9	10
Class 8	100%	9	9	9	10
Class 9	100%	9	9	9	9
TOTAL	100%	98	93	95	95

TABLE 3. TEST PERFORMED FROM MATRICES WITH K=4 AND N=4 AND *RBF* OF SIGMA 0.03, USING PSO.

Machines	Training	Test			
		Speaker 1	Speaker 2	Speaker 3	Speaker 4
Class 0	100%	10	10	10	10
Class 1	100%	9	10	9	9
Class 2	100%	9	10	10	10
Class 3	100%	10	10	9	9
Class 4	100%	10	10	9	9
Class 5	100%	10	10	10	10
Class 6	100%	10	9	10	9
Class 7	100%	10	9	10	10
Class 8	100%	10	10	9	9
Class 9	100%	10	10	10	10
TOTAL	100%	98	98	96	95

TABLE 4. TEST PERFORMED FROM MATRICES WITH K=2 AND N=2 AND *RBF* OF SIGMA 0.03, WITHOUT OPTIMIZATION.

Machines	Training	Test			
		Speaker 1	Speaker 2	Speaker 3	Speaker 4
Class 0	90%	9	9	9	9
Class 1	90%	9	8	9	9
Class 2	90%	8	9	9	8
Class 3	90%	9	9	9	9
Class 4	90%	9	9	8	8
Class 5	90%	9	9	9	9
Class 6	90%	8	9	8	9
Class 7	90%	9	9	8	9
Class 8	90%	9	9	9	10
Class 9	90%	9	9	9	9
TOTAL	90%	87	89	87	89

TABLE 5. TEST PERFORMED FROM MATRICES WITH K=3 AND N=3 AND *RBF* OF SIGMA 0.03, WITHOUT OPTIMIZATION.

Machines	Training	Test			
		Speaker 1	Speaker 2	Speaker 3	Speaker 4
Class 0	100%	9	9	9	9
Class 1	100%	8	9	9	9
Class 2	100%	9	10	10	9
Class 3	80%	8	8	9	9
Class 4	100%	8	9	8	9
Class 5	100%	9	8	8	9
Class 6	100%	9	9	9	8
Class 7	100%	9	9	8	9
Class 8	100%	9	9	9	9
Class 9	100%	9	9	9	9
TOTAL	98%	87	89	88	89

TABLE 6. TEST PERFORMED FROM MATRICES WITH K=4 AND N=4 AND *RBF* OF SIGMA 0.03, WITHOUT OPTIMIZATION.

Machines	Training	Test			
		Speaker 1	Speaker 2	Speaker 3	Speaker 4
Class 0	100%	10	10	8	8
Class 1	100%	10	10	10	8
Class 2	100%	10	10	10	8
Class 3	100%	9	10	8	6
Class 4	100%	10	10	10	10
Class 5	100%	10	10	10	10
Class 6	100%	10	8	10	10
Class 7	100%	10	10	10	10
Class 8	100%	10	10	10	10
Class 9	100%	10	10	6	10
TOTAL	100%	99	98	92	90

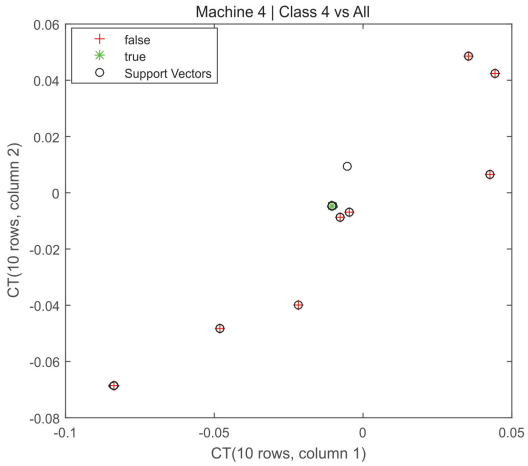

Figure 3. Machine generated for class 4 with K=3 and N=3 using the Optimization by PSO.

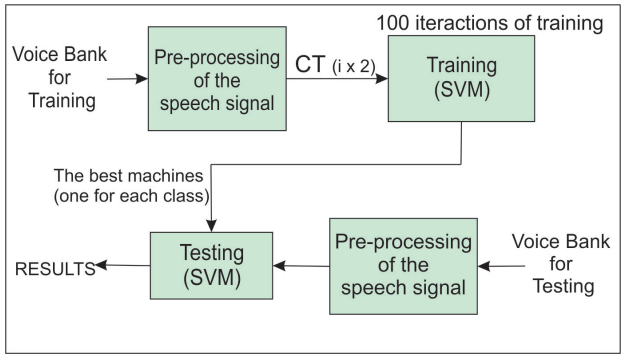

Figure 4. Recognition using SVM without optimization

7. Conclusion

Analyzing the methodology and applications of Through the application of SVM and PSO in this approach and analyzing other papers, it is possible to agree that SVM is a very promising technique because its algorithm is fast in relation to its response. Besides, it is a flexible technique that can be applied to various types of data set. The only problem found was about the proximity among the data location on the feature space (generalization problem), how close the points were between each other, it was more difficult to find the best classification. However, by modificating the parameters of kernel function used during the process and also, by using the "one versus all" technique, it was possible to solve this problem. In addition, it was noticed that SVM works finner with a larger data set to be classified; which means, how larger the number of points is, the recognition result becomes better. The patterns 1 and 8 were classified with the best obtained results from the use of RBF function with $Sigma = 0.03$. In relation to the using of the PSO algorithm, it was outstanding that the optimization was a great differential because the obtained results were better and the whole process was faster and simpler.

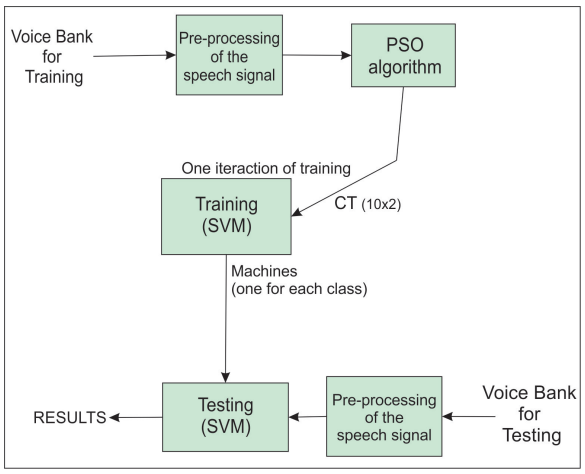

Figure 5. Recognition using SVM with optimization

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] A. 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*, UFRN,2008.

[2] A. Sloin and D. Burshtein, *Support Vector Machine Training for Improved Hidden Markov Modeling*, IEEE Trans. Signal Processing, vol.56, pp.172-188, 2008.

[3] B. Scholkopf, O. Simard, A. Smola and V. Vapnik, *Prior knowledge in support vector kernels.*, The MIT Press, Vol 2, 1999.

[4] C. Cortes and V. Vapnik, *Machine Learning*, pp. 273-297,1995.

[5] C. De-Gang, Y.W. Heng and E.C.C. Tsang, *Generalized Mercer theorem and its application to feature space related to indefinite kernels*, IEEE Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 12-15 July 2008, 2008.

[6] C. J. C. Burges, *A tutorial on Support Vector Machines for Pattern Recognition.*, Kluwer Academic Publishers, Boston, 1998.

[7] J. Kennedy and R. Eberhart, *Particle swarm optimization*, Proceedings of IEEE, Vol.4, pp.1942-1948, 1995.

[8] J. Kennedy and R. Eberhart, *Swarm Intelligence* , Morgan Kaufmann, San Francisco, 2001.

[9] J. Mercer, *Functions of positive and negative type, and their connections with theory of integral equations*, Proceedings of the Royal Society of London - Philosophical Transactions of the Royal Society, Royal Society of London, November 3, 1909.

[10] J.W. Picone, *Signal Modeling Techniques in Speech Recognition*, IEEE Transactions on Computer, Vol 81, 9th edition, Apr. 1993, pp. 1215-1247, doi: 10.1109/5.237532.

[11] K. K. Chin., *Support Vector Machines Applied To Speech Pattern Classification*, Masters Thesis, University of Cambridge, 1998

[12] P. Clarkson and P. J. Moreno, *Acoustics, Speech and Signal Processing.*, IEE Internacional Conference, Vol 2, 1999.

[13] S. Haykin, *Redes Neurais:Principio e Pratica*, Bookman, 2002.

[14] V. N. Vapnik and A. Y. Chervonenkis, *On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities*, Dokl, 2ed, 1968.

[15] V. N. Vapnik, *The nature of statical learning theory.*, Spring-Verlag, 2ed, 2000.

[16] X. Hu, R. Eberhart, and Y. Shi., *Recent advances in particle swarm*, IEEE Congress on Evolutionary Computation, Portland, Oregon, USA, 2004.