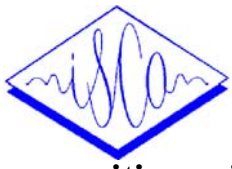

SPECOM'2004: 9th Conference
Speech and Computer
St. Petersburg, Russia
September 20-22, 2004

Noisy speech recognition using string kernels

J. Goddard (1), A. E. Martínez (1), F. M. Martínez (1), H. L. Rufiner (2)

(1) Department of Electrical Engineering, Universidad Autonoma Metropolitana, Iztapalapa, Mexico
jgc@xanum.uam.mx
(2) Cybernetics Laboratory, Engineering Faculty, National University Entre Rios, Argentina

Abstract

In the last few years, Support Vector Machine classifiers have been shown to give results comparable, or better, than Hidden Markov Models for a variety of tasks involving variable length sequential data. This type of data arises naturally in the fields of bioinformatics, text categorization and automatic speech recognition. In particular, in a previous work it was shown that certain string kernels gave a classification performance comparable to discrete Hidden Markov Models on an isolated Spanish digit recognition task.

It is known that speech recognition degrades, often quite severely, when noise is present, and it is interesting to ask whether Support Vector Machines with string kernels continue to give a similar proficiency to discrete Hidden Markov Models in this context. In the present paper, this question is explored by considering the performance of Support Vector Machines with string kernels on the same isolated Spanish digit recognition task in which the speech data has been corrupted with different types of noise. Specifically, white noise and speech babble from the NOISEX-92 database. Results of these experiments are given.

1. Introduction

Automatic Speech Recognition (ASR) systems try to recognize, at some level, human speech. One essential feature of these systems is their ability to classify phonetic units, such as phonemes, syllables or even complete words. This is an extremely difficult task in itself, and more so given that there may be additional complicating factors such as background noise.

The method of choice for classification in ASR is based on the Hidden Markov Model (HMM). Each of the phonetic units is modeled with an HMM by processing the corresponding speech signal into a sequence of fixed dimensional vectors, often using mel frequency cepstral coefficients (mfcc). This gives rise to a non-discriminative type of learning and classification, as only examples of the particular class are used to define the HMM. Deficiencies have been noted with this framework and researchers have studied other classifiers, most commonly combining HMMs in various hybrid schemes, for example with artificial neural nets.

A different type of classifier, called Support Vector Machines (SVM) was introduced in the early 1990's by Vapnik [1] and since then they have been applied to a wide variety of classification problems with excellent results, usually outperforming other techniques. Their success has to do principally with their generalization ability, nevertheless they also provide an attractive discriminative approach to classification problems through the use of kernels.

The first kernels used in SVM classifiers, were the radial basis function and polynomial kernels. These kernels are defined for problems with static data, where the patterns are represented by a single vector of a fixed dimension rather than a sequence of vectors. Researchers in ASR began to employ SVM classifiers for different speech recognition tasks at the end of the 1990's with the work of [2,3,4,5]. These papers either combined the SVM classifier within an HMM scheme or manipulated the sequences corresponding to phonetic units, using an averaging process, into a single vector with a fixed dimension. More recently a straightforward approach has produced competitive results, as is shown in the work of [6,7,8], and performs classification by utilizing the individual vectors of the phonetic units with the above mentioned 'static' kernels.

However speech is dynamic in nature, with different instances of the same phonetic unit producing different length sequences of vectors. This is one of the strengths of HMMs in as much as they can successfully handle sequential data. It is interesting to ask whether kernels can be defined directly on variable length sequences and produce similar classification results. Additionally, it would also allow any two sequences to be compared for similarity using the kernel.

The first attempt at defining kernels of this kind was motivated by research in bioinformatics and can be found in [9]. The so-called Fisher kernel was developed and has become widely used in ASR as an alternative to HMMs (c.f.[10]). Similarly, kernels applied to discrete sequential data, rather than sequences of vectors, have also been defined in [11,12]. They have produced good results in bioinformatics [13], text categorization [14] and ASR [15].

In the latter work, certain string kernels were adopted and compared to discrete HMMs (dHMM) on an isolated Spanish digit recognition task. Their classification performance was comparable to dHMMs. This is encouraging as it provides a conceptually simple and discriminative alternative to dHMMs, which works directly on the variable-length sequential data. This

formulation may also have applications in exemplar-based phonology as outlined in [16].

It is known that speech recognition degrades, often quite severely, when noise is present (c.f. [17]), and it is interesting to ask whether SVMs with string kernels continue to give a similar proficiency to dHMMs in this context. In the present paper, this question is explored by considering the performance of SVMs with string kernels on the same isolated Spanish digit recognition task in which the speech data has been corrupted with different types of noise. Specifically, white noise and speech babble from the NOISEX-92 database. The noise was added to the speech data at different signal to noise ratios (SNR).

The organization of the paper is as follows: a brief review of SVM classifiers and string kernels is presented, followed by a description of the data and results obtained using the classifiers. Finally some conclusions are given.

2. Svm classifiers

The techniques employed in the paper are based on SVM classifiers with string kernels and dHMMs. While dHMMs are well-known in the speech community, SVM classifiers, and in particular string kernels, have been introduced more recently and a brief review will be given here. For a detailed introduction to SVM classifiers the interested reader can consult [18], or find a good tutorial in [19].

The basic idea behind SVMs, for a two class classification problem, is to map the data in each class into linearly separable sets in a higher dimensional inner product vector space, called the feature space. A separating hyperplane is then found in feature space which maximizes the minimal distance, known as the margin, between the hyperplane and the closest points of the classes to it. New patterns are then classified according to the side of the hyperplane they are mapped to.

The mathematical formulation for this is the following: consider a dataset $X = \{x_1, x_2, \ldots, x_N\}$ in $\Re^n$, n-dimensional Euclidean space, each point of which belongs to one of two classes C_1, C_2 with associated labels $y_1, y_2 \in \{-1,+1\}$. When the vectors in the two classes are linearly separable, that is, there is a hyperplane which separates the two classes, then there are different algorithms which can find a separating hyperplane, such as the Perceptron algorithm. In general any hyperplane, P, is defined by:

$$P = \{z \in \Re^n \mid \langle w,z\rangle + b = 0\} \quad (1)$$

where $w = (w_1,\ldots,w_n) \in \Re^n$ is the normal vector to the hyperplane P, $b \in \Re$ and $\langle w,z\rangle$ is the inner product in $\Re^n$. The fact that P separates the two classes means that the additional condition holds:

$$y_i(\langle w,x_i\rangle + b) > 0 \ \forall i \quad (2)$$

In this case the corresponding classifier is given by:

$$f(x) = \text{sign}(\langle w,x\rangle + b) \quad (3)$$

A typical separating hyperplane, P, is shown in Figure 1. In the case of a SVM classifier, a special separating hyperplane is chosen which maximizes its distance to the patterns of X which are closest. The idea being that such a hyperplane produces the smallest generalization error. In Figure 1 this corresponds to P1. The perpendicular distance between the separating hyperplane and any which pass through the closest patterns (H1, H2 in Figure 1) is called the margin, and the hyperplane is called the maximal margin hyperplane. It is shown in [20] that the margin is half the shortest distance between the convex hulls formed using the patterns in each class. Hence certain patterns

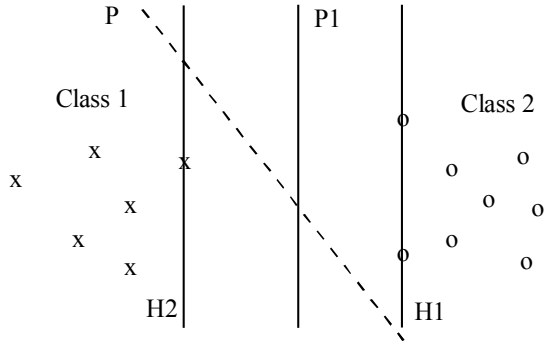

Figure 1: Example separating hyperplanes

on the boundaries of the convex hulls define the classifier.

The task of finding the maximal margin hyperplane can be formulated as the following optimization problem. Firstly, Eq (2) is normalized such that the following holds for the closest patterns in X:

$$y_i(\langle w,x_i\rangle + b) = 1 \quad (4)$$

This means that the margin is $1/\|w\|$, where $\|w\| = \sqrt{\langle w,w\rangle}$. Therefore in order to maximize the margin the following problem has to be solved:

$$\text{Min } \frac{1}{2}\|w\|^2 \quad (5)$$
$$\text{subject to: } y_i(\langle w,x_i\rangle + b) \geq 1 \ \forall i$$

Points satisfying (4) are called support vectors.

As will be seen below, it is convenient to reformulate problem (5). To this end non-negative Lagrangian multipliers, $\alpha_1, \alpha_2, \ldots, \alpha_N$ are introduced for each restriction, and (5) changes to minimizing the following Lagrangian, L, with respect to w, b:

$$L = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N}\alpha_i y_i(\langle w,x_i\rangle + b) + \sum_{i=1}^{N}\alpha_i \quad (6)$$

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
J. Goddard, F. M. Martínez, A. E. Martínez & H. L. Rufiner; "Noisy speech recognition using string kernels"
Proc. of the 9th International Conference Speech and Computer (SPECOM 2004), pp. 89-94, sep, 2004.

Taking partial derivatives with respect to *w*, *b* and equaling to zero gives:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{7}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{8}$$

Substituting the equations (7) and (8) into (6), one obtains the dual problem:

$$\text{Max} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j < x_i, x_j > \tag{9}$$

$$\text{subject to } \alpha_i \geq 0, \ i = 1,...,N \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0$$

Generally, the data is not linearly separable and the previous treatment is modified by mapping the data X to an inner product space, H, called the feature space, in which it is initially assumed they will be linearly separable. In order to reformulate the dual problem of (9) for the space H, it suffices to note that the objective function only involves the inner product of the elements of X. If ϕ is the mapping of the input space to the feature space, H, then the new dual problem is:

$$\text{Max} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j < \phi(x_i), \phi(x_j) > \tag{10}$$

$$\text{subject to } \alpha_i \geq 0, \ i = 1,...,N \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0$$

Let $\{\alpha_i^0\}_{i=1}^{N}$ be the solution to problem (10), which theoretically is unique, then the classifier, *f*, is defined by:

$$f(x) = \text{sign}\left(\sum_{i=1}^{N} \alpha_i^0 y_i < \phi(x_i), \phi(x) > + b_0\right) \tag{11}$$

where b_0 is a constant whose value can be found from the optimal solution (c.f. [18]).

The relation of (11) to the hyperplane mentioned above is that its equation is:

$$\sum_{i=1}^{N} \alpha_i^0 y_i < z, \phi(x_i) > + b_0 = 0$$

where z is in feature space. Equation (11) then signifies that classification of test points takes place according to the side of the hyperplane the point lies on.

A modification is usually introduced into this formulation to deal with noise in the data. A so-called soft margin classifier is obtained by introducing a constant *C* and changing the restrictions of (10) to:

$$0 \leq \alpha_i \leq C, \ i = 1,...,N \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0$$

This represents a trade-off between maximizing the margin and minimizing the classification error on the training set. For an optimal solution $\{\alpha_i^0\}_{i=1}^{N}$ to the last formulation, it is found that for the points which lie on or within the margin, or are incorrectly classified, $\alpha_i^0 > 0$. These points are called support vectors. The rest of the points have $\alpha_i^0 = 0$. The support vectors are therefore the most informative points in the data set for the classifier, and the reformulated problem (10), using only these points, would produce exactly the same classifier.

2.1. Multiclass classification

The SVM classifiers are only defined for a two-class classification problem. In order to extend this to a multiclass classification problem several schemes have been proposed and there is, as yet, no definitive method. In the present paper a '1 vs rest' basis is adopted; that is, an SVM is found for each class by separating the data in that class from the rest of the training data. In this way, a different function in (11) is obtained for each class. A test pattern is then classified as belonging to the class with the maximum function value.

3. Kernels

It is important to note that the original dataset, X, does not have to belong to $\Re^n$ for the above formulation to work, in fact an arbitrary set, Σ, can be taken. The only ingredient that is required is a mapping ϕ of this set into an inner product space H. This allows SVM classifiers to be defined using data from arbitrary sets. In this general case, a kernel *K* is defined by:

$$K(x,y) = <\phi(x), \phi(y)> \tag{12}$$

It then follows that a kernel *K* satisfies the following two properties:

a) *K* is symmetric: $K(x,y) = K(y,x) \quad \forall\, x,y \in \Sigma$

b) *K* is positive definite: $\forall\, n \geq 1$, $\forall\, c_1,\ldots,c_n \in \Re$ and $x_1,\ldots,x_n \in \Sigma$

$$\sum_{i,j=1}^{n} c_i c_j K(x_i, x_j) \geq 0$$

It is interesting to note that given a function, *K*, satisfying these two properties, there exists a (Reproducing Kernel) Hilbert space H and a transformation ϕ: $\Sigma \rightarrow$ H such that (12) holds. This also allows elements of X to be compared for similarity, and in fact a simple calculation shows that a pseudo-metric d can be defined on $\Sigma x \Sigma$ through the relation:

$$d(x,y) = \|\phi(x) - \phi(y)\| = \sqrt{K(x,x) - 2K(x,y) + K(y,y)}$$

This introduces the concept of a distance into Σ.

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
J. Goddard, F. M. Martínez, A. E. Martínez & H. L. Ruffiner; "Noisy speech recognition using string kernels"
Proc. of the 9th International Conference Speech and Computer (SPECOM 2004), pp. 89-94, sep, 2004.

The polynomial and radial base function kernels, mentioned in the introduction, are defined on $\Re^m x \Re^m$ by:

$$K(x,y) = (<x,y> + 1)^d$$

$$K(x,y) = \exp(-\|x-y\|^2/2\sigma^2)$$

respectively, where d = 1,2,… and $\sigma \in \Re$. For these kernels the transformation in (12) is not defined explicitly, and the kernels are applied directly in the original data space. This is known as the 'kernel trick'.

3.1. String kernels

The present paper is concerned with kernels defined on discrete sequential data, called string kernels. They are defined by explicitly choosing a transformation ϕ from the sequential data into a suitable inner product space and using the definition of the kernel given by (12).

Let A be a finite alphabet of size N and Σ the set of all sequences of elements from A. For $n \geq 1$, $\Re^{N^n}$ can be considered as the usual euclidean inner product space indexed by all sequences from A of length n. The transformation ϕ:

$$\phi: \Sigma \rightarrow \Re^{N^n} \qquad (13)$$

can be defined for $x \in \Sigma$ by forming the vector in $\Re^{N^n}$ whose value for any index $\alpha \in A^n$ is the number of occurrences of α in x; more concisely, if

$$\phi(x) = \{r_\alpha\}_{\alpha \in A^n}$$

then

$$r_\alpha = \text{number of occurrences of } \alpha \text{ in } x \qquad (14)$$

To illustrate this transformation by means of a concrete example, let $A = \{1,2,3\}$, n = 2, and x = 1,2,2,2,3,1,1. Feature space is $\Re^9$ which is indexed by all pairs $(a,b) \in AxA$. The vector x is then transformed into $(1,1,0,0,2,1,1,0,0) \in \Re^9$.

Here only contiguous sequences are considered although in other papers, such as [14,16], non-contiguous sequences have been used.

A kernel *K* is defined as in (12) using the transformation from (13). The n-gram string kernel, *K*' is a normalized version of this, given by:

$$K'(x,y) = \frac{K(x,y)}{\sqrt{K(x,x)K(y,y)}}$$

It can be seen that two sequences in Σ are similar if they have similar n-tuples in common.

A variant of the n-gram kernel, which is termed the binary n-gram kernel here, gave better classification results in [15] on the isolated digit recognition task. The binary n-gram kernel is defined by modifying (14) to the following:

$r_\alpha = 1$ if α occurs in x, and 0 if not

and performing the same normalization.

In this case the example above would be transformed into the vector (1,1,0,0,1,1,1,0,0). Furthermore, for both transformations, the number of non-zero coordinates of $\phi(x)$ is bounded above by length(x)-n+1.

For the purpose of this paper, A can be considered to be {1, 2,…, 32}, an enumeration of the prototypes in a suitably defined vector codebook. The sequence a_1, a_2,…, a_r in Σ associated with a given speech signal is the same as that which would be associated with a dHMM; that is, the a_i's are the indices of the prototypes which are closest to the vectors derived from the mfccs of the frames in the signal.

Finally, motivated by the results of [21], a type of 'mismatch' kernel was also considered. In this case, and with the same notation as above, not only were the α's occurring in the sequence x considered to have $r_\alpha = 1$, but also combinations of the closest prototypes to those in the α's. For example, if n=2 and α=1,2 and the indices of the closest prototypes corresponding to 1 and 2 were 4 and 8 respectively, then the transformed vector also included non-zero values for the coordinates corresponding to (1,8), (2,4) and (4,8). The idea was to see if this type of mismatch kernel was more robust to certain levels of noise.

4. Data and results

The clean isolated Spanish digits data considered here was the same as that used in [15]. In the case of Spanish the digits are: cero, uno, dos, tres, cuatro, cinco, seis, siete, ocho and nueve. Here all the vowels in Spanish are represented and there are a variety of phonetic sounds, making it an interesting task. The recordings were obtained from 6 young Mexican adults, 5 male and 1 female. These recordings were made in a normal office environment without special equipment, which meant that the recording quality varied amongst the utterances. A sampling frequency of 16 kHz was used and the waveforms were converted to vectors with 13 coefficients, 12 mfccs and log energy.

Roughly 66% of the clean data was used for training and the rest for testing. The testing data was corrupted by adding white noise and speech babble from the NOISEX-92 database at different SNRs. In total the training set consisted of 19,718 vectors corresponding to 396 digits, and the test set of 9,896 vectors representing 197 digits. A k-means clustering algorithm was applied

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
J. Goddard, F. M. Martínez, A. E. Martínez & H. L. Rufiner; "Noisy speech recognition using string kernels"
Proc. of the 9th International Conference Speech and Computer (SPECOM 2004), pp. 89-94, sep, 2004.

to the training set, to obtain 32 prototypes. The digits were then represented by variable-length sequences of integers from {1,…,32}.

Examples of the spectrograms for the signal of a clean version of a test digit for 'seis' is given together with the noise corrupted versions at 50dB in Figure 2. The word 'seis' has the same fricative /s/ at the beginning and end and two vowels /e/ and /i/. The two vowels are easily distinguished in the spectrograms for the original signal (clean) and that with speech babble added because their formants are well defined. The spectrogram of the signal with white noise added has lost some of the spectral characteristics, especially for the /i/.

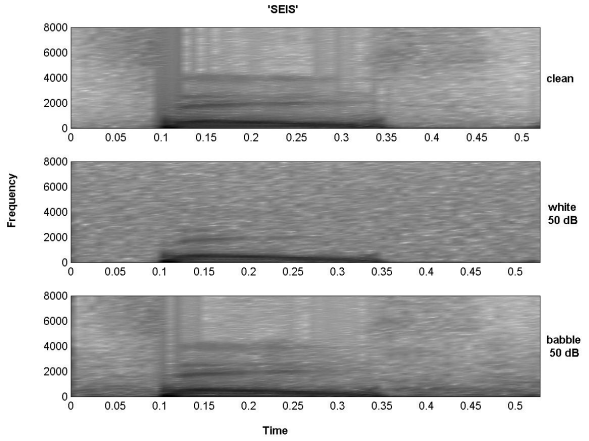

Figure 2: Spectrograms of the Spanish digit 'seis' corrupted with white noise and speech babble at 50dB

Various experiments were conducted using dHMMs, n-gram, binary n-gram and mismatch kernels. In the case of the dHMMs, a standard left-to-right architecture was used to model each individual digit, all with the same number of states. For the SVMs a value of 10 was chosen for C and no special tuning was conducted. In the case of the mismatch kernel a value of 2 was chosen.

Tables 1 and 3 show the classification results obtained for dHMMs with a different numbers of states (column heading) and a various SNRs. Tables 2 and 4 show the corresponding results for SVMs with different kernels.

Table 1: Classification results of added white noise at different SNRs with different dHMMs

white	1	2	3	4	5	6,7,8 9
0dB	10.66	11.17	9.14	9.14	9.65	11.68
10dB	11.17	12.18	9.65	10.15	9.14	12.18
20dB	13.71	20.31	15.74	13.20	13.20	14.21
30dB	17.26	24.37	21.32	19.29	19.80	17.26
50dB	31.47	37.56	38.07	37.56	39.09	39.60
clean	71.57	79.70	87.82	87.31	88.83	90.35

Table 2: Classification results of added white noise at different SNRs with different SVMs

white	binary 2-gram	binary 3-gram	binary 4-gram	2 -gram	mismatch
0dB	10.66	10.15	10.15	10.15	12.18
10dB	12.18	14.72	15.23	7.61	11.68
20dB	17.26	15.74	15.74	11.17	13.71
30dB	20.81	20.30	21.83	11.68	15.74
50dB	36.55	39.09	37.06	26.90	35.53
clean	89.34	91.37	88.83	66.50	83.76

Table 3: Classification results of added speech babble at different SNRs with different dHMMs

babbl e	1	2	3	4	5	6,7,8 9
0dB	17.26	21.83	13.71	16.24	13.71	15.23
10dB	25.89	24.37	19.29	29.44	27.92	24.87
20dB	28.93	31.47	30.47	37.56	37.06	35.03
30dB	40.61	46.70	43.66	47.72	53.81	57.36
50dB	62.94	64.98	72.08	73.10	73.60	76.65
clean	71.57	79.70	87.82	87.31	88.83	90.35

Table 4: Classification results of added speech babble at different SNRs with different SVMs

babble	binary 2-gram	binary 3-gram	binary 4-gram	2 -gram	Mismatch
0dB	18.78	17.26	19.80	15.23	19.80
10dB	23.86	24.37	23.35	20.81	23.86
20dB	26.90	26.90	29.44	25.89	32.49
30dB	44.16	50.76	53.30	37.56	46.19
50dB	73.10	79.70	74.62	48.22	64.47
clean	89.34	91.37	88.83	66.50	83.76

5. Conclusion

In [15] it was found that SVMs with binary n-gram kernels gave a comparable classification performance to those of dHMMs for the task of isolated Spanish digit recognition. In the present paper the question as to whether SVMs with string kernels can continue to give a similar proficiency to dHMMs in the context of noise was asked. To this end, the same test data as in [15] was corrupted with white noise and speech babble from the NOISEX-92 database at different SNRs. The actual training of the classifiers was conducted with the clean data, although this is known to worsen classification results when tested on noisy data.

The results obtained with the different classifiers are given in Tables 1-4. It is immediately noticable that white noise gives far worse results than speech babble for all the classifiers considered, and this is consistent with the fact that white noise is more destructive of a signals spectral properties. This is illustrated in Figure 2.

In terms of the results of the individual classifiers, the binary 3-gram and dHMMs with 6-9 states continue to be among the best although there is little to choose between them, and this is confirmed by statistical tests that were performed on them. The question arises as to why the best SVMs with string kernels maintain their performance with that of dHMMs. In [22] it is shown that n-gram kernels are in fact a particular case of a

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
J. Goddard, F. M. Martínez, A. E. Martínez & H. L. Rufiner; "Noisy speech recognition using string kernels"
Proc. of the 9th International Conference Speech and Computer (SPECOM 2004), pp. 89-94, sep, 2004.

visible Markov model. However curiously enough the 2-gram kernel gave the worst results of the kernels tried.

The mismatch kernel, whilst better than the 2-gram, did not prove to be robust as had been hoped.

6. References

[1] Vapnik, V. N., *The Nature of Statistical Learning Theory*. New York: Springer, 1995.

[2] Golowich, S., and Sun, D.X., "A Support Vector/Hidden Markov Model Approach to Phoneme Recognition", ASA Proceedings of the Statistical Computing Section, 125-130, 1998.

[3] Clarkson,P.R., and P.J. Moreno., "On the use of Support Vector Machines for Phonetic Classification", Proceedings IEEE International Conference on Speech and Signal Processing, Phoenix, USA, 1999.

[4] Ganapathiraju, A., Hamaker, J., and Picone, J,. "Support Vector Machines for Speech Recognition," Proceedings of the International Conference on Spoken Language Processing, Sydney, Australia, 2923-2926, 1998.

[5] Ganapathiraju, A., Hamaker, J., and Picone, J,. "A Hybrid ASR System Using Support Vector Machines," Proceedings of the International Conference of Spoken Language Processing, Beijing, China, vol. 4, 504-507, 2000.

[6] Salomon, J,. "Support Vector Machines for Phoneme Classification", Masters Thesis, University of Edinburgh, 2001.

[7] Abdulla, W. H., Kecman, V,. and Kasabov, N,. "Speech-background classification by using SVM technique", ICANN/ICONIP 2003 conference, Istanbul, Turkey, 310-315, 2003.

[8] Niyogi, P,. and Burges, C,. "Detecting and Interpreting Acoustic Feature by Support Vector Machines", Technical Report TR-2002-02, University of Chicago, 2002.

[9] Jaakkola, T.S., and Haussler, D., "Exploiting generative models in discriminative classifiers," in S. Kearns, S. A. Solla, and D. A. Cohn, editors, Advances in Neural Information Processing Systems 11, MIT Press, 1998.

[10] Smith, N.D., and Gales, M.J.F., "Using SVMs and discriminative models for speech recognition," International Conference on Acoustics, Speech, and Signal Processing, 2002.

[11] Haussler, D., "Convolution kernels on discrete structures," Technical Report UCSC-CRL-99-10, University of California in Santa Cruz, Computer Science Department, July 1999.

[12] Watkins, C., "Kernels from matching operations", Technical Report CSD-TR-98-07, Royal Holloway, University of London, Computer Science Department, July 1999.

[13] Leslie, C., Eskin, E., and Stafford Noble, W., "The spectrum kernel: A string kernel for SVM protein classification" *Proceedings of the Pacific Symposium on Biocomputing (PSB-2002)*, Hawaii 2-7, 564-575, 2002.

[14] Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C., "Text Classification Using String Kernels," *Journal of Machine Learning Research*, 2:419-444, Feb 2002.

[15] Goddard, J., Martínez, A.E., Martínez, F.M., and Rufiner, H.L., "A comparison of String Kernels and discrete Hidden Markov Models on a Spanish Digit Recognition Task Cancun, Mexico, Proceedings 25th Annual International Conference of the IEEE Engineering In Medicine And Biology Society, 2962-2965, 2003.

[16] Kirchner, R., "Exemplar-based phonology and the time problem: a new representational technique", 9th Conference on Laboratory Phonology, 2004.

[17] Junqua, J.C., and Haton, J.P., *Robustness in Automatic Speech Recognition: Fundamentals and Applications*, Kluwer Academic Publishers, Boston, 1996.

[18] Cristianini, N., and Shawe-Taylor, J., *Support Vector Machines*, Cambridge University Press, 2000.

[19] Campbell, C., "An Introduction to Kernel Methods,", in *Radial Basis Function Networks: Design and Applications*. R. J. Howlett and L.C Jain (eds). Berlin: Springer Verlag, 2000.

[20] Bennett, K.P., and Bredensteiner, E.J., "Geometry in Learning", Tech. Report, Department of Mathematical Sciences, Rennselaer Polytechnic Institute, 1996.

[21] Leslie, C., Eskin, E., Weston, J., and Stafford Noble, W., ``Mismatch String Kernels for SVM Protein Classification", In S. Becker, S. Thrun, and A. Obermayer, editors, A*dvances in NeuralInformation Processing Systems 15*, 2003.

[22] Saunders, C., Shawe-Taylor, J., and Vinokourov. A., "String Kernels, Fisher Kernels and Finite State Automata", In S. Becker, S. Thrun, and A.Obermayer, editors, *Advances in NeuralInformation Processing Systems 15*, 2003.

sinc(*i*) Research Center for Signals, Systems and Computational Intelligence (fich.unl.edu.ar/sinc)
J. Goddard, F. M. Martínez, A. E. Martínez & H. L. Rufiner, "Noisy speech recognition using string kernels"
Proc. of the 9th International Conference Speech and Computer (SPECOM 2004), pp. 89-94, sep, 2004.