Robust ASR using Support Vector Machines

R. Solera-Ureña, D. Martín-Iglesias, A. Gallardo-Antolín, C. Peláez-Moreno *, F. Díaz-de-María

Signal Theory and Communications Department, EPS-Universidad Carlos III de Madrid, Leganés-28911, Spain

Abstract

The improved theoretical properties of Support Vector Machines with respect to other machine learning alternatives due to their max-margin training paradigm have led us to suggest them as a good technique for robust speech recognition. However, important shortcomings have had to be circumvented, the most important being the normalisation of the time duration of different realisations of the acoustic speech units.

In this paper, we have compared two approaches in noisy environments: first, a hybrid HMM-SVM solution where a fixed number of frames is selected by means of an HMM segmentation and second, a normalisation kernel called Dynamic Time Alignment Kernel (DTAK) first introduced in [1] and based on DTW (Dynamic Time Warping). Special attention has been paid to the adaptation of both alternatives to noisy environments, comparing two types of parameterisations and performing suitable feature normalisation operations. The results show that the DTA Kernel provides important advantages over the baseline HMM system in medium to bad noise conditions, also outperforming the results of the hybrid system.

Key words: Robust ASR, additive noise, machine learning, Support Vector Machines, kernel methods, HMM, ANN, Hybrid ASR, Dynamic Time Alignment

* Corresponding author.
Email addresses: rsolera@tsc.uc3m.es (R. Solera-Ureña), dmiglesias@tsc.uc3m.es (D. Martín-Iglesias), gallardo@tsc.uc3m.es (A. Gallardo-Antolín), carmen@tsc.uc3m.es (C. Peláez-Moreno), fdiaz@tsc.uc3m.es (F. Díaz-de-María).

1 Introduction

Hidden Markov Models (HMMs) are, undoubtedly, the most employed core technique for Automatic Speech Recognition (ASR). During recent decades, research in HMMs for ASR has brought about significant advances and, consequently, the speech processing community has extensive know-how concerning the best choices for the design of HMM-based systems. Nevertheless, we are still far from achieving high-performance ASR systems. Some alternative approaches, most of them based on Artificial Neural Networks (ANNs), were proposed during the last decade ([2–5] are some examples). Some of them dealt with the ASR problem using predictive ANNs, while others proposed hybrid (HMM-ANN) approaches. Nowadays, however, the preponderance of HMMs in practical ASR systems is a fact.

However, it is well known that HMMs are generative models, i.e., the acoustic level decisions are taken based on the likelihood that the currently evaluated pattern had been generated by each of the models comprising the recognition system. Nevertheless, conceptually, the partial decisions previously alluded to are essentially classification problems that could be approached, perhaps more successfully, by means of discriminative models. Certainly, algorithms for enhancing the discrimination abilities of HMMs have also been devised. However, the underlying model remains generative.

Support Vector Machines (SVMs) are state-of-the-art tools for linear and nonlinear knowledge discovery ([6], [7]). SVMs rely on maximizing the distance between the samples and the classification function. Unlike others, such as ANNs or some modifications of HMMs that minimise the empirical risk on the training set, SVMs also minimise the structural risk [8], which results in a better generalisation ability. In other words, given a learning problem and a finite training database, SVMs generalise better than similar ANNs because they properly weigh the learning potential of the database and the capacity of the machine.

The maximised distance, known as the margin, is therefore responsible for their superior generalisation properties: the maximum margin solution allows the SVMs to outperform most nonlinear classifiers in the presence of noise, which is one of the longstanding problems in ASR. In a noise free system, this margin is related to the minimum distance a correctly classified sample should travel to be considered as belonging to the wrong class. That is to say, it indicates the 'noise' that added 'to the clean' samples is allowed into the system. This fact is one of the guidelines of this paper in which we have observed this robustness when applied to ASR in noisy environments showing improvements in their performance both using standard and robust parameterisations. More about this issue will be explained in section 3.

The improved discrimination ability of SVMs has attracted the attention of many speech technologists. Though this paper focuses on speech recognition, it is worth noting that SVMs have already been employed in speaker identification (for example, [9]) and verification (for example, [10]) or to improve confidence measurements that can help in dialog systems [11], among others.

Their application to the field of ASR, however, is not exempt from serious difficulties, some of which are shared with typical ANNs (and therefore some solutions have already been devised), but some others are exclusive to SVMs. In particular, we can indicate four fundamental shortcomings of SVMs that should be addressed in order to take advantage of them in ASR:

- **Temporal duration normalisation and speech segmentation into comprehensive units:** the ability to handle sequences of different temporal duration is one of the main strengths of the HMM systems and the reason why these generative solutions have so long prevailed over purely discriminative ones. Because this problem is so acute in ASR some of the proposals for its solution consist of taking advantage of HMM's effectiveness in that task and looking for alternative ways of obtaining the probability estimates necessary for the decoding. This is one of the most classical hybrid systems that have already been used in conjunction to ANNs. However, this raises the next of the problems we list here.
- **Probability estimation:** SVMs do not provide a probability estimation per se, thought some postprocessing of their internal variables can lead to some reasonable approximations that, nevertheless, differ from the true probability estimates needed.
- **Multiclass classification:** the SVMs original formulation solves a binary classification problem and it is not suited for the multiple class decisions we find in ASR tasks. This problem however, is a very common problem in the machine learning community and not exclusive to ASR. Therefore some effective solutions can be borrowed from that field.
- **Computational demands:** the enormous size of the speech databases used in ASR are hardly comparable with those encountered in the machine learning literature which makes the application to this very practical field an issue of major concern. Nevertheless, as we will outline, some solutions can also be found helpful.

Therefore, in this paper, we have carefully reviewed the state of the art in order to provide an overview of the main problems and solutions. We have then compared some of them under environmental noise distortions to assess their ability to cope with these problems.

This paper is organised as follows. The next section will be devoted to reviewing the state of the art on ASR using SVMs, considering the first and most exclusive ASR problem previously outlined. Then, in section 3 we explain the

fundamentals of SVMs from an ASR perspective trying to bring together the terms and definitions from both fields and providing the mainstream solutions to aforementioned problems two to four. In Section 4, we present the first of our proposals for the normalisation of the input feature vectors based on an HMM segmentation. The second, a particular instance of a sequence kernel, called Dynamic Time Alignment Kernel is introduced in section 5. In Section 6 we present the experimental framework and the results obtained. Finally, some concluding remarks and proposed further work draw the paper to a close.

2 State of the art

Although speech recognition is essentially a problem of pattern classification, the main reason that explains why ANNs, in general, have not been yet widely applied to ASR is the variable time duration of the speech segments corresponding to the acoustic units being considered for classification. In fact, this has been for many decades one of the fundamental problems to solve in the speech processing community and was the main element responsible for the success of HMMs.

Standard SVMs expect a fixed-length feature vector as input, but speech patterns are dynamic and this always leads to variable-length features. Different approaches have been presented to deal with the variable time duration of the acoustic speech units. Basically, solutions can be divided into those that aim to normalise the feature vector time dimension to fit the standard SVM input and those that explore string-related or normalizing kernels [6] to adapt the SVMs to variable input dimension and, therefore, are capable of using variable dimension vectors as inputs. In this section, we separately review the different variants of both alternatives and more details of the two instances from them we have chosen for our experimentation under noisy conditions can be found in sections 4 and 5.

2.1 Input feature vector normalisation

In this section, we will present various ways of normalizing the feature sequence, from uniform and non-uniform resamplings to hybrid SVM-HMM formulations aimed at coping with more complex tasks such as continuous speech recognition.

2.1.1 Uniform feature sequence resampling

In [12] several ways of preprocessing the speech sequence to obtain a fixed dimension vector are analyzed for a noisy digit recognition task. Two methods of uniform sequence resampling are assessed by performing variations on the size of the analysis window and the frame period: a variable window size method that makes it possible to include the whole digit utterance for a given number of windows per digit by adjusting the size of the window to the digit duration, and a fixed window size method, that maintains the window size around a fixed number of analysis instants regardless of its coverage of the digit. Therefore, in this last case, the windows are overlapping for short duration digits and, on the contrary, some information is missing for those of long duration.

In [13] their primary goal is to solve the problem of the computational complexity of the classical SVM formulation by using an alternative Lagrangian one on the TIMIT database. Their feature representation uses the previously explained variable window size method using different window lengths based on the duration of the phoneme being classified. Therefore, they concatenate 5 windows of the same size chosen from the set 32, 64, 128, 256, 400 covering the whole phoneme.

For the Indian consonant-vowel classification in [14], a different approach has been designed to account for the variation in the acoustic characteristics of the signal during the consonant-vowel transition. In this case the fixed length patterns are obtained by linearly elongating or compressing the feature sequence duration. As indicated, SVMs have shown a better performance than HMMs with the standard MFCC plus energy and delta and acceleration coefficients.

2.1.2 Non-uniform feature sequence resampling

In [15] they acknowledge the fact that the classification error patterns from SVM and HMM classifiers may be different and thus their combination could result in a gain in performance. They assess this statement on a classification task of consonant-vowel units of speech in several Indian languages obtaining a marginal gain by using a sum rule combination scheme of the two classifiers' evidences. As for feature length normalisation, they select segments of fixed duration around the vowel onset point, i.e. the instant at which the consonant ends and the vowel begins.

Another possible solution is shown in [12,16], where the non-uniform distribution of analysis instants provided by the internal state transitions of an HMM with a fixed number of states and a Viterbi decoder is used for dimensional normalisation. The rationale behind this proposal is that the uniform resampling methods are produced without any consideration of the information (or

lack of information) that speech analysis segments were providing. By selecting the utterance segments in which the signal is changing, it is hoped that a larger amount of information will be preserved in the feature vector. We have selected this scheme for our experiments as will be further explained in section 4.

2.1.3 Triphone model approach

Several authors use the so-called triphone model approach for the normalisation of the input feature length. This is motivated by the phonemes in context (PIC) model used in most state-of-the-art speech recognition systems that amounts to assuming that the speech segments (phones in most cases) are composed of a fixed number of sections. The first and third sections model the transition into and out of the segment, whereas the second section models the stable portion of the segment. In HMM based systems this is typically modelled by using 3 states per acoustic unit.

In [17] they show that SVMs provide a significant improvement in performance on a static pattern classification task based on the Deterding vowel data as well as on a continuous alphadigit one (OGI Alphadigits) and a large vocabulary conversational speech task (Switchboard). The segment vector resulting from the concatenation of the three segments corresponding to the triphone model is augmented with the logarithm of the duration of the phone instance to explicitly model the duration variability. The composite segment feature vectors are based on the alignments from a baseline three-state Gaussian-mixture HMM system. SVM classifiers are trained on these composite vectors, and recognition is also performed using these segment-level composite vectors.

In [18] they use SVMs for two different tasks using different feature length normalisation for each one. The first one is Thai tone recognition in which they try to classify the five different lexical tones in that language: mid, low, falling, high, and rising. A fixed number of measures of the pitch evolution is chosen in this case. However for the classification of Thai vowels they also divide each vowel into three regions using 12-order RASTA, plus its first and second derivatives taken from the center position of each segment.

In [19], the authors evaluate the performance of SVMs as classifiers, successfully comparing them with GMM (Gaussian Mixture Models) in both vowel-only and phone classification tasks. It is worth noting that a significant difference is observed in the problem of length adaptation between these two tasks. In the vowel case, it is acknowledged that regardless of the duration of each utterance, the acoustic representations are almost constant. Therefore simple features such as the formant frequencies or LPC coefficients corresponding to any time window are representative of the whole sequence. However, the rep-

resentation of the variations taking place in non-vowel utterances is essential to obtain an adequate input to SVMs. Thus, the triphone model approach has again been applied in this case, segmenting the number of frames obtained for each phone into three regions in the ratio 3-4-3 and subsequently averaging the features corresponding to the resulting regions.

Similar distinctions have been observed in [14], where a comparison between the performance of classical HMMs and SVMs for sub-word unit recognition is assessed for two different languages: 41 monophone units are classified in a Japanese corpus and 86 consonant-vowel units are considered for an Indian language. In this case, two different strategies have been devised to provide the SVMs with a fixed-length input: for the Japanese monophones, a similar technique to that proposed in [19] has been used. The frames comprising each monophone have been divided into a fixed number of segments. An averaged feature vector is then obtained for each segment. Each feature vector is subsequently concatenated to those resulting from other segments to form input vectors for the SVM classifier. For the Indian consonant-vowel classification, however, a uniform resampling approach has been designed as explained in subsection 2.1.2.

2.1.4 Hybrid systems

The problem of the need for a fixed-length input representation is not exclusive to SVMs. Most common ANNs also require this type of feature. Therefore, several proposals were made in the 90's to cope with this problem, the most successful being the combination of HMMs and ANNs into a single system to profit from the main properties of both approaches: the ability of HMMs to model the temporal nature of the speech signal and the discriminative learning provided by ANNs. Following this principle, different classes of hybrid ANN/HMM systems have been developed. In later paragraphs, we briefly describe some of the most relevant ones. A complete survey about this subject can be found in [20].

The most common approach to hybrid systems is that initially proposed in [5] in which ANNs are used to estimate HMM emission probabilities. In fact, one output of a neural network is associated with each HMM state and trained to estimate the posterior probability of this state given the acoustic observation sequence. This probability can be converted to the required emission probability using Bayes' rule. Several types of neural networks have been used for this purpose: MLPs [5], RNNs [21] and even Radial Basis Function (RBF) networks [22].

The application of this type of systems with SVMs is not straightforward because, as we have already mentioned, a probability estimate is not readily

available in these classifiers. We will review the solutions to this problem in section 3. An example of this system can be found in [23].

Another type of hybrid ANN/HMM system is proposed in [24] where the ANN is used to determine the non-linear transformation of acoustic vectors more suitable for a standard CDHMM-based system, in such way that, all the parameters of the combined system (feature transformation and HMM models) are jointly trained according to a global criterion.

Some alternative ways to use both HMMs and SVMs consist of using the former to generate phonetic level alignments that are treated individually by the latter to perform phoneme identification. As each segment can last for different amounts of time, some methods such as the ones outlined in the previous two subsections is needed to convert them into fixed-length vectors. We find examples of this technique in [17,25–28]. These authors aim to provide a solution for the continuous speech recognition problem. HMM classifiers are designed to provide the SVM with the appropriately segmented speech acoustic units [17,25,26]. The previously mentioned triphone model approach is then applied to normalise the length of the input vector extracted from each acoustic unit. In [27,28], a Bayesian-based modification of SVMs called RVM (Relevance Vector Machine) is proposed to improve the system.

As we have already mentioned in 2.1.2, in [16] the HMM state transitions are used to provide analysis instants. This procedure can also be included as a hybrid example and will be explained in greater detail in section 4.

A more recent approach is presented in [29], where instead of completely switching paradigm from HMM to SVM or trying to couple SVMs with HMMs they study how to directly estimate Gaussian mixture continuous density HMMs (CDHMMs) for speech recognition based on the large margin principle. In other words, they attempt to estimate CDHMM parameters in such a way that the decision boundary determined by the estimated CDHMMs achieves the maximum classification margin as in SVMs. They have evaluated this system in the speaker-independent isolated E-set recognition and the TIDIGITS connected digit string recognition tasks.

2.2 Normalizing kernels

We have already mentioned the notion of kernel in SVMs and, in fact, SVMs are examples of the more general class of kernel methods. As we will further discuss in section 3, SVMs rely on a kernel (inner product in a feature space) to obtain a nonlinear decision function. This kernel defines the space in which the solution is sought and therefore its choice is problem-dependent. In this paper we have selected RBF (Radial Basis Function) kernels as well as linear,

both in the standard solution and in conjunction with a normalizing kernel based on dynamic time alignment (DTAK). We have evaluated it in our noisy scenario task as we describe in section 5.

This kernel is first proposed in [1] and [30], where the idea of non-linear time alignment is incorporated into the kernel function. Since the time-alignment of sequential patterns is embedded in the kernel function, standard SVM training and classification algorithms can be employed. More details about this procedure can be found in section 5.

In [31–33] the emphasis is placed on the use of another type of string or sequence kernel to perform length normalisation. Here, the variable length speech sequences are mapped to vectors of fixed dimension using the so-called Fisher score.

The Fisher kernel was first used in the field of biology, in the context of DNA and protein sequence analysis [34], although there are also some interesting results in the field of speaker verification [35].

The idea of the method is to use a score function calculated using the a posteriori probability of the observation obtained with a generative model as a kernel. Since the generative model is capable of working with sequences of different lengths, the resulting Fisher kernel will also be. In [32] some generalisations of this kernel are evaluated in the ASR framework.

3 Support Vector Machine fundamentals

In this section, our purpose is to introduce the basic notions of Support Vector Machines emphasizing the characteristics related with their use in speech recognition.

An SVM is essentially a binary classifier trained to estimate whether an input vector $\mathbf{x}$ belongs to a class 1 (the desired output would be then $y = +1$) or to a class 2 ($y = -1$). The decision is made according to the following expression:

$$g(\mathbf{x}) \gtrless 0, \tag{1}$$

where the function $g(x)$ takes the form:

$$g(\mathbf{x}) = \mathbf{w}^T \cdot \phi(\mathbf{x}) + b, \tag{2}$$

where $\phi(\mathbf{x}) : \Re^n \mapsto \Re^{n'}$, $(n << n')$, is a nonlinear function which maps vector

$\mathbf{x}$ into what is called a *feature space* of higher dimensionality (possibly infinite) where classes are linearly separable. The vector $\mathbf{w}$ defines the separating hyperplane in such a space and b represents a possible *bias*. It is worth noting that the meaning of *feature space* here has nothing to do with the space of the speech features that, within the kernel methods nomenclature, belong to the *input space*. As we can observe, it is this *input space* whose dimension, n, should be fixed in the standard SVM formulation and that is to blame for all the efforts of feature length normalisation mentioned in section 2.

The reason that makes SVMs more effective than other methods based on linear discriminants is its learning criterion. The goal of any classifier must be to minimise the number of misclassifications in any possible set of samples. This is known as Risk Minimisation (RM). However, in typical classification problems we only have a limited number of samples available (in some cases we can have an unlimited number of samples but, in any case, we only can deal with a subset), and thus, all we can do is to try to minimise the number of misclassifications within the training set. This is known as Empirical Risk Minimisation (ERM), and most classifiers base their learning procedure on it.

However, having the classifier with the best ERM is not enough (or even desirable). The complexity of the classifiers must normally be fixed a priori, and so, we can end up having a too simple structure incapable of modeling correctly the classification boundaries of our problem, or a too complex one, overfitted to our training set and incapable of generalizing to unseen samples. This is known as Structural Risk, and a good classifier must maintain a compromise between the ERM and the SRM.

In SVMs, one of the best advantages is that we do not need to fix the complexity of the resultant machine a priori. What we need is to fix a parameter which establishes this compromise between ERM and SRM.

Thus the solution of the SVM is given by the following minimisation problem:

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i} \quad & \frac{1}{2}\mathbf{w}^T \cdot \mathbf{w} + C\sum_{i=1}^{N}\xi_i, \\ \text{subject to} \quad & y_i(\mathbf{w}^T \cdot \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \text{ for } i = 1, \cdots, N, \end{aligned} \tag{3}$$

where $\mathbf{x}_i \in \Re^n$ $(i = 1, \ldots, N)$ are the training vectors corresponding to the labels $y_i \in \{\pm 1\}$, and the variables ξ_i are called *slack variables* and allow a certain amount of errors that contribute to obtaining solutions in the non-separable case. The parameter C, on the other hand, allows us to establish the mentioned compromise between ERM and SRM, balancing error minimisation

and generalisation capability. Unfortunately, we do not have a method to know the most suitable value for this parameter a priori, and we must resort to cross-validation.

The SVM problem is usually solved by introducing the restrictions in the minimizing function using Lagrange multipliers, leading to the maximisation of the Wolfe dual:

$$L_d = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j), \tag{4}$$

with respect to α_i and subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq C$. This problem is quadratic and convex, so its convergence to a global minimum is guaranteed using quadratic programming (QP) schemes. This is an advantage compared to other classifiers such as ANNs that often fall in local minimums. Solving this problem, the optimum decision boundary $\mathbf{w}$ will be given by:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \phi(\mathbf{x}_i). \tag{5}$$

According to (5), only vectors with an associated $\alpha_i \neq 0$ will contribute to determining the weight vector $\mathbf{w}$ and, therefore, the separating boundary. Due to this fact, they receive the name of *support vectors*. These vectors define the separation border and the margin we have already mentioned in section 1. It is the maximisation of this margin that makes these machines robust and, in our opinion, very well suited for ASR in noisy environments.

Generally, function $\phi(\mathbf{x})$ is not explicitly known (in fact, in most of the cases its evaluation would be impossible as long as the feature space dimensionality can be infinite). However, we do not actually need to know it, since we only need to evaluate the dot products $\phi^T(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ which, by using what has been called the *kernel trick*, can be evaluated using a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$. Many of the SVM implementations compute this function for every pair of input samples producing a *kernel matrix* that is stored in memory. This is one of the main computational problems of these algorithms that prevent their application in very large speech databases. However, some solutions are already being developed [36–38].

By using this method, the form that finally adopts an SVM is the following:

$$g(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \tag{6}$$

The most widely used kernel functions are the simple *linear* kernels,

$$K_L(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \cdot \mathbf{x}_j, \tag{7}$$

the *Gaussian Radial Basis Function* (RBF),

$$K_{RBF}(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right), \tag{8}$$

where γ is proportional to the inverse of the variance of the gaussian function and whose associated feature space is of infinite dimensionality. Also very common is the polynomial kernel

$$K_P(\mathbf{x}_i, \mathbf{x}_j) = \left(1 + \mathbf{x}_i^T \cdot \mathbf{x}_j\right)^p, \tag{9}$$

whose associated feature space are polynomials up to degree p. It is worth mentioning that there are some conditions that a function should accomplish to be used as a kernel. These are often denominated KKT (Karush-Kuhn-Tucker) conditions and we will revisit them in section 5 to check that the DTA Kernel can be effectively used.

An important issue that comes up with both HMMs and SVMs is data normalisation. As data normalisation becomes essential when working with noisy speech, we have tried out several types on the SVMs input vectors. The best results were obtained by subtracting the mean to all parameters and dividing them by their maximum values in a per file basis. So, if we denote by x_n^i the i-th component of the n-th feature vector $\mathbf{x}_n$ and by $\bar{x}^i$ the mean of the i-th component over all the training samples, the normalisation is performed as follows:

$$\hat{x}_n^i = \frac{x_n^i - \bar{x}^i}{\max\limits_n \left(x_n^i\right)} \tag{10}$$

Another difficulty that arises when applying SVMs in ASR is that speech recognition is a problem of multiclass classification, whereas in the original formulation an SVM is a binary classifier.

There are several ways we can solve k-class problems using Support Vector Machines (SVM). The first one, proposed by Vapnik in 1995 [8], compares each class against all the others. We will need to train k binary classifiers, in which the true class takes a positive value and the remaining classes a negative value. To test for a new point all the classifiers are evaluated and the test sample is assigned to the classifier with largest output. This multiclass

SVM is known as one-versus-all or one-versus-the-rest. This is the type of multiclass classification implemented in the publicly available tool TorchSVM as explained in its manual [39].

There is another proposal known as one-versus-one in which each class is compared against all the other classes. In this case we will need order k^2 classifiers (specifically $\frac{k(k-1)}{2}$), but each classifier will be trained with a small fraction of the samples. Several empirical studies [40,41] show that for large datasets using this approach is more efficient (in runtime complexity) than using the one-versus-all approach. Because SVM training is nonlinear in the number of samples and we are better off training more classifiers with fewer samples than training few classifiers with many samples. Also each classifier will be simpler as some classes can be easily separated. The complexity of binary SVM classifiers can be checked in [42]. This approach is the one used by LibSVM [43] and the one we proposed in this paper.

There are other approaches to solve the multiclass SVM using binary-SVM classifiers; comparing several classes against each other. But they do not perform significantly better (or worse) than the ones previously commented. For a survey paper, the readers can refer to [44].

However, these two alternatives cannot be considered a *true* multi-class solution since it relies on the combination of several binary SVMs trained independently. Some reformulations of the SVM equations to consider all classes at once can be found in [45] and [46]. The difference between these methods is subtle, as they only differ in how the slack variables are penalised. The first one, for each sample, penalises all the incorrect classes that provide an output larger than the true class does. The second one only penalises the incorrect class that gives the largest output, if it is larger than the output of the true class. These methods are limited by its computational complexity. Because they need to compute Gram-matrix that is $kn \times kn$, where n is the number of samples, while the one-versus-all Gram-matrix is $n \times n$. These methods are far more inefficient than one-versus-all or one-versus-one for classification problems in which either n or k are large. In speech recognition in which both are large, these methods are impractical, as we have described in the text.

Support Vector Machines are state-of-the-art tools for classification tasks. Due to their max-margin training paradigm, they do not directly provide calibrated posterior probability outputs but class labels. Nevertheless, some methods have been proposed to extend SVMs for approximated probability estimates. The most widely used one (implemented in [43]) when dealing with multiclass problems is based on the calculation of Platt's probability [47] for every input sample and binary machine $(i, j)\,, \forall i, j \in [1...k]$. With input pattern $\mathbf{x}$ with associated label y, then Platt's probability of $\mathbf{x}$ belonging to class i for SVM (i, j) is calculated as follows:

$$r_{ij}(\mathbf{x}) = P\left(\text{y=i|y=i or j,}\mathbf{x}\right) = \frac{1}{1+e^{Ag(\mathbf{x})+B}} \qquad (11)$$
$$r_{ji}(\mathbf{x}) = P\left(\text{y=j|y=i or j,}\mathbf{x}\right) = 1 - r_{ij}(\mathbf{x})$$

where sigmoid's parameters A and B are estimated by minimizing the negative log-likelihood function over training data.

Then, a version of the Refregier-Vallet method ([48],[49]) is used to translate these two-class probabilities $r_{ij}(\mathbf{x})\ \forall i,j$ to multiclass probabilities $p_i(\mathbf{x}) = p(y=i|\mathbf{x})\ \forall i$. Posterior probability vector $\mathbf{p}(\mathbf{x}) = [p_1(\mathbf{x}),\ p_2(\mathbf{x}), ..., p_k(\mathbf{x})]$ for input pattern $\mathbf{x}$ is obtained by solving the following optimisation problem:

$$\min_{\mathbf{p}} \frac{1}{2} \sum_{i=1}^{k} \sum_{j:j\neq i}^{k} \left(r_{ji}p_i - r_{ij}p_j\right)^2 \quad \text{s.t.} \quad \sum_{i=1}^{k} p_i = 1,\ p_i \geq 0\ \forall i \qquad (12)$$

This problem is convex and can be solved by means of a simple iterative method.

4 Hybrid HMM-SVM system

As we have already mentioned in section 2, the use of SVMs in ASR is by no means straightforward. The main problem stems from the fact that usual kernels can only deal with (sequences of) fixed length vectors. However, standard parameterisation techniques generate variable length sequences of feature vectors depending on the time duration of each speech utterance. Therefore, either we perform a previous dimensional (time) normalisation or we employ a non-standard kernel capable of handling such variation. We will review the first of these options in this section explaining the first of our candidates for robust speech recognition we already introduced in [12,16], leaving the second option for section 5.

The basic idea underlying this method is that an appropriate selection of the time instants at which the speech signal is analysed can improve the results. Due to the fact that most discriminative information in speech is associated with spectral changes, it seems sensible to consider a segmentation made by HMMs based on state transition instants, which are very likely related to those at which changes of the speech spectra happen.

The HMM-guided parameterisation procedure used in [16] has two main stages. The first one consists of an HMM classifier with a Viterbi decoder that yields the best sequence of states for each utterance and also provides a set of state

boundary time marks. The second stage extracts the speech feature vectors at the time instants previously marked. For the first stage, we have used left-to-right HMMs with a three continuous density Gaussians mixture per state. Each HMM is a whole-word model and consists of N states. In order to ensure that no state is skipped (since we need to always have the same number of state boundary time marks), only self loops or transitions to adjacent states are allowed; in other words, transitions between non-adjacent states are not allowed.

As previously mentioned, these acoustic models are used to generate alignments at state-level for all utterances in the speech database. In this process, each utterance is compared to each of the HMMs and only the segmentation produced by the acoustic model yielding the best score is saved for the next stage. Note that the segmentation obtained may not be correct or accurate enough, even when the utterance is properly recognised by the HMM-based system. It will be shown later in the paper that this is a major drawback for this system.

In the second stage, feature vectors are extracted at the time instants derived from the HMM-guided segmentation. This way, the number of feature vectors per utterance turns out to be equal to the number of state transitions ($N - 1$), determined by the HMM topology used. In our case, N was fixed to 15 (the same number of states we used for the baseline HMM-based recognition system) as a trade-off between word recognition accuracy and computational cost. Finally, all of the 14 feature vectors are concatenated to form a single vector for each speech utterance.

The results presented in section 6 show that these systems do not take full advantage of the SVMs' classification properties, very likely due to errors in the selection of analysis instants provided by the HMM-guided segmentation. As previously stated, this segmentation may not be accurate enough, so time stamps might not correspond to time instants at which speech spectra changes. Besides, the preselected number of instants may not be result appropriate for all the words in the vocabulary, thus failing to properly identify the true spectral changes. It would be interesting to develop a system in which the previous segmentation process is not needed. These are the reasons that led us to use Dynamic Time Alignment Kernel (DTAK).

5 SVM-based System: Dynamic Time Alignment Kernel

This method was introduced in [1] and [30], and it tries to solve the problem of different length sequences by adapting the kernel of the SVM to one capable of working with samples of variable dimensionality. This seems to be a more

natural approach than performing a previous segmentation, and allows us to completely avoid the use of HMMs in the recognition process.

In short, this technique uses the score obtained by means of a Dynamic Time Warping (DTW) algorithm as a kernel. DTW algorithms were one of the first techniques used in speech recognition and they were widely used in the 70's [50].

DTW measures the distance between a target signal and a template, expanding or contracting the temporal axis of the target to find the *path* or *warping function* which maximises the similarity between the two signals. The distance of the signals is calculated at each instant along the warping function, and the final score given by the algorithm is the accumulated similarity. Any metric can be used to calculate this distance but usually the Euclidean is employed. In the case of DTAK, the inner product is used and therefore this distance can be interpreted as a linear kernel that is employed internally for the computation of the DTA Kernel. With such an interpretation, it is now possible to substitute this distance metrics for the one provided by non-linear kernels such as RBF as we will introduce further on.

Specifically, we use the following procedure to calculate the linear kernel: if X and Y are the two sequences of feature vectors to be compared, and $\psi_I(k)$ and $\psi_J(k)$ are warping functions which normalise the temporal axis of the sequences in the instant k, we must find the solution to the new *inner product*:

$$\begin{aligned} K_{DTA}(X,Y) = X \circ Y \;\; &= \;\; \max_{\psi_I,\psi_J} \frac{1}{M_\psi} \sum_{k=1}^{L} m(k) \mathbf{x}_{\psi_I(k)}^T \cdot \mathbf{y}_{\psi_J(k)}, \\ &\text{subject to} \;\; 1 \leq \psi_I(k) \leq \psi_I(k+1) \leq |X|, \\ &\qquad\quad\;\; 1 \leq \psi_J(k) \leq \psi_J(k+1) \leq |Y|, \end{aligned} \tag{13}$$

where M_ψ is a normalisation factor which normally has the value $M_\psi = |X| + |Y|$, and $m(k)$ is a non-negative scale factor which gives more importance to some particular "steps" in the "path".

This optimisation problem is normally solved by means of *dynamic programming*, using the following recursive equation:

$$D(i,j) = \max \begin{cases} D(i-1,j) + \mathbf{x}_i^T \cdot \mathbf{y}_j, \\ D(i-1,j-1) + 2\mathbf{x}_i^T \cdot \mathbf{y}_j, \\ D(i,j-1) + \mathbf{x}_i^T \cdot \mathbf{y}_j. \end{cases} \tag{14}$$

where scale factor '2' favors translations along the diagonal, which should be the most probable ones. Therefore, the DTA Kernel becomes reduced to:

$$K_{DTA}(X,Y) = X \circ Y = D(|X|,|Y|)/(|X|+|Y|) \tag{15}$$

It is worth mentioning that in contrast with the classical template-based ASR solutions where the difficulty of finding an appropriate template was the main drawback that led to the supremacy of model based approaches like HMM, the DTAK solution automatically finds the best reference templates using the max-margin criterion.

Effectively, if we look at equation (5) in section 3, we see that only those templates with an associated $\alpha_i \neq 0$ will be relevant and will contribute to determining the separating boundary. Only a few templates will have a non-zero α_i, and these will be those closest to the decision function. Now, the *support vectors* are *support sequences* or templates.

Furthermore, the algorithm not only selects those appropriate templates that define the decision boundary but also the number of them that minimise the structural risk, and this is accomplished by giving an appropriate value to parameter C in equation (3). Unfortunately, we do not have a method to calculate the best value for this parameter *a priori*, so we must resort to cross-validation.

With the previous formulation it is now easy to consider the generalisation that allows us to find the separating border in a higher dimension space (the feature space) by means of a non-linear kernel like an RBF. We have said that, basically, DTAK consists of using DTW as the kernel of an SVM. Now, a generalisation that performs the time-warping in the feature space can be considered. In other words, in equation (13), we could use a kernel function (for example, an RBF) instead of a conventional dot product and the DTAK kernel would have the following form:

$$\begin{aligned} &K_{sDTA}(X,Y) \\ &\quad = \phi(X) \circ \phi(Y) = \max_{\psi_I,\psi_J} \frac{1}{M_\psi} \sum_{k=1}^{L} m(k) K_{RBF}(\mathbf{x}_{\psi_I(k)}, \mathbf{y}_{\psi_J(k)}). \end{aligned} \tag{16}$$

Now, to prove that K_{sDTA} is a valid kernel we only have to show that it is symmetrical and positive semidefinite as we mentioned in section 3. The former is obvious, since the warping function is the same if we interchange sequences X and Y. Regarding the latter, we must demonstrate that:

$$\mathbf{u}^t\mathbf{K}_s\mathbf{u} \geq 0 \ \ \forall \mathbf{u}. \tag{17}$$

This is easily proved if we consider that DTW is the (weighted) sum of the inner products (kernels) of the vectors composing sequences X and Y at the instants defined by the optimal warping function $\psi^*(k)$. That is (omitting the scale factors):

$$\mathbf{K}_s = \mathbf{K}_{(1)} + \cdots + \mathbf{K}_{(L)}, \tag{18}$$

where $\mathbf{K}_{(k)}$ is the kernel at the instant defined by $\psi^*(k)$. So,

$$\begin{aligned} \mathbf{u}^t\mathbf{K}_s\mathbf{u} &= \mathbf{u}^t(\mathbf{K}_{(1)} + \cdots + \mathbf{K}_{(L)})\mathbf{u} \\ &= \mathbf{u}^t\mathbf{K}_{(1)}\mathbf{u} + \cdots + \mathbf{u}^t\mathbf{K}_{(L)}\mathbf{u} \\ &\geq 0, \end{aligned} \tag{19}$$

since $\mathbf{K}$ is a valid kernel and, therefore, positive semidefinite.

6 Experiments and results

6.1 Databases

For our experimentation, we have used two different speech databases: a proprietary isolated digits database (should be denoted here as SI database) and a subset of the Spanish SpeechDat database.

The SI database consists of 72 speakers with 11 utterances per speaker for each of the 10 Spanish digits (7920 files). This database was recorded at 8 KHz in clean conditions. Since the database is too small to achieve reliable speaker-independent results, we have used a 9-fold cross validation to artificially extend it, averaging the results afterwards. Specifically, we have split the database into 9 balanced speaker-disjoint groups, each of them containing all utterances corresponding to 8 of the speakers. One different group is kept for testing in each fold, while the remainder are used for training. We name the particular fold being used by specifying the group used for testing. For example, 'fold 1' refers to the division of data in which the first of the 9 groups is used for testing and the remaining 8 are used for training, 'fold 2' refers to the division of data in which the second of the 9 parts is used for testing and the remaining 8 are

used for training, and so on. This way, for every fold all the utterances from 64 of the speakers are used for training and 8 for testing; speaker-independent task is achieved given that we consider disjoint training and testing subsets. However, there are some parameters in the SVM implementation that need to be tuned (in particular, γ and C for the RBF versions and only C for the linear). Thus we have used fold 1 to provide a tuning set for optimizing the values of these parameters. After these parameters are optimised, this fold is removed from the final averaged results, i.e., only folds 2-9 are used to provide the final numbers. Therefore, utterances used to decide on parameter tuning are never reused in testing. In order to have the same test conditions for both HMM and SVM experiments we have also removed fold 1 from the HMM results. In summary, the total number of testing utterances is 7040.

In order to validate the conclusions obtained with the first database, we designed a second set of experiments using other data. For this purpose, we have used a subset of the SpeechDat Spanish Database for Fixed Telephone Network [51], a speaker-independent speech corpus collected over the Spanish telephone network with 4000 different speakers and recorded at 8 KHz (A-law). The subset consists of the files containing isolated digits (only one utterance per speaker). The protocols regarding speech-independent training and testing are similar to the previous case. Again, to narrow the confidence intervals we have designed a 5-fold cross validation experiment with 4 groups (roughly $4 \cdot 800 = 3200$ speakers and utterances) for training and the remaining one (approximately 800 speakers and utterances) for testing in each fold. We have also kept the first fold for tuning γ and C in the SVM implementations. After these parameters are optimised, this fold is removed from the final averaged results in both HMM and SVM experiments. The total number of testing utterances is 3120 due to the fact that some of the available files are corrupted. As with the previous database, data from one speaker is never used in both training and testing in the same fold in order to achieve a speaker-independent speech recognition task.

6.2 Database contamination

We have tested our systems in clean conditions and in the presence of additive noise. For that purpose, we have corrupted our database with two kinds of noises, namely: white noise and the noise produced by a F16 plane. Both noises have been extracted from the NOISEX database [52] and added to the speech signal to achieve four different signal-to-noise ratios (SNRs): 12 dB, 9 dB, 6 dB and 3 dB.

To add a certain noise at a desired SNR, noise samples are multiplied by an attenuation factor before adding them to the speech samples. This factor

depends on the speech and noise rms values calculated over the corresponding whole file, so it is computed for each speech file [1] .

As we have used clean speech for estimating the acoustic models (in both, HMM and SVM-based recognisers), the noises are only added for testing the recognition performance.

It is worth clarifying that the aim of the experiments with additive noises is to perform a robustness comparison between the HMM- and the SVM-based systems by themselves. Therefore, specific methods to deal with additive noises are not implemented in any of the systems.

6.3 Front-end description

In our experimentation, we have used two different parameterisations: MFCCs and LP-MFCCs. Both parameterisations are well-known and their characteristics have been deeply studied. In particular, several studies have shown that LP-MFCCs are more robust against noisy conditions than MFCCs. As will be shown in subsection 6.4, our experiments show that this conclusion is still valid when using SVM-based recognisers instead of the conventional HMM-based back-end.

The difference between both front-ends relies on how the speech spectrum is obtained. In particular, the *Spectral Analysis* stage in the MFCC computation is replaced by two different steps in the LP-MFCC case: *Pole Modeling* and *Spectrum Envelope Computation* (see [53] for further details).

In the MFCC-based front-end, the spectral analysis is performed by using a 256-point FFT, from which we only use the 128-sample positive half of the full spectrum. In the LP-MFCC parameterisation, the order of the all-pole model is 12, so 12 LP coefficients are computed. Next, a 256-point spectral envelope of the speech frame is derived from these LP coefficients. As before, only the half of the full spectrum is used.

In both cases, we have used a 25 ms Hamming analysis window, a preemphasis filter with a preemphasis coefficient of 0.97 and a filter bank composed of 40 triangular filters distributed following a Mel scale. Finally, 12 coefficients (MFCC or LP-MFCC) are obtained at a frame period of 10 ms. These static features are extended with the log-energy of each frame and the corresponding first order delta parameters, making a total vector dimension of 26 components.

[1] In [16] the rms values were computed over the whole database.

It is well-known that the normalisation of the acoustic features is very convenient for achieving a better performance in noisy conditions. Therefore, we carried out a set of preliminary experiments in order to choose the best feature normalisation for each kind of back-end. Finally, the conventional CMN is used for the HMM-based recogniser, while in the case of the SVM-based back-end, each component of the feature vector is re-normalised by dividing it by its maximum value over the whole utterance.

6.4 Experimental results

In this section, we describe the experiments carried out in order to test the different recognisers proposed: HMM-based, hybrid HMM-SVM-based (with linear and RBF kernels) and DTAK-based systems (with linear and RBF kernels) in clean and noisy conditions.

In order to get a fair comparison between all the systems proposed, for each of them we have properly selected the values of the different configuration parameters involved through a set of preliminary experiments.

The HMM-based approach is an isolated-Spanish digit, speaker-independent ASR system developed using the HTK toolkit [54]. Each of the whole-digit models is a left-to-right HMM with continuous observation densities in which the number of states has been selected based on preliminary experiments in order to maximise performance. Finally, we have used 15 states per model and three Gaussians per state.

The SVM-based recognisers are described in sections 4 and 5. However, in following paragraphs we will describe some relevant details concerning their configuration and training procedure.

With regard to the number of states used for segmentation in the hybrid HMM-SVM system, we have the problem that the different words in the dictionary have different lengths. This implies that for a given number of states some words can be oversampled, while, for others, the number of selected acoustic vectors is not large enough. For both hybrid systems (linear and RBF), we have used a 15 state HMM to produce the sampling instants at which the speech signal is analysed. Thus, in this case, we have used 14 feature vectors (the transitions) per utterance as SVM input. The number 15 was chosen because with less, the recognition rate was poor, and with more, the computational cost was very high, while the improvement in recognition rate was not so noticeable.

In the experiments where we have used RBF kernels for both approaches, hybrid and DTAK-based systems, we have found values for γ and for the reg-

ularisation parameter C of the SVM using grid search and the cross validation procedure implemented in [43]. We did not find a considerable difference in performance using different values of C but we have found that the system is moderately sensitive to the values of γ. Besides, due to computational issues the possible paths in the RBF-DTAK system have been restricted.

Finally, all the back-ends have been trained using clean speech (without additive noise).

With the objective of stating the statistical significance of the experimental results shown in following subsections, we have calculated the confidence intervals (for a confidence of 95%) using the following formula ([55], pp. 407-408):

$$\frac{\Delta}{2} = 1.96\sqrt{\frac{p\,(100-p)}{n}} \tag{20}$$

where p is the word recognition rate and n is the number of examples to be recognised (7,040 and 3,120 words for the SI and SpeechDat databases, respectively). Thus, any recognition rate in the subsections below is presented as belonging to the band $[p - \frac{\Delta}{2}, p + \frac{\Delta}{2}]$ with a confidence level of 95%.

Confidence intervals are displayed only in the graphics, since the tables became difficult to read when they are included.

6.4.1 Results with the SI database

Tables 1 and 2 show the word recognition rates obtained with the different alternatives of hybrid and DTAK systems in comparison to those achieved by the HMM-based system for the SI database and the MFCC and LP-MFCC parameterisations, respectively. These results correspond to clean conditions and white and F16 noise conditions at four signal-to-noise ratios: 12 dB, 9 dB, 6 dB and 3 dB.

As can be observed, for both, parameterisations noise significantly degrades the performance of all the systems considered. The decrease in the recognition performance is more noticeable for the white noise, which is known to produce more distortion in speech signals than other noises with low-pass characteristics (as the F16 noise).

6.4.2 Results with the SD database

Tables 3 and 4 show the recognition rates for the SpeechDat database and several noise conditions for the MFCC and LP-MFCC parameterisations, respec-

Table 1
Word Accuracy Rate (%) obtained with the MFCC parameterisation and five different back-ends: HMM, HMM-SVM linear, HMM-SVM RBF, DTAK linear and DTAK RBF for several noise conditions and the SI database.

		White noise				**F16**			
System	**Clean**	**12 dB**	**9 dB**	**6 dB**	**3 dB**	**12 dB**	**9 dB**	**6 dB**	**3 dB**
HMM	99.36	95.75	90.28	76.02	58.00	98.32	96.89	91.54	77.65
HMM-SVM linear	99.33	95.94	91.08	77.74	60.93	98.02	96.70	92.27	79.96
HMM-SVM RBF	99.42	95.96	91.27	77.94	61.16	98.39	97.19	92.79	80.86
DTAK linear	98.38	94.40	91.99	88.21	82.24	96.72	95.53	93.92	90.70
DTAK RBF	99.18	95.17	92.34	87.19	77.11	97.66	96.65	94.53	90.04

Table 2
Word Accuracy Rate (%) obtained with the LP-MFCC parameterisation and five different back-ends: HMM, HMM-SVM linear, HMM-SVM RBF, DTAK linear and DTAK RBF for several noise conditions and the SI database.

		White noise				**F16**			
System	**Clean**	**12 dB**	**9 dB**	**6 dB**	**3 dB**	**12 dB**	**9 dB**	**6 dB**	**3 dB**
HMM	99.52	95.81	90.67	79.69	61.66	98.22	96.79	92.98	83.16
HMM-SVM linear	99.50	95.64	90.93	81.04	65.05	98.12	96.62	92.84	83.83
HMM-SVM RBF	99.60	95.86	91.09	81.34	64.51	98.39	97.06	93.33	84.52
DTAK linear	98.71	95.33	92.63	87.84	80.67	96.92	95.92	94.17	91.07
DTAK RBF	99.18	95.98	93.85	89.87	83.59	97.87	97.00	95.55	92.86

tively, and the five classifiers tested: HMM, SVM-HMM linear, SVM-HMM RBF, DTAK linear and DTAK RBF.

As for the SI database, noise clearly degrades the performance of all the back-ends tested for the two parameterisations considered. It is worth noting that results show the same trend in both databases, even when they are of a different nature: SI was initially recorded in clean conditions with a high-quality microphone while SpeechDat was captured through the Public Switch Telephone Network. In addition, SpeechDat has fewer speech data than the SI database. However, as will be shown in section 6.5, SpeechDat will allow us to corroborate the conclusions extracted from the experimentation with the SI database.

Table 3
Word Accuracy Rate (%) obtained with the MFCC parameterisation and five different back-ends: HMM, HMM-SVM linear, HMM-SVM RBF, DTAK linear and DTAK RBF for several noise conditions and the subset of the SpeechDat database.

		White noise				**F16**			
System	**Clean**	**12 dB**	**9 dB**	**6 dB**	**3 dB**	**12 dB**	**9 dB**	**6 dB**	**3 dB**
HMM	99.39	93.44	87.19	73.33	54.90	97.48	95.19	89.27	75.35
HMM-SVM linear	99.24	93.06	87.07	73.50	55.92	97.61	95.29	89.84	77.39
HMM-SVM RBF	99.33	93.12	87.17	73.98	55.95	97.64	95.45	90.13	77.62
DTAK linear	98.60	92.23	88.09	82.46	75.10	96.21	93.89	91.63	87.39
DTAK RBF	98.92	92.26	88.00	82.58	73.63	96.46	94.62	91.81	87.45

Table 4
Word Accuracy Rate (%) obtained with the LP-MFCC parameterisation and five different back-ends: HMM, HMM-SVM linear, HMM-SVM RBF, DTAK linear and DTAK RBF for several noise conditions and the subset of the SpeechDat database.

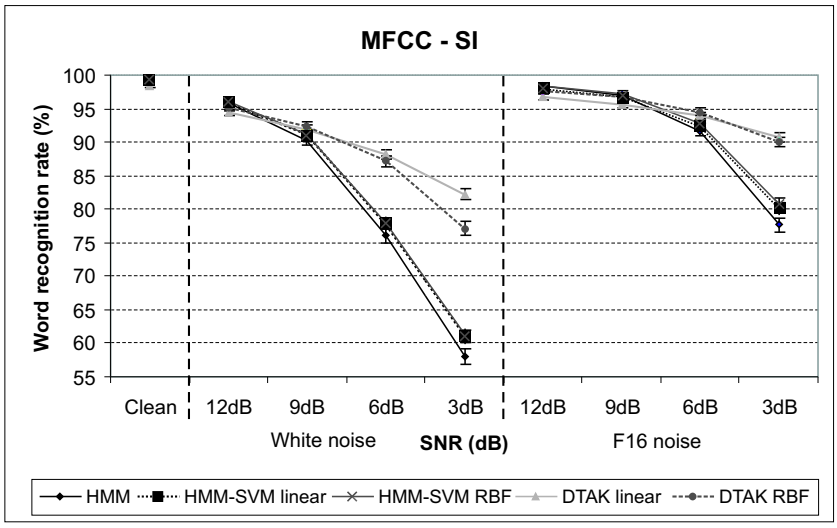
		White noise				**F16**			
System	**Clean**	**12 dB**	**9 dB**	**6 dB**	**3 dB**	**12 dB**	**9 dB**	**6 dB**	**3 dB**
HMM	99.62	94.96	89.55	77.16	56.40	97.96	96.27	93.26	82.30
HMM-SVM linear	99.52	94.52	89.14	78.25	58.95	98.02	96.50	93.12	84.12
HMM-SVM RBF	99.52	94.50	89.14	78.25	59.02	98.03	96.50	93.12	84.12
DTAK linear	98.66	93.44	90.13	84.62	77.77	96.82	95.32	93.15	89.71
DTAK RBF	98.89	92.71	89.18	83.03	73.35	96.91	95.70	92.74	87.71

6.5 Analysis of the results

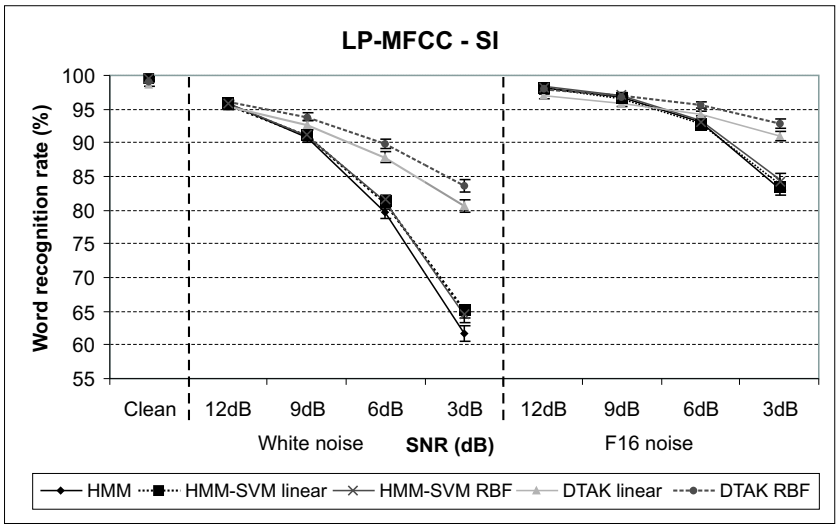
In the previous subsection, we have presented the performance of the two SVM-based systems using two isolated-Spanish digit recognition tasks, two parameterisations, two types of noise, four SNRs and two kernels. In this subsection, we present a detailed and comparative analysis of the results achieved in these experiments and the main conclusions drawn from them.

For a better display of the results, we have represented the recognition rates contained in Tables 1 and 2 in Figures 1 (a) and (b), respectively. These results correspond to the different back-ends tested in clean and noisy conditions (white and F16 noise) for the SI database and for the two parameterisations considered. We have also depicted the corresponding confidence intervals. Similarly, Figures 2 (a) and (b) show the recognition rates obtained with the SpeechDat database for MFCC and LP-MFCC, respectively, together with

their confidence intervals. In this case, these results correspond to Tables 3 and 4.



(a)

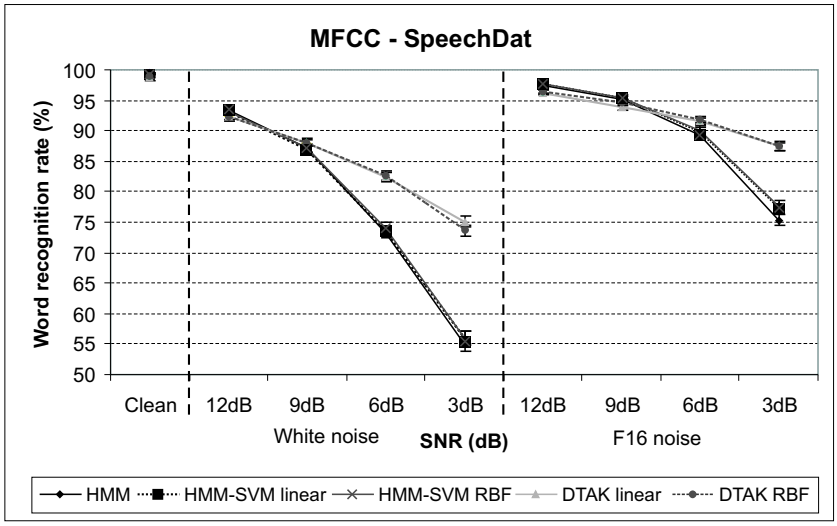


(b)

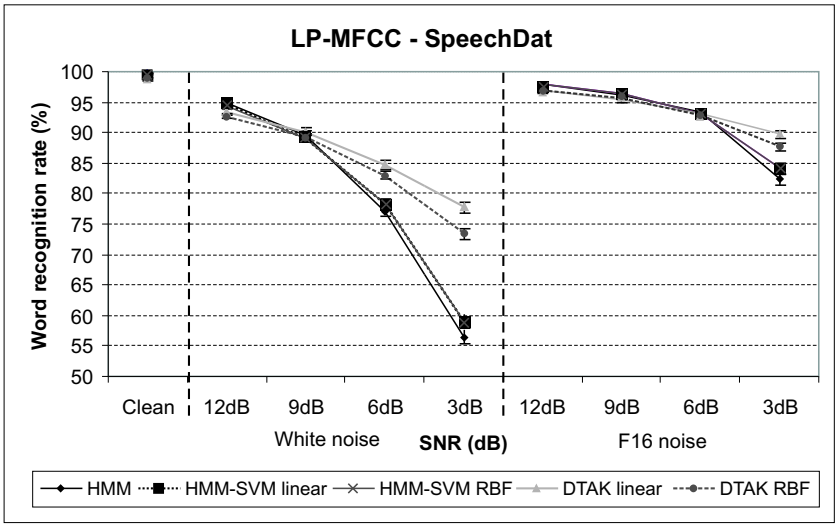
Fig. 1. Comparison of the performance of the different back-ends considered: HMM, HMM-SVM linear, HMM-SVM RBF, DTAK linear and DTAK SVM for the SI database (a) MFCC parameterisation; (b) LP-MFCC parameterisation.

6.5.1 Comparison between SVM kernels: linear and RBF

Regarding the use of linear or RBF kernels in hybrid systems, it can be observed in Figures 1 and 2 that the recognition rates obtained with both methods are very close in both databases considered.



(a)



(b)

Fig. 2. Comparison of the performance of the different back-ends considered: HMM, HMM-SVM linear, HMM-SVM RBF, DTAK linear and DTAK SVM for the SpeechDat database (a) MFCC parameterisation; (b) LP-MFCC parameterisation.

The choice between the RBF kernel or the linear kernel in the DTAK classifier is more difficult. The linear kernel outperforms the RBF kernel for some cases with the MFCC parameterisation (SI and SpeechDat databases) and with LP-MFCC (SpeechDat). However, in general, the differences are not statistically significant. On the contrary, the RBF kernel is superior for the LP-MFCC parameterisation and the SI database, which gives the highest recognition scores. In this case, the improvements are statistically significant except for the case of white noise at 12 dB.

In short, from these comparisons we can conclude that the differences between linear and RBF kernels are slight and depend on the database and front-end considered. It is worth noting that the accuracy of both kernels relies on the correct selection of the SVM parameters, which was performed by cross-validation over an independent set with a limited amount of data. We hypothesise that the RBF kernel is most sensitive than the linear kernel to the correct choice of these values and so it is indirectly more influenced by the size of the validation set. Finally, from a practical perspective, it should be taken into account that the RBF kernel is much more computationally cumbersome.

6.5.2 Comparison between parameterisations

When comparing the results in Tables 1 and 3 corresponding to the MFCC parameterisation and Tables 2 and 4 corresponding to the LP-MFCC front-end for the HMM based systems and the hybrid systems, we can conclude that on most occasions LP-MFCC outperforms MFCC, especially in low SNR conditions. Furthermore, as the noise conditions worsen, the performance improvement becomes more significant (for example, this improvement is statistically significant for the SpeechDat database contaminated by F16 noise at 3 and 6 dB for the HMM and hybrid systems). However, the differences are not statistically significant in the few cases in which MFCC is superior to LP-MFCC (for example, for the HMM system and F16 noise at 12 dB). In our opinion, the robustness of LP-MFCC is due to the LP spectrum analysis carried out as a part of the parameterisation process which acts as a smoothing step.

However, the improvements achieved for the LP-MFCC parameterisation lose statistical significance when a more robust back-end is used. This is the case for the DTAK SVM systems, in which both parameterisations have a similar performance.

6.5.3 Comparison between HMM and hybrid HMM-SVM systems

As can be observed in Figure 1, hybrid HMM-SVM classifiers do not produce any considerable improvement with respect to HMMs for the SI database. In fact, the improvements are statistically significant only for low SNRs (in particular, white and F16 noises at 3 dB). In Figure 2, the same behaviour can be observed for the SpeechDat database. In this case, performance differences between HMM and hybrid HMM-SVM systems are not statistically significant for all the conditions tested. All we can say is that results show a certain trend: hybrid systems are more robust than HMMs when working in very noisy conditions while their performance is very similar for clean conditions and high SNRs.

In our opinion, the explanation for these little improvements is that the hybrid

system inherits an HMM-based segmentation that sets a ceiling on its performance level. In other words, when HMM fails, the segmentation obtained is far from optimal, and thus, the SVMs are not able to overcome these errors.

6.5.4 Comparison between HMM and DTAK systems

For the SI database and the two parameterisations considered, the DTAK SVM system achieves excellent results, clearly superior to those achieved by the HMM system for low SNRs, incurring some performance losses for high SNRs (i.e. when the noise level is low). In particular, as can be observed in Figures 1 (a) and (b), the improvements of DTAK (linear and RBF) with respect to HMMs are statistically significant for white noise at 3, 6 and 9 dB and F16 noise at 3 and 6 dB. On the contrary, the HMM-based system significantly outperforms the DTAK systems in clean conditions and several cases of noise at 12 dB (for example, for F16 noise and the MFCC parameterisation). In the remaining conditions, which correspond to medium SNRs, the systems do not present statistically significant differences.

Figures 2 (a) and (b) show that similar conclusions can be drawn from the results obtained with the SpeechDat database.

In conclusion, it is now clear that SVMs exhibit a robust behaviour as we suspected. For a better illustration of this important property of DTAK systems, we have included Figure 3 that shows the relative error reduction achieved by DTAK systems (linear and RBF kernels) with respect to the HMM-based system for the SI database and the MFCC parameterisation in all the conditions tested (clean, white and F16 noises). Note that the other experiments (SI database with LP-MFCC and SD database with MFCC and LP-MFCC) follow the same trend.

As can be observed in this Figure, although the DTAK-based classifiers perform slightly worse than the HMM-based system for clean speech and high SNRs, they achieve a high relative error reduction with respect to the baseline at low SNRs. In fact, the advantage due to the DTAK algorithm increases as the noise conditions worsen. For example, for white noise at 3 dB, the relative error reduction with respect to the HMM system is around 136.5% for DTAK - linear and 83.49% for DTAK RBF.

These results indicate that DTAK-based systems are very effective in noisy scenarios. One possible explanation is that the SVM discrimination capabilities are more robust against additive noise than the HMM ones, so SVMs are able to classify better the resulting segments.

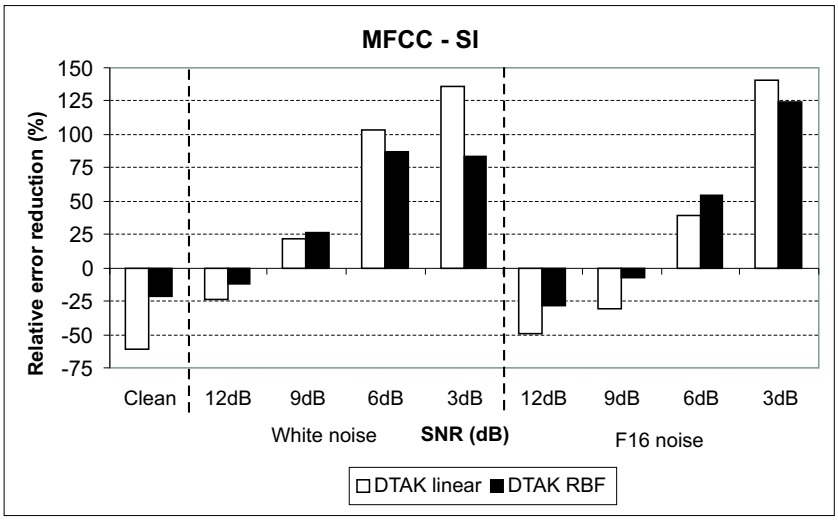


Fig. 3. Relative error reduction with respect to the HMM-based system for the MFCC parameterisation and SI database.

7 Conclusions and further work

The speech recognition problem is essentially a pattern classification problem. Although discriminative models are more suitable to deal with this type of problems than generative models, like HMMs, it is a fact that the core technology for most current ASR systems is HMM-based. This is mostly due to the HMMs' ability to cope with the variable time duration of speech utterances, an issue difficult to manage by ANN or SVMs.

In this paper, SVMs are proposed as a promising alternative to HMMs for several reasons: first, SVMs (as well as ANNs) are discriminative models, thus more appropriate for classification problems; second, due to the maximum margin criterium used for their training, they exhibit an excellent generalisation ability that makes them especially suitable to deal with noisy speech ; and third, some good solutions to tackle the variable time duration problem have recently been reported in the SVM framework.

We have compared two SVM-based approaches to speech recognition with the classical HMM system. The first SVM-based approach is a hybrid method: a segmentation generated by HMMs is used to provide the SVMs with a fixed-dimensional input. The second approach is a genuine SVM-based system that manages the variable input dimension through the DTA string kernel. The DTAK clearly outperforms the hybrid systems in moderate to highly noisy environments which leads us to think that the HMM segmentation is weighting down the performance of the SVM in the hybrid approach. Furthermore, DTAK achieves results clearly superior to the HMM system under the same circumstances. On the other hand, however, it incurs some performance losses

for clean speech or high SNRs.

In conclusion, we believe that SVMs should be considered as a promising paradigm for the development of robust speech recognition systems. The maximum margin solution provided by SVMs, responsible for their good generalisation properties, can be successfully applied to the speech recognition problem. However, it is still an issue in need of further investigation a comparison with more classical ANN hybrid systems.

We highlight two main future lines of research: first, improving the results for high SNRs and second, extending the system to the continuous speech recognition area. With respect to the former, some analysis should be performed to gain more insight into the behaviour of the DTAK algorithm: the optimisation process could be falling in local minima which could very likely be avoided by including some constraints in the search space.

Regarding the extension to continuous speech, several questions should be addressed, from the search for the more suitable types of acoustic units to the manner of obtaining compact SVM representations, but the more acute problem is to obtain an appropriate segmentation. In our opinion, some alternative segmentation techniques such as for example, [56], should be revisited because they could provide a better match with the abilities of the SVM and allow the independence of idiosyncratic HMMs.

Additionally, another line of research not in the scope of this paper but which should be considered in the mid-term, is overcoming the difficulties that SVM's algorithms have in effectively handling very large databases. However, there are already some published solutions on this subject ([37,38,36]).

8 Acknowledgements

The authors would like to thank F. Pérez-Cruz and A. Navia-Vázquez for their theoretical support on SVM related issues.

This paper is partially supported by the regional grant UC3M-TEC-05-059.

References

[1] H. Shimodaira, K. Noma, M. Nakai, S. Sagayama, Support vector machine with dynamic time-alignment kernel for speech recognition, in: Proceedings of Eurospeech, Aalborg, Denmark, 2001, pp. 1841–1844.

[2] H. Sakoe, R. Isotani, K. Yoshida, K. Iso, T. Watanabe, Speaker-independent word recognition using dynamic programming neural networks, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Glasgow, Scotland, 1989, pp. 439 – 442.

[3] K. Iso, T. Watanabe, Speaker-independent word recognition using a neural prediction model, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Alburquerque, New Mexico (USA), 1990, pp. 441–444.

[4] J. Tebelskis, A. Waibel, B. Petek, O. Schmidbauer, Continuous speech recognition using predictive neural networks, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, Canada, 1991, pp. 61–64.

[5] H. Bourlard, N. Morgan, Connectionist speech recognition: a hybrid approach, Boston: Kluwer Academic, Norwell, MA (USA), 1994.

[6] B. Schölkopf, A. Smola, Learning with kernels, MIT Press, Cambridge, MA (USA), 2002.

[7] V. Vapnik, Statistical learning theory, Wiley, Chichester, GB, 1998.

[8] V. Vapnik, The nature of statistical learning theory, Springer Verlag, New York, 1995.

[9] S. Fine, J. Navratil, R. Gopinath, A hybrid gmm/svm approach to speaker identification, in: Proceedings of the International Conference on Acoustics, Speech,and Signal Processing (ICASSP), Vol. 1, Salt Lake City, Utah (USA), 2001, pp. 417–420.

[10] Q. Le, S. Bengio, Client dependent GMM-SVM models for speaker verification, in: International Conference on Artificial Neural Networks, ICANN/ICONIP,Springer-Verlag, 2003, pp. 443–451.

[11] C. Ma, M. Randolph, J. Drish, A support vector machines-based rejection technique for speech recognition, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vol. 1, Salt Lake City, Utah (USA), 2001, pp. 381–384.

[12] J. García-Cabellos, C. Peláez-Moreno, A. Gallardo-Antolín, F. Pérez-Cruz, F. Díaz-de-María, SVM classifiers for ASR: a discusion about parameterization, in: Proceedings of EUSIPCO 2004, Wien, Austria, 2004, pp. 2067–2070.

[13] A. Ech-Cherif, M. Kohili, A. Benyettou, M. Benyettou, Lagrangian support vector machines for phoneme classification, in: Proceedings of the 9th International Conference on Neural Information Processing (ICONIP '02), Vol. 5, Singapore, 2002, pp. 2507–2511.

[14] C. Sekhar, W. Lee, K. Takeda, F. Itakura, Acoustic modelling of subword units using support vector machines, in: Workshop on spoken language processing, Mumbai, India, 2003.

[15] S. Gangashetty, C. Sekhar, B. Yegnanarayana, Combining evidence from multiple classifiers for recognition of consonant-vowel units of speech in multiple languages, in: Proceedings of the International Conference on Intelligent Sensing and Information Processing, Chennai, India, 2005, pp. 387–391.

[16] D. Martín-Iglesias, J. Bernal-Chaves, C. Peláez-Moreno, A. Gallardo-Antolín, F. Díaz-de-María, Nonlinear analyses and algorithms for speech processing, Vol. LNAI 3817 of Lecture Notes in Computer Science, Springer, 2005, Ch. A speech recognizer based on multiclass SVMs with HMM-guided segmentation, pp. 256–266.

[17] A. Ganapathiraju, J. Hamaker, J. Picone, Applications of support vector machines to speech recognition, IEEE Transactions on Signal Processing 52 (2004) 2348–2355.

[18] N. Thubthong, B. Kijsirikul, Support vector machines for Thai phoneme recognition, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 9 (2001) 803–813.

[19] P. Clarkson, P. Moreno, On the use of support vector machines for phonetic classification, IEEE International Conference on Acoustics, Speech and Signal Processing 2 (1999) 585–588.

[20] E. Trentin, M. Gori, A survey of hybrid ANN/HMM models for automatic speech recognition, Neurocomputing 37 (2001) 91–126.

[21] T. Robinson, M. Hochberg, S. Renals, Automatic speech and speaker recognition - advanced topics, Kluwer Academic Publishers, 1995, Ch. The Use of Recurrent Neural Networks in Continuous Speech Recognition (Chapter 19), pp. 159–184.

[22] W. Reichl, G. Ruske, A hybrid RBF-HMM system for continuous speech recognition, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP), Detroit, MI (USA), 1995, pp. 3335–3338.

[23] J. Stadermann, G. Rigoll, A hybrid SVM/HMM acoustic modeling approach to automatic speech recognition, in: Proceedings of the International Conference on Spoken Language Processing (ICSLP), Jeju Island, Korea, 2004, pp. 661–664.

[24] Y. Bengio, Neural networks for speech and sequence recognition, London International Thomson Computer Press, London, UK, 1995.

[25] A. Ganapathiraju, J. Hamaker, J. Picone, Hybrid SVM/HMM architectures for speech recognition, in: Proceedings of the 2000 Speech Transcription Workshop, Vol. 4, Maryland (USA), 2000, pp. 504–507.

[26] A. Ganapathiraju, Support vector machines for speech recognition, PhD Thesis, Mississipi State University (2002).

[27] J. Hamaker, J. Picone, A. Ganapathiraju, A sparse modeling approach to speech recognition based on relevance vector machines, in: Proceedings of the International Conference of Spoken Language Processing, Vol. 2, Denver, Colorado (USA), 2002, pp. 1001–1004.

[28] J. Hamaker, J. Picone, Advances in speech recognition using sparse Bayesian methods, unpublished (January 2003).

[29] H. Jiang, X. Li, C. Liu, Large margin hidden markov models for speech recognition, IEEE Transactions on Audio, Speech and Language Processing 14 (2006) 1584–1595.

[30] H. Shimodaira, K. Noma, M. Nakai, Advances in neural information processing systems 14, Vol. 2, MIT Press, Cambridge, MA (USA), 2002, Ch. Dynamic Time-Alignment Kernel in Support Vector Machine, pp. 921–928.

[31] N. Smith, M. Gales, Using SVMs and discriminative models for speech recognition, in: IEEE International Conference on Acoustics, Speech and Signal Processing, Vol. 1, Orlando, Florida (USA), 2002, pp. 77–80.

[32] N. Smith, M. Gales, Advances in neural information processing systems 14, Vol. 14, MIT Press, Cambridge, MA (USA), 2002, Ch. Speech recognition using SVMs, pp. 1197–1204.

[33] N. Smith, M. Niranjan, Data-dependent kernels in SVM classification of speech patterns, in: Proceedings of the International Conference on Spoken Language Processing (ICSLP), Vol. 1, Beijing, China, 2000, pp. 297–300.

[34] T. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, Tech. rep., Dept. of Computer Science, Univ. of California (1998).
URL `citeseer.ist.psu.edu/jaakkola98exploiting.html`

[35] V. Wan, S. Renals, Support vector machine speaker verification methodology, in: International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vol. 2, Hong Kong, 2003, pp. 221–224.

[36] A. Navia-Vázquez, F. Pérez-Cruz, A. Artés-Rodríguez, A. Figueiras-Vidal, Weighted least squares training of support vector classifiers leading to compact and adaptive schemes, IEEE Trans. Neural Networks 12 (5) (2001) 1047–1059.

[37] C. Burges, Simplified support vector decision rules, in: Proceedings of the Thirteenth International Conference on Machine Learning, Bari, Italy, 1996, pp. 71–77.

[38] E. Osuna, R. Freund, F. Girosi, An improved training algorithm for support vector machines, in: IEEE Workshop on Neural Networks for Signal Processing, Amelia Island, Florida (USA), 1997, pp. 276–285.

[39] R. Collobert, SVMTorch: a support vector machine for large-scale regression and classification problems, IDIAP.
URL `www.idiap.ch/learning/SVMTorch.html`

[40] J. Fürnkranz, Round robin classification, Journal of Machine Learning Research 2 (2002) 721–747.

[41] C. Hsu, C. Lin, A comparison of methods for multi-class support vector machines, IEEE Transactions on Neural Networks 13 (2) (2002) 415–425.

[42] T. Joachims, Advances in kernel methods— support vector learning, in: B. Schölkopf, C. J. C. Burges, A. J. Smola (Eds.), Making Large Scale SVM Learning Practical, MIT Press, Cambridge, (MA), 1999, pp. 169–184.

[43] C. Chih-Chung, L. Chih-Jen, LIBSVM: a library for support vector machines (2004).
URL `http://www.csie.ntu.edu.tw/ cjlin/libsvm/`

[44] E. Allwein, R. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, Journal of Machine Learning Research 1 (2000) 113–141.

[45] J. Weston, C. Watkins, Multi-class support vector machines, in: M. Verleysen (Ed.), Proc. of the European Symposium on Artificial Neural Networks, 1999.

[46] K. Crammer, Y. Singer, On the algorithmic implementation of multiclass kernel-based vector machines, Journal of Machine Learning Research 2 (2001) 265–292.

[47] J. C. Platt, Advances in large margin classifiers, MIT Press, 1999, Ch. Probabilities for SV Machines, pp. 61–74.

[48] H. T. Lin, C. J. Lin, R. C. Weng, A note on Platt's probabilistic outputs for support vector machines, Tech. rep., Department of computer science and information engineering, National Taiwan University (2003).

[49] T. F. Wu, C. J. Lin, R. C. Weng, Probability estimates for multi-class classification by pairwise coupling, The Journal of Machine Learning Research 5 (2004) 975–1005.

[50] L. R. Rabiner, A. Rosenberg, S. Levinson, Considerations in dynamic time warping algorithms for discrete word recognition, IEEE Transactions on Acoustics, Speech and Signal Processing 26 (6) (1978) 575–582.

[51] A. Moreno, SpeechDat documentation [cd-rom], ver 1 (1998).

[52] A. Varga, J. Steenneken, M. Tolimson, D. Jones, The NOISEX-92 study on the effect of additive noise on automatic speech recognition, Tech. rep., DRA Speech Research Unit (1992).

[53] J. Vicente-Peña, A. Gallardo-Antolín, C. Peláez-Moreno, F. D. de María, Band-pass filtering of the time sequences of spectral parameters for robust wireless speech recognition, Speech Communication 48 (10) (2006) 1379–1398.

[54] S. Young, HTK-Hidden Markov Model toolkit (ver 2.1), Cambridge University (1995).

[55] N. A. Weiss, M. J. Hasset, Introductory statistics, 3rd Edition, Addison-Wesley, Reading, MA, 1993.

[56] J. R. Glass, A probabilistic framework for segment-based speech recognition, Computer Speech and Language 17 (2003) 137–152.