Multiclass SVM-Based Isolated-Digit Recognition using a HMM-Guided Segmentation

Jorge Bernal-Chaves, Carmen Peláez-Moreno, Ascensión Gallardo-Antolín, and Fernando Díaz-de-María

Signal Theory and Communications Department, EPS-Universidad Carlos III de Madrid, Spain
EPS-Universidad Carlos III de Madrid
Avda. de la Universidad, 30, 28911-Leganés (Madrid), SPAIN

Abstract. Automatic Speech Recognition (ASR) is essentially a problem of pattern classification, however, the time dimension of the speech signal has prevented to pose ASR as a simple static classification problem. Support Vector Machine (SVM) classifiers could provide an appropriate solution, since they are very well adapted to high-dimensional classification problems. Nevertheless, the use of SVMs for ASR is by no means straightforward, because SVM classifiers are well developed for binary problems but not so for the multiclass case. In this paper we compare two approaches to implement the multiclass SVM from binary SVMs (1-vs-all and 1-vs-1) for a specific ASR task. We show that the 1-vs-all multiclass SVM clearly outperforms the conventional HMM-based ASR system (the largest improvement, 18.23 %, is achieved for speech corrupted with white noise).

1 Introduction

Hidden Markov Models (HMMs) are, undoubtedly, the most employed core technique for Automatic Speech Recognition (ASR). During the last decades, research in HMMs for ASR has brought about significant advances and, consequently, the HMMs are currently accurately tuned for this application. 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 ([10], [7], [13], [2] and [3] are some examples). Some of them tackled 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.

Speech recognition is essentially a problem of pattern classification, but the high dimensionality of the sequences of speech feature vectors has prevented researchers to propose a straightforward classification scheme for ASR. Support Vector Machines (SVMs) are state-of-the-art tools for linear and nonlinear knowledge discovery [11], [14]. Being based on the maximum margin classifier, SVMs are able to outperform classical classifiers in the presence of high dimensional data even when working with nonlinear machines.

Some researchers have already proposed different approaches to speech recognition aiming at taking advantage of this type of classifiers. Among them, [4], [5] and [12] use different approaches to perform the recognition of short duration units, like isolated phoneme or letter classification. In [4], the authors carry out a length adaptation based on the triphone model approach. In [5] and [12], a normalizing kernel is used to achieve the adaptation. Both cases show the superior discrimination ability of SVMs. Moreover, in [5], a hybrid approach based on HMMs has been proposed and tested in a CSR (Continuous Speech Recognition) task.

Nevertheless, the use of SVMs for ASR is by no means straightforward. We see two main problems: dimensional normalization and multiclass SVMs. Both problems are described in the next paragraphs.

Typical speech analysis generates sequences of feature vectors of variable lengths (due to the different acoustic units durations and the constant frame rate analysis commonly employed), while SVM classifiers require a fixed-dimension input. A recently published work offers an effective solution to this problem [20]. Specifically, the non-uniform distribution of analysis instants provided by the internal states of an HMM with a fixed number of states and a Viterbi decoder is used for dimensional normalization.

With respect to the second problem, the extension of the SVM formulation to the multiclass case is very hard. We have decided to test a couple of approaches to multiclass classification from binary SVMs; namely: 1-vs-1 and 1-vs-all. In this paper we compare both approaches for an specific ASR task.

This paper is organized as follows. In next section, we describe the dimensional normalization problem. Section 3 summarizes the SVM training and multiclass implementation procedures. In Section 4 we present the experimental framework and the results obtained. Finally, some conclusions and further work close the paper.

2 Feature Extraction and Dimensional Normalization

Since the speech signal is quasi-stationary, speech analysis must be performed on a short-term basis. Typically, the speech signal is divided into a number of overlapping time windows and a speech feature vector is computed to represent each of these frames. The size of the analysis window, w_a, is usually of 20-30 ms. The frame period, T_f, (the time interval between two consecutive analysis windows) is set to a value between 10 and 15 ms. Habitually, $w_a = K\,T_f$, where K is called the overlapping factor.

With respect to the feature vectors themselves, for each analysis window, twelve Mel-Frequency Cepstral Coefficients (MFCC) are obtained using a mel-scaled filterbank with 40 channels. Then, the log-energy, the twelve delta-cepstral coefficients and the delta-log energy are appended, making a total vector dimension of 26.

Typically, the values of w_a and T_f are kept constant for every utterance that, on the other hand, exhibits a different time duration. Consequently, the speech analysis generates sequences of feature vectors of variable length. As we have already mentioned, a normalization of these lengths is required to use SVM classifiers.

In a previous work [20] we proposed and compared three procedures to perform this dimensional normalization. Two of them were very straightforward approaches consisting on adjusting either the analysis window size or the frame period to obtain a fixed number of time analysis instants. The third one, more sophisticated, used and HMM-based segmentation to select the time analysis instants. The next subsection describes this last method, that is the one selected for the experiments conducted in this work

2. 1 **Non-uniform distribution of analysis instants**

An appropriate selection of the time instants at which the speech signal is analysed can presumably improve the classification results.

To determine the appropriate analysis instants, we propose to use the implicit information in the segmentation made by HMM, i.e., to consider those instants at which state transitions occur (very likely related to those at which the changes of the speech spectra happen).

This HMM-guided parameterization procedure consists of two main stages. The first stage is a HMM classifier (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 continuous density HMMs with three Gaussian mixtures per state. Each HMM represents a whole-word and consists of N_s states with the topology shown in **Figure 1**. These models have been trained using only the training set of the speech database and the conventional parameterization module used for the baseline experiments. In particular, the speech parameters consists of 12 MFCC, the log-energy, 12 delta-MFCC and the delta-log energy, extracted using a frame period of 10 ms and an analysis Hamming window of 25 ms.

As mentioned before, these acoustic models are used to generate alignments at state level for each utterance 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 obtained segmentation may not be correct, even when the utterance is properly recognized by the HMM-based system. Segmentation errors may produce some degradation in the performance of the whole system, however, for our task, the results obtained show that the segmentation is accurate enough. Anyway, it is necessary to consider this issue for further research.

In the second stage, the feature vectors are extracted at the time instants derived from the HMM-guided segmentation. In particular, a 25 ms analysis window is subsequently located at these time instants. In this way, the number of feature vectors per utterance used as the SVM input turns out to be equal to the number of states (N_s), determined by the HMM topology. In our case, the number of states was fixed to 17 (the same number of states we use for HMM-based recognition).

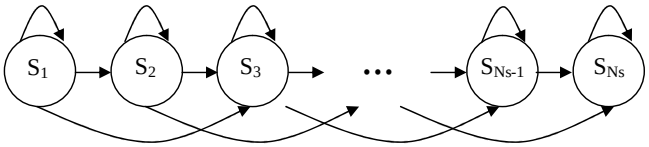

Fig. 1. HMM topology

3 SVM training and classification

3. 1 SVM fundamentals

Given a labelled training data set $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ ($\mathbf{x}_i \in \Re^d$ and $y_i \in \{\pm 1\}$, where $\mathbf{x}_i$ is the input vector and y_i is its corresponding label), an SVM solves the following equation

$$\min_{\mathbf{w},b,\xi_i} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \right\} \tag{1}$$

subject to

$$y_i\left(\mathbf{w}^T \phi(\mathbf{x}_i) + b\right) \geq \varepsilon - \xi_i \tag{2}$$
$$\xi_i \geq 0$$

Where $\mathbf{w}$ and b define the linear classifier in the feature space and $\phi(\cdot)$ is the non-linear transformation to the feature space ($\mathbf{x}_i \in \Re^d \rightarrow \phi(\mathbf{x}_i) \in \Re^H$, $d \leq H$). Unless $\phi(\mathbf{x}) = \mathbf{x}$, the solution in the input space will be nonlinear. The SVM minimizes the norm of $\mathbf{w}$ subject to correct classification of all the samples (for every $\xi_i = 0$). If the training samples are not separable, the slack variables, ξ_i, corresponding to the samples that can not be correctly classified will become nonzero and will be penalised in the objective function. The SVM is usually solved introducing the restrictions in the minimizing functional using Lagrange multipliers, leading to the maximization of the Wolfe dual:

$$L_d = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_{ij} \alpha_i \alpha_j \phi^{\hat{O}}(\mathbf{x}_i)\phi(\mathbf{x}_j) \tag{3}$$

with respect to α_i and subject to $\sum_{i=1}^{n} \alpha_i = 0$ and $0 \leq \alpha_i \leq C$. This procedure can be solved using quadratic programming (QP) schemes. To solve Wolfe dual, we do not need to know the nonlinear mapping $\phi(\cdot)$, but only its Reproducing Kernel in Hilbert

Space (RKHS) $\kappa(\mathbf{x}_i,\mathbf{x}_j) = \phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j)$. The value of $\mathbf{w}$ and b can be recovered from the Lagrange multipliers α_i, that are associated with the first linear restriction in the SVM formulation.

3.1 Multiclass SVM

For ASR, the input vectors, $\mathbf{x}_i$, will be the concatenation of a fixed number (fixed input dimension) of feature vectors that will be obtained using the dimensionality normalization described in section 2.

Besides, an extension to use a primarily binary classification method as the SVM to cope with the multiclass problem is needed and several approaches can be used. Basically, we can either solve several binary classification problems or we can directly try to solve the multiclass classification by extending the SVM binary formulation. However, this last option is very hard to implement and train and therefore in this paper we have decided to compare two different variations of the former.

There are typically two approaches to solve non-binary classification problems with the standard binary SVM. First, by comparing each class against all the rest (1-vs-all). Second, by confronting each class against all the other classes separately (1-vs-1) [1].

In particular, we have compared an variation of the 1-vs-all approach with the 1-vs-1 solution. In the former, probability-like outputs are obtained using the implementation in [17]. Afterwards the outputs of the binary 1-vs-all classifiers are compared and the most probable one among the ones showing a positive output is chosen (positive meaning that the binary classifiers has selected the 'one' against the 'rest'). For the 1-vs-1 alternative we have used the implementation described in [18] where error correcting codes are used to compare the outputs of the classifiers.

4 Experimental Results

4.1 Baseline System and Database

We use a database consisting of 72 speakers and 11 utterances per speaker for the 10 Spanish digits. This database was recorded at 8 kHz in clean conditions. Since the database is limited to achieve reliable speaker-independent results, we have used a 9-fold cross validation to artificially extend it. Specifically, we have split each database into 9 balanced groups; 8 of them for training and the remaining one for testing, averaging the results afterwards. In summary, we use a total of 7,920 words for testing our systems.

The baseline HMM-based ASR system is an isolated-word, speaker-independent system developed using the HTK package [15]. Left-to-right HMM with continuous observation densities are used. Each of the whole-digit models contains a different number of states (which depends on the number of allophones in the phonetic transcription of each digit) and three Gaussian mixtures per state.

For the baseline experiment with the HMM classifier, a Hamming window with a width of 30 ms was used and the feature vectors (consisting of 12 MFCC, the log-energy, 12 delta-MFCC and the delta-log energy) were extracted once every 10 ms.

For both SVM classifiers (1-vs-all and 1-vs-1), we have used a 17 state HMM to produce the sampling instants in which the speech signal is analysed. Thus, in this case we use 17 feature vectors (each one consisting of 12 MFCC, the log-energy, 12 delta-MFCC and the delta-log energy) per utterance as the SVM input.

4.2 Experiments and Results

We have tested our systems in clean conditions and in 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 [19] and added to the speech signal to achieve a signal-to-noise ratio of 12 dB. As we have used clean speech for estimating the acoustic models (in both, HMM and SVM-based recognizers), the noises are only added for testing the recognition performance.

Word recognition rates obtained with both alternatives of the multiclass SVM-based system and in comparison to those achieved by the HMM-based system are shown on **Table 1**. As it can be observed, in clean conditions SVM classifiers perform slightly better than the HMM-based system (99.72 % recognition rate vs. 99.67 %).

For speech corrupted with white noise, SVMs systems outperform HMM results. In particular, the 1-vs-all approach produces the best results with an improvement of 18.23 % over the HMM-based system.

For F16 noise, recognition rates obtained with the 1-vs-1 SVM recognizer are slightly worse than those achieved with the conventional HMM-based system (48.09 % vs. 48.37 %). However, the 1-vs-all SVM classifier clearly outperforms the HMM-based system (50.75 % vs. 48.37 %, i.e., an improvement of 4.92 %).

5 Conclusions and further work

In this paper, we have proposed two different approaches to a multiclass SVM classifier (1-vs-all and 1-vs-1) with application to a specific ASR task. Experimental results have shown that recognition rates obtained with SVM-based systems are very close to that achieved by a conventional HMM-based ASR system in clean conditions. However, for noisy conditions, the 1-vs-all SVM-based classifier outperforms both, the 1-vs-1 SVM approach and the baseline HMM system. From our point of view, these results are very encouraging since HMM-based systems have been accurately tuned during the last three decades for automatic speech recognition.

With respect to the further work, we consider several lines: First of all, to tune the training parameters of the SVM (we have used default ones). Second, to explore alternative parameterizations (we have used a parameterization specially designed for a back-end based on HMMs). And third, we expect to extend the SVM framework for ASR by using string kernels, which has been used with success for protein [8] and

text [9] classification, could be easily extended to speech processing provided we define a similarity measure for voice utterances.

Table 1. Recognition results obtained with the two proposed hybrid HMM-SVM-based classifiers for two types of noises (white and F16). Results obtained with the conventional HMM-based ASR system are presented as well

	Clean	White (SNR=12 dB)	F16 (SNR=12 dB)
HMM-based ASR	99.67 %	33.36 %	48.37 %
Hybrid HMM-SVM ASR system (1-vs-all)	99.72 %	39.44 %	50.75 %
Hybrid HMM-SVM ASR system (1-vs-1)	99.72 %	37.07 %	48.09 %

References

1. Allwein, E. L., Schapire, R. E., and Singer, Y, "Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers," Journal of Machine Learning Research, 1:113-141. 2000.
2. Bengio, Yoshua, "Neural networks for speech and sequence recognition", London International Thomson Computer Press , 1995.
3. Bourlard, Hervé A. and Morgan, Nelson, "Connectionist speech recognition: a hybrid approach", Boston: Kluwer Academic, 1994.
4. Clarkson, P.; Moreno, P.J, "On the use of support vector machines for phonetic classification", IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 2 , pp.585 –588, 1999.
5. Ganapathiraju, A., "Support vector machines for speech recognition" PhD Thesis, Mississipi State University, 2002.
6. Hermansky, H., Morgan, N., "RASTA processing of speech", "IEEE Trans. On Speech and Audio Processing", vol. 2, no. 4, pp. 587-589, Oct. 1994.
7. Iso, K. and Watanabe, T., "Speaker-Independent Word Recognition using a Neural Prediction Model", Proc. ICASSP-90, pp. 441-444; Albuquerque, New México, USA, 1990.

8. Leslie. C., Eskin, E., Weston, J., and Noble, W. S., “Mismatch String Kernels for SVM Protein Classification,” in Advances in Neural Information Processing Systems 15, Editors S. Becker and S. Thrun and K. Obermayer, 2002.
9. 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, 2002.
10. Sakoe, H., Isotani, R., Yoshida, K., Iso, K., Watanabe, T., “Speaker-Independent Word Recognition using Dynamic Programming Neural Networks”; Proc. ICASSP-89, pp. 29-32; Glasgow, Scotland; 1989.
11. Schölkopf, B. and Smola, A., “Learning with kernels”, M.I.T. Press, 2001.
12. Smith, N.D., Gales, M.J.F., “Using SVMs and discriminative models for speech recognition”, IEEE International Conference on Acoustics, Speech, and Signal Processing, 2002.
13. Tebelskis, J., Waibel A., Petek, B., and Schmidbauer, O., “Continuous Speech Recognition using Predictive Neural Networks”, Proc. ICASSP-91, pp. 61-64; Toronto, Canada; 1991.
14. Vapnik, V., “Statistical Learning Theory”, Wiley, 1998.
15. Young, S. et al., “HTK-Hidden Markov Model Toolkit (ver 2.1)”, Cambridge University, 1995.
16. Pérez-Cruz, F. and Artés-Rodríguez, A, “Puncturing Multi-Class Support Vector Machines” Proceedings of the ICANN'02. August, 2002.
17. Chih-Chung, Ch., Chih-Jen, L., “LIBSVM: a library for support vector machines”, [on-line] http://www.csie.ntu.edu.tw/~cjlin/libsvm/, August, 2004.
18. T.-K. Huang, R. C. Weng, and C.-J. Lin. “A Generalized Bradley-Terry Model: From Group Competition to Individual Skill.”,[on-line] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/ July 2004.
19. A.P. Varga, J. M. Steenneken, M. Tomlinson y D. Jones. “The NOISEX-92 Study on the Effect of Additive Noise on Automatic Speech Recognition”, Tech. Rep. DRA Speech Res. Unit. Malvern, Worcestershire, U. K. 1992.
20. J.M. García-Cabellos, C. Peláez-Moreno, A. Gallardo-Antolín, F. Pérez-Cruz, F. Díaz-de-María, “SVM Classifiers for ASR: A Discusión about Parameterization”, Proceedings of EUSIPCO 2004, pp. 2067-2070, 2004