SVM CLASSIFIERS FOR ASR: A DISCUSSION ABOUT PARAMETERIZATION

José Miguel García-Cabellos, Carmen Peláez-Moreno, Ascensión Gallardo-Antolín, Fernando Pérez-Cruz, 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-dimension classification problems. Nevertheless, the use of SVMs for ASR is by no means straightforward, because SVM classifiers require a fixed-dimension input. In this paper we propose and compare three alternatives for adapting the parameterization to the fixed-input dimension required by SVMs. We show that SVM classifiers outperforms the conventional HMM-based ASR system, when the speech signal is parameterised at properly selected instants.

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 ([12], [9], [15], [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 [13], [16]. 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, [6], [7] and [14] use different approaches to perform the recognition of short duration units, like isolated phoneme or letter classification. In [6], the authors carry out a length adaptation based on the triphone model approach. In [7] and [14], a normalizing kernel is used to achieve the adaptation. Both cases show the superior discrimination ability of SVMs. Moreover, in [7], 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. In fact, 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. In this paper we propose and compare three ways of adapting the parameterization to the fixed-input dimension required by SVMs. The first is based on adjusting the duration of the analysis time window. The second adapts the analysis frame rate. The third one uses the non-uniform distribution of analysis instants provided by the internal states of an HMM and a Viterbi decoder.

This paper is organized as follows. In next section, we describe the proposed approaches for the design of a fixed-dimension parameterization module. Section 3 summarizes the SVM training and classification 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 NORMALIZATION

Since the speech signal is not stationary, speech analysis must be performed on a short-term basis, in which the signal is assumed to be quasi-stationary. Typically, a speech signal is divided into a number of overlapping temporal windows (typically, Hamming) and a speech feature vector is computed to represent each of these temporal frames. The size of the analysis window, w_s, is usually of 20-30 ms. The frame period, f_p, (the interval between two consecutive feature vectors) is set to a value between 10 and 15 ms. The selected values for these parameters (w_s and f_p) in our particular approach will be discussed later.

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_s and f_p are kept constant for every utterance that, on the other hand, presents a different time duration. Consequently, the speech analysis generates sequences of feature vectors of variable lengths. As we have already mentioned, a normalization of these lengths is required to use SVM classifiers. In next paragraphs, we propose three different alternatives for solving this problem.

2.1 Variable window size

As in the traditional parameterization procedures, in this case, the window size is chosen to be proportional to the frame period (i.e. $w_s = K f_p$), with K (the overlapping factor) being constant for all utterances. Nevertheless, the value of w_s for every utterance is selected in such a way that the number of feature vectors results the same. Next, f_p is computed as w_s / K as illustrated in Figure 1.

Using this method, we are able to provide the SVM with sequences of feature vectors of the same length. However, when the value w_s is too large, the analysed speech segment does not meet the required stationarity properties and, therefore, some relevant details of the speech signal are missing.

2.2 Fixed window size

To overcome the above mentioned weakness, we propose to keep the value of the window size constant for every utterance. Our intention is to work with an analysis window length (30 ms) more consistent with the hypothesis of quasi-stationarity, avoiding to some extent the smoothening effect due to longer windows. However, for obtaining a fixed input vector dimension, we need to dynamically select the frame period (or the overlapping factor, K) for each speech utterance. Figure 1 b) shows an extreme example in which analysis windows are no longer overlapped. Therefore, again, some information is missing for long speech utterances.

It is important to notice, however, that through delta parameters (which consider two previous and posterior frames) some information of the surrounding of the current analysis window is included in the feature vectors.

2.3 Non-uniform distribution of analysis instants

In the two previous procedures the speech feature vectors were produced without any consideration about the information (or lack of information) that speech analysis segments were providing. Obviously, it would be better if we could use those utterance segments in which the signal is changing, that is, those segments carrying a bigger amount of information.

To determine more appropriate analysis instants, we propose to use the information contained in the HMM segmentation, i.e., to consider those instants in which state transitions are produced. A fixed size window is subsequently located in those positions as in the previous case (see Section 2.2). Note, however that, with this information-driven procedures we have eliminated the arbitrariness in the process of selecting the analysis instants.

3. SVM TRAINING AND CLASSIFICATION

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

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

subject to

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

$$\xi_i \geq 0$$

Where $\mathbf{w}$ and b define the linear classifier in the feature space and $\boldsymbol{\phi}(\cdot)$ is the non-linear transformation to the feature space ($\mathbf{x}_i \in \Re^d \rightarrow \boldsymbol{\phi}(\mathbf{x}_i) \in \Re^H$, $d \leq H$). Unless $\boldsymbol{\phi}(\mathbf{x}) = \mathbf{x}$, 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, $\xi_{I,}$ 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:

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

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 $\boldsymbol{\phi}(\cdot)$, but only its Reproducing Kernel in Hilbert Space (RKHS) $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \boldsymbol{\phi}^T(\mathbf{x}_i)\boldsymbol{\phi}(\mathbf{x}_j)$. 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.

For ASR, the input vectors, $\mathbf{x}_i$, will be the concatenation of a fixed number (fixed input dimension) of feature vectors. Furthermore, the classification problem should be formulated as a binary classification. 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]. The decisions are usually taken using a majority vote. It is still unclear which one is the most powerful technique and our experience with 1-vs-all techniques suggests that it is an advisable approach for this problem.

4. EXPERIMENTAL RESULTS

4.1 Baseline System and Database

The baseline is an isolated-word, speaker-independent HMM-based ASR system developed using the HTK package [17]. 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.

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.

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. In this case, both, the window size and the frame period were kept constant for parameterizating all utterances. The average recognition rate achieved by the HMM system was 99,67%. In other terms, only 25 errors over the 7,920 tested words.

4.2 Experiments and Results

Table 1 shows the word recognition rates achieved using the three methods described in Section 2. In the variable window size and in the fixed window size methods and SVM classifiers (always 1 vs. all) we have made an heuristic search for the best number of feature vectors per utterance. In both cases the optimal number we found was 13, rendering the variable window size method better results than the fixed window method (98.38 % recognition rate vs. 97.03 %), but slightly worse than that by the HMM-based system (99.67 %).

For the third method proposed 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 per utterance as the SVM input. The results obtained are the best ones, even better than those obtained with the HMM-based ASR system (99.89 % vs. 99.67 %), reaching only 9 errors over 7,920 tested words.

5. CONCLUSIONS AND FURTHER WORK

The performance of the three approaches for the SVM-based ASR system proposed in this paper is very close to that achieved by a conventional HMM-based ASR system. From our point of view, this result is very appealing since HMM-based systems has been tuned during the last three decades for this task.

In particular, the SVM-based system with the fixed-dimension parameterization based on a non-uniform distribution of sampling instants outperforms the conventional HMM-based ASR system. We even expect to improve results investigating if there exist an optimum number of HMM states. We also intend to search for the best SVM kernel parameters .

Obviously, though this method needs to be refined, we expect substantial improvements using a smarter parameterization procedure. We intend to work on a more sofisticated procedure to achieve the fixed-dimension input. Since the human auditory system is relatively insensitive to slowly varying stimuli [8], we propose to focus the spectral sampling on the time instants corresponding to the sharpest transitions in the spectral domain. Specifically, we propose to distribute the sampling instants in each utterance according to the derivative of the spectral features, instead of the sampling instants provided by the internal states of an HMM.

On the other hand, we expect to extend the SVM framework for ASR along two directions: string kernels and kernel target alignment. The first one, which has been used with success for protein [10] and text [11] classification, could be easily extended to speech processing provided we define a similarity measure for voice utterances. The second approach is a subtle transformation of the kernel matrix to tune its entries to the labels of the given problem [5], [4], significantly improving the performance of the machine obtained. This last approach is mainly described for *transductive learning* [16] in which the whole test set needs to be known a priori, which is its most severe limitation.

	Variable window size	Fixed window size	Fixed window size with HMM segmentation	HMM based ASR
Word Recognition Rate %	98.38 %	97.03 %	99.89 %	99.67 %

Table 1. Recognition results using a SVM-based classifier and the three proposed time–normalized parameterizations: variable window size, fixed window size and fixed window size with HMM segmentation. The result obtained with the conventional HMM-based ASR system is presented as well.

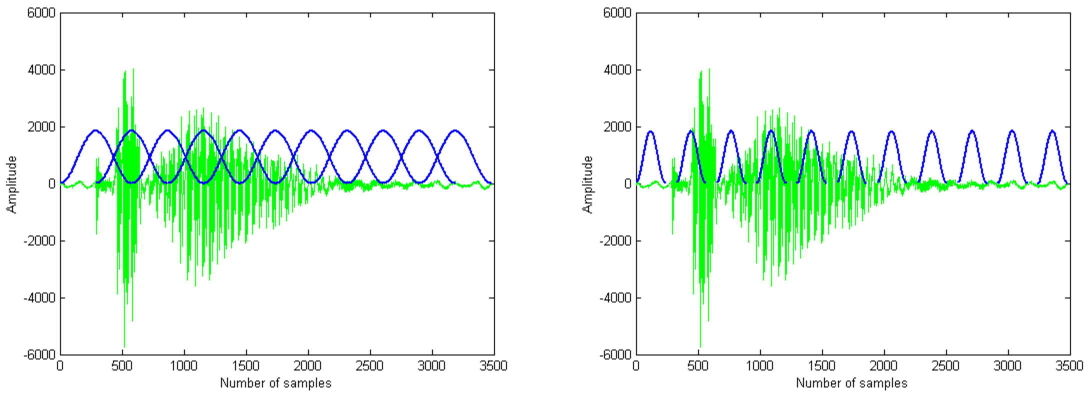

Figure 1. Two different approaches for normalizing the length of feature sequences: variable (figure 1a) and fixed (figure 1b) window size

6. 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] Bousquet, O., and Herrmann, D. J. L., "On the Complexity of Learning the Kernel Matrix," in Advances in Neural Information Processing Systems 15, Editors S. Becker and S. Thrun and K. Obermayer, 2002.

[5] Cristianini, N., Shawe-Taylor, J., Elisseeff, A., and Kandola, J., "On kernel Target Alignment," in Advances in Neural Information Processing Systems 14, Editors T. G. Dietterich and S. Becker and Z. Ghahramani, 2001.

[6] 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.

[7] Ganapathiraju, A., "Support vector machines for speech recognition" PhD Thesis, Mississipi State University, 2002.

[8] Hermansky, H., Morgan, N., "RASTA processing of speech", "IEEE Trans. On Speech and Audio Processing", vol. 2, no. 4, pp. 587-589, Oct. 1994.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] Schölkopf, B. and Smola, A., "Learning with kernels", M.I.T. Press, 2001.

[14] 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.

[15] 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.

[16] Vapnik, V., "Statistical Learning Theory", Wiley, 1998.

[17] Young, S. et al., "HTK-Hidden Markov Model Toolkit (ver 2.1)", Cambridge University, 1995.