Voice Activity Detection Using MFCC Features and Support Vector Machine

Tomi Kinnunen[1], *Evgenia Chernenko*[2], *Marko Tuononen*[2], *Pasi Fränti*[2], *Haizhou Li*[1]

[1] Speech and Dialogue Processing Lab, Institute for Infocomm Research (I²R), Singapore
[2] Speech and Image Processing Unit, Department of Computer Science, University of Joensuu, Finland
{echernen,mtuonon,franti}@cs.joensuu.fi {ktomi,hli}@i2r.a-star.edu.sg

Abstract

We define voice activity detection (VAD) as a binary classification problem and solve it using the support vector machine (SVM). Challenges in SVM-based approach include selection of representative training segments, selection of features, normalization of the features, and post-processing of the frame-level decisions. We propose to construct a SVM-VAD using MFCC features because they capture the most relevant information of speech, and they are widely used in speech and speaker recognition making the proposed method easy to integrate with existing applications. Practical usability is our driving motivation: the proposed SVM-VAD should be easily adapted into new conditions.

Index Terms: voice activity detection (VAD), machine learning, support vector machine (SVM)

1. Introduction

Voice activity detection (VAD) aims at classifying a given sound frame as a speech or non-speech. It is needed as a front-end component in voice-based applications such as speech recognition, speech enhancement, variable frame-rate speech coding, and speaker recognition. Furthermore, VAD is an important tool for a forensic analyst to locate the speech-only parts from large audio collections which can consists of tens of hours of data [1].

A large number of methods have been proposed. Simple methods are based on comparing the frame energy, zero crossing rate, periodicity measure, or spectral entropy with a detection threshold to make the speech/non-speech decision. More advanced models include statistical hypothesis testing [2], long-term spectral divergence measure [3, 4], amplitude probability distribution [5], and low-variance spectrum estimation [6]. The common property in these methods is that they include estimation of the background noise levels and/or noise suppression as a part of the process. The methods usually have a large number of control parameters, which are more or less tuned to a specific application. As an example, in [1] it was reported that the accuracy of the long-term spectral divergence VAD [3] depends much on the selection of the seven control parameters of the method.

In this paper, we propose to extract the standard mel-frequency cepstral coefficients (MFCC) with delta and double coefficients and train a binary classifier using training files with speech/non-speech annotation. The VAD then labels each test utterance frame by using the trained classifier. We use the support vector machine (SVM) as the classifier since this has shown excellent performance in other classification tasks, e.g. speaker verification [7].

An advantage of this supervised learning is that it can be easily adapted to new operating conditions by providing representative training examples for the new condition. In this way, optimization of the parameters is absorbed to the training algorithm of the SVM whereas optimizing the parameters of conventional VADs, on the other hand, is more difficult.

We compare the proposed method with existing ones based on energy levels, long-term spectral information, and Gaussian mixture modeling. We provide comparative results on three different datasets with a varying degree of difficulty and discuss our results.

2. SVM-based VAD

We are aware of two prior studies on using SVM for voice activity detection [4, 8]. In [8], the authors used four-dimensional features from the G.729B VAD as input to the SVM. The method reached 4% absolute improvement in the error rate in comparison to the G.729B VAD. Since both method use the same set of features, the improvement was due to the SVM. In [4], the authors used contextual speech features from a long-term spectral envelope as the features. The SVM-based VAD was compared with nine alternative methods and it yielded the best performance when the two control parameters of the feature extraction were set properly.

2.1. Basic SVM Structure

SVM is a *binary* classifier, which models the decision boundary between the two classes as a *separating hyperplane*. The training set for an SVM consists of positive and negative training vectors. In our case, the positive vectors labeled as +1 correspond to speech feature vectors and the negative vectors labeled as -1 correspond to non-speech feature vectors. The SVM decision function is defined as follows:

$$f(\mathbf{y}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i, \mathbf{y}) + b \qquad (1)$$

where $\boldsymbol{y}$ is the unclassified tested vector, $\boldsymbol{x}_i$ are the support vectors and α_i their weights and b is a constant bias. $K(\boldsymbol{x},\boldsymbol{y})$ is the *kernel function* which performs implicit mapping into a high-dimensional feature space. The support vectors are obtained from the training sample through an optimization process, and therefore they are a subset of the training sample. We use the publicly available SVM^{light} tool for optimizing the support vectors [9][1]. As for the kernel function, we consider *linear* and *radial basis function* (RBF) kernels, which are defined respectively as follows [9]:

$$K_{\text{lin}}(\mathbf{x},\mathbf{y}) = <\mathbf{x},\mathbf{y}> \qquad (2)$$

$$K_{\text{RBF}}(\mathbf{x},\mathbf{y}) = \exp\left(-\gamma \,||\, \mathbf{x}-\mathbf{y} \,||^2\right), \qquad (3)$$

where <.> denotes the inner product, γ is a control parameter (kernel width) and ||.|| denotes the Euclidean norm. Potentially the RBF kernel gives better results compared with the linear

[1] http://svmlight.joachims.org/

kernel [4]. On the other hand, it includes an additional control parameter γ and the running times of both training and testing are much longer.

2.2. MFCC features

We propose to use the MFCC features mostly because they are standard features used in speech processing and readily available in various software packages, which make the integration of the feature extraction and VAD easy. Typical MFCC vectors appended with delta and double-delta coefficients are 36-dimensional. In a feature space of this dimension, it is reasonable to assume sufficiently good separation between the speech and non-speech vectors.

MFCCs are also relatively independent of the absolute signal level that would be beneficial in cases where the energy-based methods fail by classifying low-energy speech frames as non-speech or high-energy non-speech frames as speech. By including or excluding the first MFCC coefficient (i.e. the DC component of the short-term spectral envelope), the MFCC setup can be fine-tuned to be more or less sensitive to the absolute energy levels, respectively. A disadvantage of MFCCs would be that they are sensitive to channel mismatch between training and testing, and they are also speaker-dependent. For the VAD application the channel and speaker factors should be normalized.

It is reasonable to hypothesize that the accuracy of SVM-based VAD depends much on the selection of the training speech material and the features. Therefore, it is important that the training material is representative of the operating conditions and that the features can discriminate speech from non-speech.

2.3. Training and classification using SVM

In the training phase, we extract features from multiple training files with speech/non-speech annotation. The positive and negative vectors are combined in the respective pools and a single SVM is trained.

In the operation phase, feature vectors are extracted from the unknown sample. SVM output score is computed for each vector using (1). Since the frame-level output score of the SVM is rather noisy, we apply median filtering to smooth the score. The median filtered score is compared with a detection threshold in order to get decisions. Fig.1 shows an example of the VAD steps. Median filtering gives, on average, a relative reduction of 30% in the error rates relative to the unfiltered scores.

3. Materials and Methods

3.1. Data sets and Features

In our experiments, we use three datasets whose attributes are listed in Table 1. The first dataset is a subset of the NIST2005[2] speaker recognition evaluation corpus, consisting of conversational telephone-quality speech having a sampling rate of 8 kHz. We selected 15 files for our purposes, all from different speakers and having duration of 5 minutes per file.

[2] NIST Speaker Recognition Evaluations. http://www.nist.gov/speech/tests/spk/

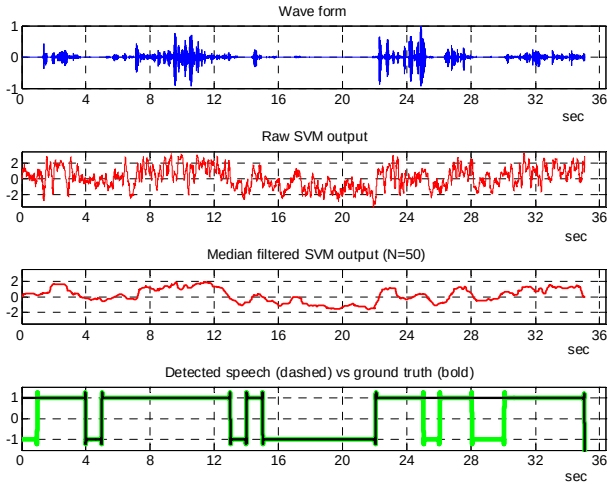

Fig.1: Example of VAD in action. From top to bottom: waveform, raw SVM output score, median-filtered SVM output score, and decisions. Manually annotated ground truth is shown for reference.

Table 1: Data sets used in the experiments and their partitioning into training and test sections.

	NIST 2005	Bus stop	Lab
Recording equipment	Telephone	Telephone	Labtec PC microphone
Training section	25 min (5 spk. x 5min)	61 min	85 min
Test section	50 min (10 spk. x 5min)	105 min	170 min
Speech-to-non-speech ratio	53%:47%	75%:25%	12%:88%

The second data set, referred to as "Bus stop" data, consists of timetable system dialogues recorded in 8 kHz sampling rate. The material consists of human speech commands that are mainly very short, and synthesized speech that provides rather long explanations about bus schedules. Finally, the "Lab" data set consists of a one long continuous recording from the lounge of our laboratory in 44.1 kHz. The goal of the material was to simulate wiretapping material collected by the detectives. For more details on the Bus stop and Lab data sets, refer to [1].

From the three datasets, the NIST data is used for studying the effect of feature extraction and SVM parameters. The other two data sets are used in validating the results and comparing the SVM results with three existing VAD methods. Each data set has been manually annotated using a resolution of 1 second as described in [1]. We divided each data set into non-overlapping training and test sections of 30% and 70%, respectively.

For the MFCC features, the frame length is 30 milliseconds and frame shift 20 milliseconds. We use 27 triangular filters and 12 cepstral coefficients, excluding the 1st coefficient. The MFCC vectors are appended with delta and double delta coefficients to yield 36-dimensional features. In preliminary experiments, we reduced the number of mel filters and cepstral coefficients so as to reduce speaker variability which indeed lead to some improvements. However, we purposely kept the spectral front-end similar to those used in the speech and speaker recognition front-ends so that VAD and feature extraction can be easily integrated.

3.2. Comparative VAD Methods

We include the following methods in our comparisons:

- Short-term energy-based method [10]
- Long-term spectral divergence method (LTSD) [3]
- Gaussian mixture model (GMM) [11]

The energy-based method first measures the energy of each frame in the file and then sets the speech detection threshold relative to the maximum energy level. This method includes two parameters and they were optimized on the NIST datasets during the preparations of the Institute for Infocomm Research to the NIST 2006 speaker recognition benchmarking [10].

The LTSD method uses long-term spectral divergence between speech and noise, and its parameters were set up as explained in [1]. The speech/non-speech decision rule is formulated by comparing the long-term spectral envelope to the average noise spectrum. The noise model is initialized using the beginning part of each file.

The GMM-based VAD uses the concept of adapted GMMs [11]. First, we train a general universal background model (UBM) of 256 diagonal-covariance Gaussian components using all the training data of the given corpus. This is followed by maximum a posteriori adaptation of the mean vectors to give the adapted speech- and non-speech models. The log likelihood ratio computed using the fast *N*-top scoring algorithm [11] is used as the VAD indicator.

3.3. Evaluation methodology

For VAD, we have two error types: miss and false alarm. *Miss* refers to miss of true speech segment when the VAD declares a frame as non-speech but it is speech. *False alarm* refers to the case when the VAD declares a frame as speech but it is non-speech. Depending on the application, either error type can be considered more harmful. The operating point can be selected by adjusting the decision threshold. By lowering the threshold, we can reduce the number of missed speech segments at the cost of increased number of false alarms.

We use the detection error trade-off curve (DET) as the evaluation tool. The DET plot shows the probability of miss (P_{miss}) as a function of the probability of false alarm (P_{fa}) on a normal deviate scale. For a detection error curve, we can also compute the *equal error rate* (EER) which corresponds to the threshold for which $P_{\text{miss}}=P_{\text{fa}}$.

In parameter optimizations, we used mostly the EER as the evaluation metric. To reflect the differences of the methods in realistic application scenario, the final comparisons include two extreme operating points to minimize the probability of either miss or false alarm, in addition to the EER point. We set the threshold to yield $P_{\text{miss}}=2\%$ (or $P_{\text{fa}}=2\%$) and measure the other error rate at this threshold.

4. Parameter Optimization Results

First, we compare the effect of the training material using SVM with the linear kernel. In particular, we are interested to see the effect of using multiple files (speakers) for training as opposed to using only a single speaker, as well as the effect of the training data length. Prior to pooling the feature vectors from different files for training, we normalize the features within each file to zero-mean and unit variance to reduce between-file variability due to speaker and channel differences. The results presented in Fig.2 shows that combining the training files improves accuracy as expected. For the rest of the experiments, we use the pooled training set.

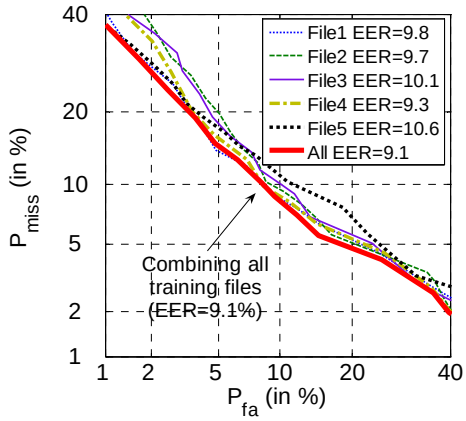

Fig.2: comparing single-speaker and.multi-speaker training for the SVM on the NIST dataset.

The results in the Table 2 indicate that SVM is rather independent of the training data length and even 10 seconds of training data is sufficient for this material.

Table 2: effect of the training material length.

Training length	10 sec	30 sec	1min	3 min	5 min	10 min
EER (%)	9.3	8.8	9.1	9.3	9.4	9.4

Next, we study the effect of the SVM kernel. The results are presented in Table 3. The RBF slightly outperforms the linear kernel which is consistent with the statement in [4]. However, the running times of both the training and testing for the RBF kernel are much higher compared with the linear kernel, which can be a limitation for practical applications. In the rest of the experiments, we therefore use the linear kernel unless otherwise mentioned.

Table 3: Comparison of SVM kernels.

SVM kernel	Accuracy (EER %)	Training time (s)	Test time(s)
Linear	9.1	429	107
RBF (γ=0.3)	8.1	1776	924
RBF (γ=0.6)	8.0	2010	1062

5. Comparative Evaluation of the Methods

The comparison of the different VAD variants is shown in Table 4 and the corresponding DET plots for each dataset are shown in Fig.3.

Table 4: Comparison of different VAD methods on the three datasets using three different operating points.

		Adaptive		Trained	
		Energy [10]	LTSD [3]	SVM	GMM [11]
NIST 2005	EER	1.5	40.0	8.0	12.8
	$P_{\text{miss}}@P_{\text{fa}}$= 2%	1.4	40.0	26.7	43.3
	$P_{\text{fa}}@P_{\text{miss}}$= 2%	1.2	62.5	21.5	32.6
Bus stop	EER	14.6	19.2	13.1	34.0
	$P_{\text{miss}}@P_{\text{fa}}$= 2%	62.3	100.0	40.9	99.1
	$P_{\text{fa}}@P_{\text{miss}}$= 2%	27.2	36.0	53.4	68.4
Lab	EER	16.8	14.4	19.0	15.3
	$P_{\text{miss}}@P_{\text{fa}}$= 2%	80.6	76.8	54.7	59.8
	$P_{\text{fa}}@P_{\text{miss}}$= 2%	65.3	19.3	89.1	67.8

The energy-based VAD clearly outperforms SVM and LTSD on the NIST data set. This is not surprising since it was optimized for the NIST corpuses through extensive testing. The LTSD fails miserably on the NIST data set, and the SVM falls in between the energy and LTSD methods. Detailed investigation of the LTSD results revealed that the noise model initialization failed on some of the NIST files, causing the high error rates. The beginning of the problematic files were speech whereas the method assumes it to be nonspeech when initializing the noise model.

In the case of Lab data set, none of the methods is superior to each other but the performance depends on the error (miss or false alarm) that we wish to minimize. If we wish to keep the speech miss rate low (forensics application and voice-dialogue system), LTSD and energy methods yield the lowest false alarm rates. On the other hand, if we wish to have a low false alarm rate (automatic speaker verification), the SVM yields the lowest speech miss rates from all the methods. The situation is quite similar for Bus stop data. SVM shows the lowest speech miss rates and in this case the lowest EER as well.

The GMM performance varies a lot between the three data sets. The number of Gaussian components was optimized on the NIST data set, and this may not be the best choice for data sets which have a very different training set size and speech-to-nonspeech ratio on the training data, as is the case here. For SVM, over- or underfitting is less an issue because much less data is needed for training the hyperplane parameters than the density estimates in GMM, which are notorious for needing large training data per dimensionality ratio. Further optimizations of the GMM adaptation parameters and fusion of SVM and GMM classifiers are points for future research.

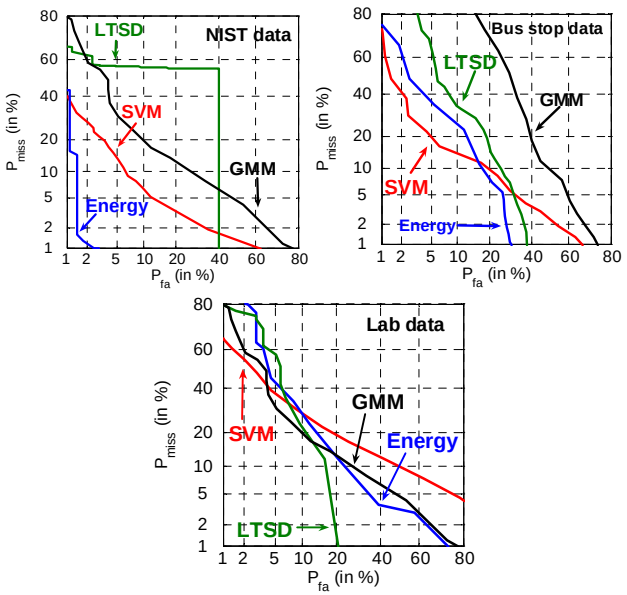

Fig.3: Comparison of different VADs on the three data sets.

6. Conclusions

Voice activity detection using MFCC features and support vector machine was proposed. The method works excellently when small false alarm rate is desired, which is the case in text-independent speaker verification, for example. Main advantage of the SVM-based VAD is that it works consistently in the same manner with different corpora: smooth DET curve without sudden peaks. The other methods were more prone to the change of data set and variations of their parameters. Our main conclusion is that, according to our experiments, SVM is easier to adapt to the new data sets than conventional methods as long as we have a short training audio sample from the recording environment.

7. References

[1] M. Tuononen, R. González Hautamäki, P. Fränti, "Applicability and Performance Evaluation of Voice Activity Detection", submitted to *IEEE Trans. On Information Forensic and Security*.

[2] J.-H. Chang, N.S. Kim and S.K. Mitra, "Voice Activity Detection Based on Multiple Statistical Models", IEEE Trans. Signal Processing, 54(6), June 2006, pp. 1965-1976.

[3] J. Ramírez, J.C Segura, C. Benítez, A. de la Torre, A. Rubio (2004) "Efficient voice activity detection algorithms using long-term speech information". *Speech Comm.* 42, pp. 271–287.

[4] J. Ramírez, P. Yelamos, J.M. Gorriz, J.C. Segura (2006) "SVM-based speech endpoint detection using contextual speech features". *Elec.Letters* 42(7), 2006.

[5] S.G. Tanyer and H. Özer, "Voice Activity Detection in Nonstationary Noise". IEEE Trans. Speech and Audio Processing, 8(4), July 2000.

[6] A. Davis, S. Nordholm, R. Togneri, "Statistical Voice Activity Detection Using Low-Variance Spectrum Estimation and an Adaptive Threshold", 14(2), March 2006.

[7] W.M. Campbell, J.P. Campbell, D.A. Reynolds, E. Singer, P.A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition", *Computer Speech and Language* 20(2-3), pp. 210-229, April 2006.

[8] D. Enqing, L. Guizhong, Z. Yatong and Z. Xiaodi, "Applying Support Vector Machines to Voice Activity Detection", 6th Int. Conf. on Signal Processing, 2, 26-30 Aug. 2002 pp.1124 – 1127.

[9] T. Joachims, "Making large-Scale SVM Learning Practical", *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf, C. Burges and A. Smola (eds.), MIT-Press, 1999.

[10] R. Tong, B. Ma, K.A. Lee, C.H. You, D.L. Zhou, T. Kinnunen, H.W. Sun, M.H. Dong, E.S. Ching and H.Z Li, "The IIR NIST 2006 Speaker Recognition System: Fusion of Acoustic and Tokenization Features". Accepted for presentation in 5th Int. Symp. on Chinese Spoken Language Processing, ISCSLP, December 2006, Singapore.

[11] D.A. Reynolds and T.F. Quatieri and R.B. Dunn, "Speaker Verification Using Adapted Gaussian Mixture Models", *Digit. Signal Processing*, 10(1), pp.19-41, January 2000.