Silence/Speech Detection Method Based on Set of Decision Graphs*

Jan Trmal, Jan Zelinka, Jan Vaněk, and Luděk Müller

University of West Bohemia, Department of Cybernetics,
Univerzitní 8, 306 14 Plzeň, Czech Republic
{jtrmal, zelinka, vanekyj, muller}@kky.zcu.cz

Abstract. In the paper we demonstrate a complex supervised learning method based on a binary decision graphs. This method is employed in construction of a silence/speech detector. Performance of the resulting silence/speech detector is compared with performance of common silence/speech detectors used in telecommunications and with a detector based on HMM and a bigram silence/speech language model. Each non-leaf node of a decision graph has assigned a question and a sub-classifier answering this question. We test three kinds of these sub-classifiers: linear classifier, classifier based on separating quadratic hyper-plane (SQHP), and Support Vector Machines (SVM) based classifier. Moreover, besides usage of a single decision graph we investigate application of a set of binary decision graphs.

1 Introduction

A silence/speech detection is an inseparable part of speech signal processing problems. Silence/Speech Detectors (SSD) are used to reduce data bandwidth demands, in voice compression algorithms, serves in noise adaptation techniques and in adaptation techniques in general. A high-quality SSD should be able to perform well under different operational conditions, in presence of a noise and independently on speaker.

Different signal processing algorithms can have very distinct requirements on a SSD. In general, both false positive errors (marking a non-speech signal as a speech signal) and false negatives (marking a speech signal as a non-speech signal) are targets of minimization. However, some applications can give priority to false negatives reduction (for example Voice Activity Detectors (VAD) used in telecommunication speech codecs), some applications prioritize false positives (for example adaptive echo cancellation algorithms and speaker verification tasks), some applications prefer only to recognize silence between complete sentences and not to recognize pauses between words (such as speech recognition tasks).

In this paper we describe methods based on construction of decision graphs i.e. hierarchical ordered set of classifiers. We use different sub-classifiers (linear classifiers, quadratic classifiers and SVM based classifiers) and compare their performance to the prevalent GMM classifier and to the detectors from telecommunication environment.

* Support for this work was provided by the Grant Agency of Academy of Sciences of the Czech Republic, project No. 1ET101470416 and by the Ministry of Education of the Czech Republic, project No. MŠMT LC536.

Petr Sojka, Ivan Kopeček and Karel Pala (Eds.): TSD 2006, LNAI 4188, pp. 539–546, 2006.

2 Sub-classifiers

The first described sub-classifier is a SVM based classifier. The SVM is a set of supervised learning methods for classification and regression. Usually, these methods use a transformation of a input vector to a vector with (much) higher dimensionality. The transformed data are supposed to be linearly separable.

A special method for the purpose of training the linear classifier is used. Resulting classifier is represented by the maximum margin hyper-plane, i.e. hyperplane separating two clusters and being equally distant from each of them. Quadratic optimization problem must be solved for finding this hyper-plane.

Given training vectors $\mathbf{x}_i \in R^n$, $i = 1, \ldots N$, vectors of supposed classifier decisions $y \in R$, such that $y_i \in \{-1, +1\}$, and some non-linear transformation $\phi(\mathbf{x})$, the C-SVM solves the following problem:

$$\min_{w,b,\xi} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \tag{1}$$

subject to

$$\begin{aligned} y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) &> 1 - \xi_i \\ \xi_i &\geq 0 \; i = 1 \ldots N \end{aligned}$$

This formulation is the primary problem of Quadratic Programming (QP). It is necessary to explicitly transform the data and use these transformed data. This fact can be prohibiting (so-called "curse of dimensionality"), so an alternative formulation is used. This alternative formulation is obtained by transforming the problem into its dual formulation.

The dual formulation is

$$\max_{\alpha} \quad -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \Phi(\mathbf{x}_i, \mathbf{x}_j)\alpha_i\alpha_j + \sum_{j=1}^{N}\alpha_j \tag{2}$$

subject to

$$\begin{aligned} &\sum_{i=1}^{N}\alpha_i y_i = 0 \\ &0 \leq \alpha_i \leq C \qquad \forall i \in \{1, \ldots, N\} \end{aligned}$$

where $\Phi(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ is a kernel function. A kernel function is a function fulfilling conditions stated in Mercer's Theorem and can be interpreted (in this context) as a dot product of its input vectors transformed into some high-dimensional space. The classifier resulting from the dual problem has a form

$$y(\mathbf{x}) = \operatorname{sign}\sum_{i=1}^{N}\alpha_i\Phi(\mathbf{x}, \mathbf{x}_i) \tag{3}$$

The dual QP formulation uses an implicit mapping of the input vectors. We do not provide the function transforming the input feature vectors, all we have to do is to supply some function representing the dot product in that resulting space.

Training of SVM often requires solution of a very large QP optimization problem. Although for solving QP general algorithms could be used, there exist special algorithms for the SVM training. These algorithms exploit the special structure of the SVM QP problems. Probably the most used algorithm is the Sequential Minimal Optimization (SMO). SMO breaks the large QP problem into a series of the smallest possible QP problems. These small QP problems are solved analytically, so the time-consuming iterative numerical QP optimization is avoided. The amount of memory required for SMO is linear in the training set size which allows SMO to handle very large training sets. For more detail see [2].

The second described classifier is a quadratic classifier which dissects the feature vector space with a quadratic hyper-plane. This classifier is motivated by a probabilistic approach where a feature vector $\mathbf{x}$ is classified as a class $y \in \{-1; +1\}$ and PDF $p(\mathbf{x}|y)$ is approximated by a normal distribution with diagonal covariance matrices. The implemented quadratic separating hyper-plane is of the following form:

$$\sum_{i=1}^{n} a_{2,i} \cdot x_i^2 + \sum_{i=1}^{n} a_{1,i} \cdot x_i + a_0 = 0, \tag{4}$$

where $a_{i,j}$ and a_0 are parameters of the hyper-plane.

The third sub-classifier is a linear classifier which is the consequence of the equality of the covariance matrices.

We used a Genetic Algorithm (GA) to find the optimal parameters estimation for the linear and the quadratic classifier. An initial result which serves as a population initialization is computed as a linear hyper-plane separating two feature vectors belonging to different classes randomly selected from the training set. The mutation operator modifies some randomly chosen parameter by addition of a random number having the PDF with zero mean and fixed variance. The crossover operator was not used.

3 Telecommunication Silence/Speech Detectors

In most cases, the silence/speech detectors used in the telecommunication environment were designed as simple rule-based systems which are driven by a set of features and flags obtained during the speech parameterization and compression. The purpose was to preserve the low complexity of voice codecs. They must be able to operate reliably even under marginal condition with respect to different environments, different speakers and different languages.

Their primary purpose was to enable the bandwidth saving effect through the means of discontinuous transmission (DTX) and to enable some basic adaptation of algorithms to a background noise, but the false positives were given priority over speech cut-offs. Often, a simple hangover scheme is included into the standard to ensure transmission of the undetected parts of speech.

We tested four commonly different VAD used. The first one belonging to G.729 coding is the historically oldest. The G.729 coding is still used today and is connected with Voice over IP technologies. There exists standards defining voice coding and transmission at 8 kbps, 6.4 kbps and 11.8 kbps.

The three other VAD are of similar age and belong to the Adaptive Multi-Rate technology (AMR). AMR speech codec is a mandatory codec for the third generation mobile phone systems (3GPP) and is supposed to be widely used in cellular systems. The AMR-WB codec works with signal sampled 16 kHz, while the "plain" AMR works with 8 kHz sampled speech signal.

All mentioned codecs use the Algebraic-Code-Excited Linear Prediction technique (ACELP). They partially differ in the codebook searching strategy. For details, see [8].

4 Binary Decision Graphs

A binary decision graph (BDG) for classification is an acyclic graph, whose each leaf has assigned a probability distribution of classification, and each non-leaf node has assigned a YES-NO question. The higher accuracy of classification and the lower number of nodes are the main contributions in comparison with a binary decision tree. The BDG construction algorithm tries to find the BDG having the lowest error rate on the training data. In our experiments we restricted the number of nodes and we limited the maximal accuracy to avoid overtraining.

There are two basic approaches to BDG construction. The first approach is to transform a binary decision tree into a BDG. The goal is not only to maximize the accuracy but we also try to minimize number of nodes. Thus, the transformation converts a binary decision tree into a BDG with the same or higher accuracy and simultaneously tries to reduce the number of nodes as much as possible. The transformation consists in a sequence of several relatively elementary operations such as nodes merging and nodes deletion.

The second approach to BDG construction is to construct a BDG directly. A simple algorithm implementing this approach is the modification of top-down binary decision tree construction algorithm and works as follows:

1. Start with a set of all examples at the root node.
2. While the number of nodes is lower than the selected threshold and the accuracy is lower than the selected threshold do:
   (a) Apply the transformation.
   (b) Select some leaf node n with the set of examples M.
   (c) Evaluate all possible questions for node n, choose the optimal (or suboptimal) question q and associate it with the node n.
   (d) According to the question q, divide the set of examples M into the sets M_{YES} and M_{NO} and make two new successor nodes: node n_{YES} with the set of examples M_{YES} and the node n_{NO} with the set of examples M_{NO}.
3. Apply the transformation.

There are two crucial points in the BDG construction algorithm. The first one is the question evaluation and the suboptimal question searching algorithm. A question is relevant only if $|M_{YES}| > 0$ and $|M_{NO}| > 0$. The suboptimal question searching algorithm concedes only a relevant question. In our experiments we used three kinds of classifiers (linear, quadratic and SVM based) and we used only one kind at a time.

Our question evaluation function E evaluating a question q is defined:

$$E(q, TS) = \sum_{i=1}^{n_E} c_i \cdot E_i(q, TS), \tag{5}$$

where TS is the training set, E_i is the i-th partial evaluating function and manually fixed c_i is the weight of function E_i. Besides entropy we used the relative number of corrected classification error estimation as a partial evaluating function. The weights of these partial evaluating functions are higher than weights of the other simpler partial evaluating functions.

We apply a genetic algorithm as a (sub)optimal question searching algorithm for linear and quadratic classifiers, and we apply a grid search algorithm for SVM based questions.

The second point is the selection of a leaf node which will be expanding later. Our algorithm tries to expand all leaf nodes and selects the leaf node of which expansion corrects most errors.

In addition to one single BDG, we utilized a set of BDG. Each BDG in the set is constructed from a unique part of the training set. Unique parts can be selected in a fully random manner or can be determined by some automatic clustering method. We utilized the second alternative. Our implemented clustering is a simple non hierarchical k-means algorithm. The influence of the number of clusters k on the overall performance is considered in section Experiments and Results.

5 HMM Based Silence/Speech Detector

An HMM based SSD is a modified HMM based speech recognition system. Hence, we can distinguish three parts of the HMM based SDD.

The first part is an acoustic model. The acoustic model consists of models of phonemes or more differentiated units such as triphones, i.e. phones situated between two specific phones. The acoustic model in our HMM based speech detection consists of only two models: model of silence and model of speech. Both models permit generation of exactly one segment. The output probability density function in each state is approximated by Gaussian Mixture Model (GMM) with diagonal covariance matrices. The number of mixtures in GMM is given in the experimental results in the Section 6. In our experiment we gradually increased the number of mixtures by one and saved and tested the particular models.

The second part is a pseudo-language model which is implemented as a pseudo-word net. We use a n-gram based pseudo-language model. The model is based on language having only two words: “silence” and “speech”. In our experiments we compared the zerogram, unigram and bigram based pseudo-language models. The detector based on the zerogram pseudo-language model makes a decision in accordance with GMM models only. The unigram model uses information about a priori probability of the speech and silence. The bigram model is more complicated and use four transition probabilities. All the a-priori probabilities were computed from the training part of the data.

6 Experiments and Results

In our experiment we used the Czech high-quality speech corpus [6]. In the first experiment we tested the telecommunication SSD. We downsampled the test recordings because the

corpus contains recordings sampled at 44.1 kHz and the detectors operate at signals sampled at 8 kHz and 16 kHz respectively.

Table 1. Comparison of performance of detectors used in the telecommunication SSD

	G.729	AMR/2	AMR/1	AMR-WB
mean	85.1 %	89.22 %	89.22 %	85.51 %
max	94.9 %	98.96 %	98.96 %	98.38 %
min	39.9 %	56.69 %	56.69 %	49.23 %

In all following experiments, the MFCC feature vectors were used. In the second experiment the method based on BDG with linear sub-classifiers was considered. In the experiment we studied the influence of the maximal number of nodes m in one BDG and the number of clusters k on the accuracy of the silence/speech classification. The results of the second experiment are in Table 2. In the first column, i.e. for $m = 1$, only an a priori information for each cluster is applied. In the second column, i.e. for $m = 3$, only one sub-classifier of given kind is used in each cluster.

Table 2. The results of classification by means of sets of BDG with linear sub-classifiers

k	$m = 1$	$m = 3$	$m = 10$	$m = 20$	$m = 30$
1	78.73 %	96.41 %	96.98 %	97.03 %	97.03 %
10	92.36 %	96.50 %	97.11 %	97.21 %	97.22 %
100	95.44 %	96.91 %	97.26 %	97.30 %	97.32 %
200	95.39 %	97.04 %	97.26 %	97.28 %	97.29 %
300	95.92 %	96.96 %	97.20 %	97.22 %	97.23 %

The results of the third experiment where the method based on BDG with quadratic sub-classifiers was considered are slightly better as shown in the Table 3. Also, the overtraining effect is not so noticeable.

Table 3. The results of classification by means of sets of BDG with quadratic sub-classifiers

k	$m = 1$	$m = 3$	$m = 10$	$m = 20$	$m = 30$
1	78.73 %	96.85 %	96.95 %	97.03 %	97.03 %
10	92.36 %	96.37 %	97.15 %	97.35 %	97.37 %
100	95.44 %	96.91 %	97.40 %	97.41 %	97.44 %
200	95.39 %	97.17 %	97.43 %	97.45 %	97.47 %
300	95.92 %	97.17 %	97.36 %	97.40 %	97.41 %

Table 4. The results of classification by means of sets of BDG with SVM sub-classifiers

k	$m = 1$	$m = 3$	$m = 10$	$m = 20$
1	78.73 %	93.84 %	96.05 %	96.85 %
10	92.36 %	96.37 %	97.02 %	96.83 %
100	95.44 %	96.67 %	97.39 %	96.94 %

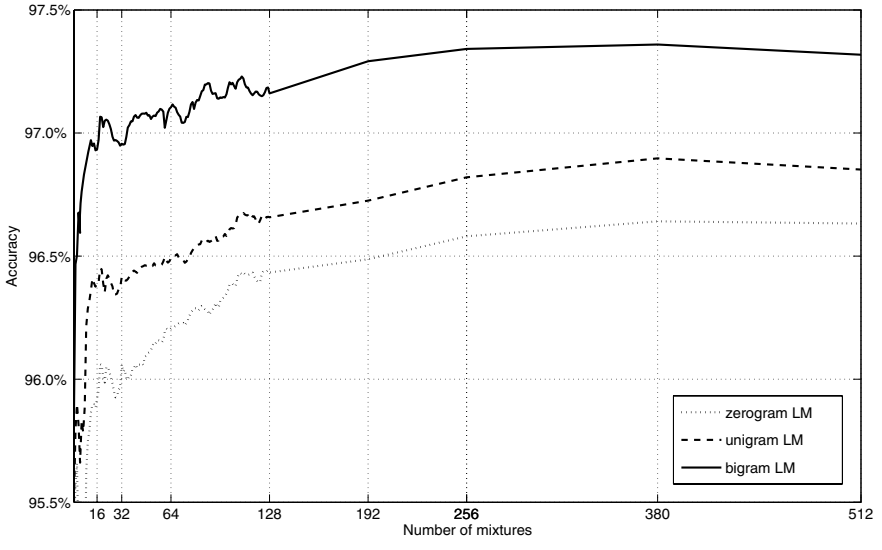

Fig. 1. Performance of the HMM Based Silence Detector

As we already mentioned, we used also the SVM classifier. However, we were not able to train the SVM classifiers using the whole set M during the node n expansion, because the process of training of SVM was exceedingly time consuming. To resolve this, each time when the number of input feature vectors was higher than some empiricaly chosen number Q, we had chosen exactly Q feature vectors in a fully random manner from the set M. This can be the reason of the worse performance (compare results).

In the last experiment the HMM based SSD was considered. The results are shown in Figure 1. As can be seen from the figure, the pseudo-language modeling is a beneficial option. With the increase of n in n-gram models the accuracy increases. The best result (97.36 %) has been obtained for 380 mixtures with the bigram pseudo-language model.

7 Conclusion

The speech detection still cannot be declared as a solved problem. After all, the highly accurate silence detection problem is comparable to a speech recognition problem. Our experiments have shown that there exists a wide performance gap between methods which

are expected to provide stable results across many mutually different operating environments while keeping low computational demands and methods which are constructed using algorithms which are computationally somewhat more expensive and targeted to operate in defined environment.

In the future, we intend to develop a robust real-time SSD for dialog systems. Another aim is to apply the decision based methods in audio-visual speech recognition system as a posteriors estimation algorithm.

References

1. Vapnik, V.: Statistical Learning Theory, John Wiley & Sons, Inc., New York, (1999). ISBN 0-471-03003-1.
2. Platt, J.: Using Sparseness and Analytic QP to Speed Training of Support Vector Machines, in Advances in Neural Information Processing Systems 11, M. S. Kearns, S. A. Solla, D. A. Cohn, eds., MIT Press, (1999).
3. Voice Activity Detector for Adaptive Multi-Rate speech traffic channels, GSM 06.94 version 7.1.1 Release 1994 Telecommunications Standards Institute (1994).
4. AMR Wideband speech codec; Voice Activity Detector (VAD), 3GPP TS 26.194 version 6.0.0 Release 6. European Telecommunications Standards Institute (1994).
5. VAD for Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear-Prediction (CS-ACELP) – ITU-T Recommendation G.729 Annex B.
6. Müller, L.; Psutka, J.: Building robust PLP-based acoustic module for ASR applications. In SPECOM 2005 proceedings. Moscow: Moscow State Linguistic University, 2005. pp. 761–764. ISBN 5-7452-0110-X.
7. Radová V.; Psutka J.: UWB_S01 Corpus: A Czech Read-Speech Corpus, Proceedings of the 6th International Conference on Spoken Language Processing ICSLP2000, Beijing 2000, China. Volume IV., pp. 732–735.
8. Chu, W. C., Speech Coding Algorithms: Foundation and Evolution of Standardized Coders, John Wiley and Sons, Inc., New Jersey, USA, 2003, ISBN 0-471-37312-5.
9. Šmídl, L.; Prcín, M.; Jurčíček F.: How to Detect Speech in Telephone Dialogue Systems. In: Proceedings of EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services ECMCS 2001, Hungary, Budapest, 2001, on CD-ROM. (ISBN 963-8111-64-X).
10. Cornu, E.; Sheikhzadeh H.; Brennan R. L.; Abutalebi H. R. et al: ETSI AMR-2 VAD: Evaluation and Ultra Low-Resource Implementation. In: International Conference on Acoustics Speech and Signal Processing (ICASSP '03), 2003, available at `www.amis.com/tech_resources/dsp_technology_papers/ICASSP2003_VAD.pdf`