# VOICE ACTIVITY DETECTION IN THE PRESENCE OF BREATHING NOISE USING NEURAL NETWORK AND HIDDEN MARKOV MODEL

*Mikko Myllymäki and Tuomas Virtanen*

Department of Signal Processing, Tampere University of Technology
Korkeakoulunkatu 1, 33720, Tampere, Finland
email: mikko.v.myllymaki@tut.fi, tuomas.virtanen@tut.fi

## ABSTRACT

This paper proposes a voice activity detection algorithm to be used in the presence of breathing noise. We use a hybrid approach where a neural network is first applied in individual frames using mel-band energies within the frame as inputs. The output of the neural network is then processed using a hidden Markov model, which takes into account the temporally continuous nature of speech activity. Both the neural network and the hidden Markov model can be trained in supervised manner. On simulations with realistic acoustic material, the proposed method achieved average frame-level sensitivity above 97% and average specificity above 95%. The proposed algorithm enables a good rejection of noise and breathing frames while retaining the intelligibility of input speech.

## 1. INTRODUCTION

Voice activity detection (VAD) refers to the task of locating speech segments from an input signal which consists of user's speech and other sound sources captured by the recording device. It is commonly used to decrease the battery and bandwidth usage in communication devices by switching the transmitter off during speech pauses. Speech segments can be detected, e.g., by measuring the energy of the input signal [1], or by making other assumptions of the statistical properties of the speech and noise signals [2]. There exist several standards of VAD algorithms for telecommunication devices, see [3, pp. 357-377] for a review. In practice all existing VAD algorithms process an input signal in short frames and produce a speech/non-speech decision for each frame, and therefore VAD can be viewed as a classification problem where the frames are observed sequentially.

This paper deals with the VAD in the presence of high-level breathing noise. This kind of algorithms are useful for example in communication devices, in which the microphone is located directly in front of the mouth of the user. Such devices are used, e.g., by professionals working on security and safety fields, which may include physically demanding situations. Because of the physically demanding conditions, a user cannot easily control the transmitter manually by, e.g., pressing a tangent, and there is a need to detect the speech activity automatically. The placement of the microphone and the conditions where the device is used results in signals which include a strong breathing sound, which places an extra challenge for VAD algorithms: the breathing sound may have a relatively high level, as illustrated in Figure 1. The devices may also be attached to a breathing apparatus which further amplifies the level of the breathing noise. Since existing VAD algorithms have not been developed to function in the presence of breathing sound, a specific algorithm has to be designed to obtain a good rejection of breathing segments. The only previous study which tries to solve this problem used linear predictive coding derived from a pre-recorded sample of breathing noise, and calculating the prediction error for each frame of the observed signal [4].

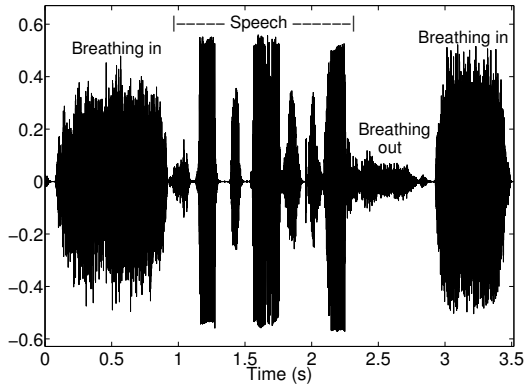


Figure 1: Example signal containing both speech and breathing segments.

This paper proposes an algorithm for VAD in the presence of breathing noise, which is based on combining a discriminatively trained neural network (NN) with a hidden Markov model (HMM). This kind of hybrids of NNs and HMMs have previously been used, e.g., in automatic speech recognition [5]. The block diagram of the proposed algorithm is illustrated in Figure 2. First, the input signal is segmented into 20 ms non-overlapping frames and acoustic features are calculated in each frame. The features are fed into a neural network which does non-linear mapping in order to make the speech and non-speech frames linearly separable. The output of the NN is then modeled using a HMM with Gaussian emission densities in order to take into account the temporally continuous nature of speech segments. Finally, thresholding is used to give the final classification result for the frame. An advantage of the algorithm is that its parameters do not require manual tuning, but it can be automatically optimized using recorded signals. This approach proved to be very efficient and the algorithm provided good separation between speech and noise frames.


Figure 2: Block diagram of the proposed VAD algorithm.

The structure of the paper is as follows: Section 2 presents the feature extraction, Section 3 the neural network and Section 4 the hidden Markov model and the final classification. Section 5 presents the simulations and Section 6 conclusions.

## 2. FEATURE EXTRACTION

The input of the algorithm is a digital signal having sampling frequency of 8 kHz. First the signal is prefiltered with a high-pass FIR filter defined by the equation

$$\hat{s}(n) = s(n) - 0.97s(n-1), \tag{1}$$

where $s(n)$ is the value of the speech signal at the time $n$ and $\hat{s}(n)$ is the corresponding filtered value. The filtering is done in order to flatten the spectrum of the input signal and to remove possible low-frequency noise. After the prefiltering the speech signal is processed and features extracted in 20 ms frames. Here we present the feature extraction which take place within a single frame, and omit the frame index for clarity.

We tested several features which have been previously used in VAD systems, including the frame energy and zero crossing rate, the amount of periodicity [6, 7], an entropy measure calculated from the magnitude spectrum of the frame [8], the linear prediction coding coefficients and the prediction error [1], and cepstral features [9]. All the aforementioned features and different combinations of them were evaluated. However, the best results were obtained using frequency-band energies (used, e.g., in [10]). They also have a low computational complexity.

In this work we use 20 triangular bands spaced linearly on the mel frequency scale, so that adjacent bands overlap by 50%. The energy within each band is obtained by the fast Fourier transform and summing the corresponding squared transform coefficients.

The energy $E_{\text{lin}}(k)$ of the $k$th frequency band is further converted to decibel scale as

$$E_{\text{dB}}(k) = 10 \cdot \log_{10}(E_{\text{lin}}(k) + \varepsilon),\ k = 1, \ldots, K, \tag{2}$$

where $K = 20$ is the number of frequency bands, and $\varepsilon$ is a small constant value. The decibel scale compresses the dynamic range of the input and $\varepsilon$ prevents small values caused by noise from affecting the feature. Here we used $\varepsilon = 2 \cdot 10^{-5}$, while the average level of $E_{\text{lin}}(k)$ was about 10.

The resulting features $E_{\text{dB}}(k),\ k = 1, \ldots, K$ are further normalized to zero mean and unity variance using the means and variances obtained from the training set (see Section 5). The output of the feature extraction stage is a vector of normalized frequency-band energies $E_{\text{dB}}(1), \ldots, E_{\text{dB}}(K)$ for each frame, which is then used as an input for a neural network.

## 3. NEURAL NETWORK

The power of NNs lie in their ability to create nonlinear decision functions. We use a three layer (input, hidden and output) feedforward NN, which maps the input feature vector of each frame into a single output. The hidden layer performs a nonlinear transformation of the input feature vector to make the speech/non-speech classes linearly separable in the space of the output variable. The output $y_i$ of the $i$th hidden neuron is calculated from the input feature vector by

$$y_i = G\Big(\sum_{k=1}^{K} E_{\text{dB}}(k) w_{ki} + b_i\Big), \tag{3}$$

where $w_{ki}$ is the weight of the $k$th feature for the $i$th hidden neuron and $b_i$ is the bias of the $i$th hidden neuron. $G$ is the tangential sigmoid transfer function defined as

$$G(x) = \frac{2}{1 + e^{-2x}} - 1,$$

which maps the input values of a hidden neuron to an interval between minus one and one. The NN has a single output $z$, which is a weighted sum of the hidden neuron outputs, defined as

$$z = \sum_{m=1}^{M} y_m s_m + c, \tag{4}$$

where $M$ is the number of hidden neurons, $s_m$ is the hidden to output weight, and $c$ is the output bias.

The parameters of the NN were trained using a database (see Section 5) which contains acoustic signals and annotations of speech segments. The frequency band energies were calculated framewise for every signal in the database. The target value of the NN output for a feature vector was one if a frame was annotated as a speech frame, and zero otherwise. The NN was trained using the Quasi-Newton Backpropagation algorithm [11, Chapter 2] which minimizes the squared error between the NN output and the target output. We used the NN implementation included in the Matlab Neural Networks Toolbox[1].

In addition to NNs we also tested a Gaussian mixture model (GMM) and a support vector machine (SVM) for the classification of the feature vector of a single frame. The SVM was found to be impractical for our purposes since it was computationally too complex. The framewise classification results (see Section 5) obtained with GMMs were comparable to those obtained with NNs, but the post-processing explained in the next section improved the performance of NNs more, and therefore a NN was used in proposed the system.

## 4. HIDDEN MARKOV MODEL

The output of the NN is further processed in order to take into account the property that speech utterances appear in sequences of consecutive frames. The input signal is modeled using a two-state hidden Markov model (HMM), consisting of speech and noise states. The speech state models speech frames and the noise state models silence, breathing, and other noise sources. HMMs have been previously used in VAD systems for example in [2, 12]. Other methods which model the temporal continuity include the "hangover"-method [13, 10], which classifies a predefined number of frames succeeding a detected speech frame also as speech.

Let us denote the speech state by "s" and the noise state by "n". In the HMM, the probability of a state in a particular time frame depends only on the state probabilities in the previous frame, and the output of the NN in that particular frame. For the following processing, four state transition probabilities have been calculated beforehand:

[1] http://www.mathworks.com/products/neuralnet/

1. probability, that the current frame is speech, given that the previous frame was speech ($a_{\mathrm{s|s}} = 0.982$)
2. probability, that the current frame is speech, given that the previous frame was noise ($a_{\mathrm{s|n}} = 0.002$)
3. probability, that the current frame is noise, given that the previous frame was speech ($a_{\mathrm{n|s}} = 0.018$)
4. probability, that the current frame is noise, given that the previous frame was noise ($a_{\mathrm{n|n}} = 0.998$)

These above constants have been calculated from the training data by finding the fraction of the frames that corresponds to the given conditions.

Let us use $p_{t-1}(\mathrm{s}|z_{t-1})$ to denote the posterior probability of a speech state in frame $t-1$, when the output of the NN has been observed in that frame and all previous frames, and similarly $p_{t-1}(\mathrm{n}|z_{t-1})$ to denote the posterior probability of a noise state in frame $t-1$. The prior probability $p_t(\mathrm{s})$ of the speech state (when the NN outputs have been observed in all previous frames but not in frame $t$) and the prior probability $p_t(\mathrm{n})$ of the noise state in frame $t$ are then given as

$$p_t(\mathrm{s}) = p_{t-1}(\mathrm{s}|z_{t-1})a_{\mathrm{s|s}} + p_{t-1}(\mathrm{n}|z_{t-1})a_{\mathrm{s|n}}$$
$$p_t(\mathrm{n}) = p_{t-1}(\mathrm{s}|z_{t-1})a_{\mathrm{n|s}} + p_{t-1}(\mathrm{n}|z_{t-1})a_{\mathrm{n|n}}.$$

The NN output class-conditional probability density function (pdf) for the speech state is denoted by $p_t(z_t|\mathrm{s})$, which is modeled by Gaussian distribution with mean $\mu_{\mathrm{s}} = 1$ and variance $\sigma^2 = \frac{1}{2}$, and the NN output class-conditional probability density function for the noise state is denoted by $p_t(z_t|\mathrm{n})$, which is modeled by Gaussian distribution with mean $\mu_{\mathrm{n}} = 0$ and variance $\sigma^2 = \frac{1}{2}$. We also tried other distributions instead of the Gaussian distributions, for example GMMs where the parameters were estimated from the training data, but the above Gaussian distributions produced the best results.

The posterior probability that a state is speech is obtained by the Bayes equation as follows [14, pp.179-224]:

$$\begin{aligned} p_t(\mathrm{s}|z_t) &= \frac{p_t(z_t|\mathrm{s})p_t(\mathrm{s})}{p_t(z_t|\mathrm{s})p_t(\mathrm{s}) + p_t(z_t|\mathrm{n})p_t(\mathrm{n})} \\ &= \frac{1}{1+\exp(-a)}, \end{aligned} \tag{5}$$

where

$$a = \ln \frac{p_t(z_t|\mathrm{s})p_t(\mathrm{s})}{p_t(z_t|\mathrm{n})p_t(\mathrm{n})}. \tag{6}$$

Using the above-defined Gaussian distributions to model the speech and noise class-conditional pdfs, $p_t(z_t|\mathrm{s})$ and $p_t(z_t|\mathrm{n})$, respectively, Equation (6) can be simplified to

$$\begin{aligned} a &= \ln \frac{\frac{1}{\sigma\sqrt{2\pi}} \exp[-\frac{1}{2\sigma^2}(z_t - \mu_{\mathrm{s}})^2] p_t(\mathrm{s})}{\frac{1}{\sigma\sqrt{2\pi}} \exp[-\frac{1}{2\sigma^2}(z_t - \mu_{\mathrm{n}})^2] p_t(\mathrm{n})} \\ &= \ln \frac{\exp[-(z_t-1)^2] p_t(\mathrm{s})}{\exp[-z_t^2] p_t(\mathrm{n})} \\ &= 2z_t - 1 + \ln \frac{p_t(\mathrm{s})}{p_t(\mathrm{n})}. \end{aligned} \tag{7}$$

By substituting Eq. (7) back to Eq. (5) we can write the posterior probability of the speech state in frame $t$ as

$$p_t(\mathrm{s}|z_t) = \frac{1}{1+\exp[1-2z_t - \ln(\frac{p_t(\mathrm{s})}{p_t(\mathrm{n})})]} \tag{8}$$

The posterior probability $p_t(\mathrm{n}|z_t)$ of the noise state is obtained simply by $1 - p_t(\mathrm{s}|z_t)$. Once the posterior probability of the speech state in a frame has been calculated, the final decision is made by using a fixed threshold $T$:

$$\begin{cases} p_t(\mathrm{s}|z_t) \geq T & \longrightarrow \text{speech} \\ p_t(\mathrm{s}|z_t) < T & \longrightarrow \text{noise} \end{cases} \tag{9}$$

Here we use threshold $T$ instead of a fixed value 0.5 to allow tuning the sensitivity of the algorithm easily. In practice $T$ can be tuned manually, but in our simulations it was tuned to maximize the performance on the test data.

The proposed VAD algorithm is relatively simple: in addition to the calculations of the fast Fourier transform it consist approximately of 27000 multiplications, 26000 summations, 1050 logarithms, 550 exponents and 50 comparisons per second and it does not require information of the speech signal in upcoming frames. Therefore it does not produce a significant delay in the processing of the speech signal and the VAD can effectively be used in real time.

## 5. SIMULATIONS

Simulations were conducted using a communication device having the microphone in front of the speaker's mouth. The device is usually used in physically demanding situations and the signals recorded by the device have a high level of breathing. On the other hand, the device is not very sensitive to external noise signals.

To evaluate the performance of the algorithm, a database of acoustic signals was recorded in conditions that correspond to the device's intended environment of use. The database consists of microphone signals recorded from four men and one woman. The total amount of data is approximately 2 hours 10 minutes. There are 2-3 signals from each speaker with a signal length of approximately 3-10 minutes. The percentage of speech in the signals is approximately 2-20% depending on the speaker. The recorded signals were manually labeled into speech and noise segments with a temporal resolution of 10 ms.

The performance evaluation of the VAD algorithm was done using the leave-one-out cross-validation method where the signals of one speaker were regarded as a test set and the rest as the training set. A NN was trained using the acoustic data and the annotations in the training set using the Quasi-Newton Backpropagation algorithm. The test signals were processed using the VAD algorithm, which produces speech/noise decision for each frame.

The classification accuracy was measured by comparing the classifications to the annotated speech activity. The following four measures were used to judge the classification accuracy:

- *Sensitivity* gives the percentage of the frames correctly classified as speech from all the speech frames in the signal
- *Specificity* gives the percentage of the frames correctly classified as noise from all the noise frames in the signal
- *Positive predictive value* gives the percentage of the frames that actually are speech from all the frames classified as speech
- *Negative predictive value* gives the percentage of the frames that actually are noise from all the frames classified as noise

The higher the sensitivity, the less speech frames are classified as noise, which is the most important requirement in this study, because classifying speech frames as noise degrades the intelligibility of the transmitted speech signal. On the other hand, classifying noise frames as speech is not so a critical error and therefore a high specificity value is less important than a high sensitivity value. Because noise frames are more easily classified as speech than the other way round, the positive predictive value is relatively low while the negative predictive value is very high.

In addition to the proposed algorithm, four alternative algorithms were also tested:

- NN+GMM+HMM algorithm models the NN output class-conditional pdfs with GMMs instead of the single Gaussians used in the proposed algorithm. HMMs are used to model the temporal continuity.
- GMM+HMM algorithm uses GMMs directly for the feature vectors for the speech and noise classes, and HMM for post-processing.
- NN algorithm uses the output of the NN without HMM post-processing.
- GMM algorithm trains GMMs for the feature vectors for the speech and noise classes and does not use post-processing.

Each algorithm was tested by the leave-one-out procedure where the above measures were calculated for each test speaker at time, and the results were averaged to give the final results. For all the algorithms the threshold $T$ was tuned so that the average sensitivity for the test set was above 97%, since this was found to be sufficient for retaining the intelligibility of the speech, and the specificity was tuned as high as possible. The final average results for the tested methods are illustrated in Table 1.

| Algorithm | Sens. | Spec. | PPV | NPV |
|---|---|---|---|---|
| Proposed | 97.4 | 95.2 | 69.2 | 99.4 |
| GMM modeling the pdf | 97.0 | 82.8 | 47.7 | 99.4 |
| GMM + HMM | 97.1 | 57.2 | 26.1 | 99.2 |
| NN without HMM | 97.2 | 43.6 | 21.0 | 99.0 |
| GMM without HMM | 97.0 | 44.5 | 21.6 | 99.0 |

Table 1: VAD algorithm results (%)

The proposed algorithm has sensitivity of 97.4% and specificity of 95.2%, which are very satisfactory results. Informal listening tests where the detected non-speech frames had been removed showed that the speech intelligibility was not affected and only short bursts of (breathing) noise could occasionally be heard. Therefore, the proposed VAD can provide a significant reduction in battery usage in a communication device.

Modeling the class-conditional output of the NN using a GMM which parameters were fitted to the training data did not produce as good results as single Gaussians used in the proposed system. This is partly explained by the parameter estimation procedure where the parameters of the GMMs and the HMM were estimated separately. The NN classifier offers significantly higher specificity than the GMM classifier as illustrated by the first and third rows of Table 1. The performance of all the tested methods is clearly improved by using the HMM postprocessing. When the HMM postprocessing is not utilized, using either NN or GMM for classifying the frames provide equally good results as is illustrated by the two lowest rows in Table 1.

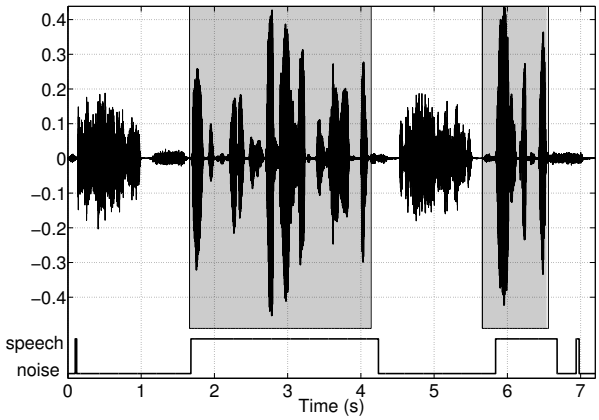


Figure 3: Example of the classification results obtained with the proposed algorithm: gray/white background = annotated as speech/noise, high/low discrete value = classified as speech/noise.

Figure 3 illustrates the results of the classification obtained with the proposed algorithm. The background is gray, when the signal has been annotated as speech, and white, when the signal has been annotated as noise. The discrete value below the signal indicates a speech decision, when it is high, and noise decision, when it is low. The figure illustrates very well the most common mistakes made by the VAD algorithm: classifying noise as speech after a speech segment and classifying speech as noise at the beginning of a speech segment. The former is due to the postprocessing, which makes the posterior probability of speech decrease slowly after the speech segment, which in turn leads to false speech detections. The latter is the same but in reverse, the posterior probability of speech does not increase fast enough for the algorithm to detect speech at the beginning of the speech segment. The figure also shows some random errors made by the algorithm.

It should be noted here, that the sensitivity value could easily have been made higher by adjusting the threshold $T$ as illustrated in Figure 4. This was not necessary, however, because a large percentage of the incorrect classifications that cause the sensitivity to drop, are a result of the fact that the algorithm sometimes classifies short intervals of silence during speech as noise, although they have been annotated as speech. The 97% sensitivity limit does not degrade the intelligibility of the transmitted speech, which is the most important requirement. Furthermore, increasing the sensitivity decreases the specificity significantly after the 97% sensitivity limit as can be seen in Figure 4.

## 6. CONCLUSIONS

We have proposed a novel algorithm for voice activity detection in the presence of breathing noise. The proposed method combines successfully a discriminatively trained neural network and hidden Markov model postprocessing. The neural network maps the input acoustic feature vector nonlinearly to allow better distinguishing speech from breathing

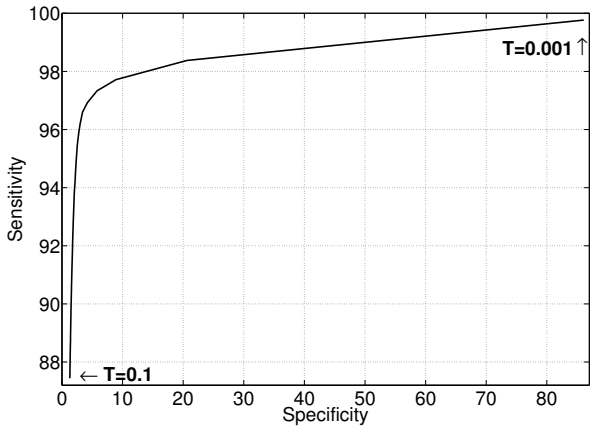


Figure 4: ROC-plot of the algorithm performance.

and other noise sounds, and the HMM provides a probabilistic framework for modeling the temporally continuous nature of speech activity. Simulations with realistic acoustic signals show that the proposed method allows average frame-level sensitivity above 97% and specificity above 95%. The algorithm is able to effectively reject most of the non-speech frames while retaining the speech intelligibility.

## REFERENCES


[1] B. S. Atal and L. R. Rabiner, "A pattern recognition approach to voiced-unvoiced-silence classification with applications to speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, no. 3, 1976.

[2] S. Gazor and W. Zhang, "A soft voice activity detector based on a laplacian-gaussian model," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, 2003.

[3] A. Kondoz, *Digital Speech – Coding for Low Bit Rate Communication Systems*. John Wiley & Sons, Ltd, Chichester, 2004.

[4] W. M. Kushner, M. S. Harton, and R. J. Novorita, "The distorting effects of SCBA equipment on speech and algorithms for mitigation," in *EUSIPCO*, 2005.

[5] H. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*. Kluwer Academic Publishers, Boston, 1994.

[6] J. Rouat, Y. C. Liu, and D. Morisette, "A pitch determination and voiced/unvoiced decision algorithm for noisy speech," in *Speech Communication*, vol. 21, pp. 191–207, 1997.

[7] R. Tucker, "Voice activity detection using a periodicity measure," in *IEE Proceedings I*, vol. 139, pp. 377–380, 1992.

[8] P. Renevey and A. Drygajlo, "Entropy based voice activity detection in very noisy conditions," in *Eurospeech, Aalborg*, pp. 1887–1890, 2001.

[9] J. A. Haigh and J. S. Mason, "Robust voice activity detection using cepstral features," in *IEEE Tencon*, vol. 3, pp. 321–324, 1993.

[10] K. Srinivasant and A. Gersho, "Voice activity detection for cellular networks," in *Speech Coding for Telecommunications*, pp. 85–86, 1993.

[11] S. Haykin, *Neural Networks, A Comprehensive Foundation*. Pearson Education, New York, 2nd ed., 2004.

[12] J. Sohn, N. S. Kim, and W. Sung, "A statistical-model based voice activity detection," *IEEE Signal Processing Letters*, vol. 6, no. 1, 1999.

[13] F. Beritelli, S. Casale, and A. Cavallaro, "A robust voice activity detector for wireless communications using soft somputing," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 9, 1998.

[14] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.