Semi-Supervised Speech Activity Detection with an Application to Automatic Speaker Verification

Alexey Sholokhov[a,b,c], Md Sahidullah[a,*], Tomi Kinnunen[a]

[a] *School of Computing, University of Eastern Finland, P.O. Box 111, FI-80101 Joensuu, Finland*
[b] *ITMO University, Kronverkskiy pr. 49, St. Petersburg, 197101, Russia*
[c] *St. Petersburg Electrotechnical University, ul. P. Popova 5, St. Petersburg, 197376, Russia*

Abstract

We propose a simple speech activity detector (SAD) based on recording-specific Gaussian mixture modeling (GMM) of speech and non-speech frames. We extend the conventional expectation-maximization (EM) algorithm for GMM training using semi-supervised learning. It provides a methodology to incorporate unlabeled data into the SAD training process, leading to more accurate statistical models by exploiting the structure of data distribution. It fits naturally to off-line applications that may require partial human assistance, or applications that involve processing large quantities of audio data, such as text-independent speaker verification, speaker diarization or audio surveillance. The proposed SAD does not require any off-line training data as supervised SADs do. Rather, it employs initial labels produced from a tiny fraction of a given audio recording with the help of another simpler SAD (or a human operator). The proposed SAD is analyzed for the different covariance types, the initialization methods for speech and non-speech class, the amount of labeled data required for initialization, and the speech features. In experiments with a stand-alone SAD system, we observe increased accuracy on the challenging dataset from the recent NIST OpenSAD evaluation. Our extensive automatic speaker verification (ASV) experiments, including text-independent experiments with NIST

*Corresponding author
Email addresses: sholohov@speechpro.com (Alexey Sholokhov), sahid@cs.uef.fi (Md Sahidullah), tomi.kinnunen@uef.fi (Tomi Kinnunen)

2010 speaker recognition evaluation (SRE) data and text-dependent experiments with RSR2015 and RedDots corpora, show benefits of the new approach for the long speech segments containing non-stationary noise. For the shorter data conditions in the text-dependent experiments, simpler unsupervised SADs perform however better. Further, we study the impact of SAD misses and false alarms to ASV performance on the NIST 2010 SRE data. By deriving an empirical cost function with the two SAD errors, we have observed that ASV error rate reaches a minimum value around the same SAD operating point irrespective of SAD method and signal-to-noise ratio (SNR). The optimum ASV performance occurs approximately at an SAD operating region where falsely included non-speech is considered 4 to 5 times more costly than missed speech. Importantly, the proposed semi-supervised SAD is relatively less dependent on the SAD decision threshold compared to the other contrastive SAD methods.

Keywords: Speech activity detection, Semi-supervised learning, Gaussian mixture model, Speaker recognition, NIST OpenSAD, NIST SRE.

---

1. Introduction

Speech activity detection (SAD) [1], the task of locating speech segments in a given recording, plays a key role in any speech processing system, including coding [2] and speaker recognition applications [3], to prevent unnecessary processing of non-speech segments. A large number of methods has been studied to solve the SAD problem. Classic digital signal processing (DSP) methods first compute scalar features such as short-term energy, zero-crossing rate [4], periodicity [5] or spectral divergence [6], and compare these values against fixed thresholds to classify audio segments.

Though being simple and providing sufficient performance on clean conditions, the accuracy of these methods rapidly degrades with low signal-to-noise ratios (SNRs). To address this problem, a variety of *statistical model-based* approaches have been explored [7, 8, 9]. These methods assume that the spectral coefficients follow a particular parametric distribution. The SAD decision is

sought by calculating the likelihood ratio based on the hypothesized models. For instance, [7] models the spectra of the noisy speech and noise using complex Gaussian distributions, while [8] and [9] adopt, respectively, Laplacian and generalized Gamma models. The statistical methods often outperform the classic methods in the presence of stationary noise, but non-stationary noise conditions remain challenging.

The above classic SADs are **unsupervised** as they do not involve a separate training process. In contrast, **supervised** methods, based on machine learning (ML), have recently yielded promising results [10, 11, 12, 13, 14]. By leveraging from prior knowledge in large annotated audio collections, these SADs can partially cope with the aforementioned problem of non-stationary noise. Similar to other machine learning tasks, however, supervised SADs tend to be sensitive to acoustic mismatch between the training and test conditions [15]. The authors of [16] made a step towards noise-independence by training a universal model for clean speech to detect the presence of speech in noisy signal. Their method, however, assumes noise additivity which may be violated in the presence of additional channel mismatch.

Adaptive supervised SADs, such as [17] and [18] represent a compromise between the powerful supervised approaches, such as neural networks, and statistical model-based methods which require no prior training but whose parametric modeling assumptions might be over-simplistic. For instance, [18] studied a heuristic SAD not requiring any prior training data beyond the given utterance. It first uses an auxiliary energy-based SAD to label a small portion of speech frames, used for training two codebooks to represent feature distributions of speech and nonspeech, which are then used to classify all the frames. This approach is simple and, in principle, relatively insensitive of the type of noise (assuming the initial labeling can be done reliably). The authors in [18] used energy-based SAD for the initial frame labeling and standard mel-frequency cepstral coefficient (MFCC) features as the features in the final SAD.

In this work, we extend the heuristic method in [18] using a more principled *semi-supervised learning* (SSL) [19] paradigm. SSL considers classification set-

ups where, for reasons such as time, cost or human labor, only a small fraction of the available training data has labels. In conjunction with the labeled data, SSL-based methods take use of unlabeled data to train robust models, often surpassing the accuracy of models trained using only the small labeled data sample [20]. This is a natural approach for large-scale off-line applications not requiring real-time operation, such as speaker diarization, speech data mining or forensic audio annotation or any other cases where it is convenient or cost-effective to produce an initial labeling from very small amount of data, either by a human annotator or another SAD that is known to work reasonably well.

In specific, we modify the conventional expectation-maximisation (EM) algorithm to train GMMs with only partially labeled data. Semi-supervised training of GMMs, as such, has been studied elsewhere, including image segmentation [21], audio classification [22] and instrument recognition [23]) tasks but the authors are unaware of its prior use in the context of SAD. Besides providing a principled probabilistic formulation of the problem instead of ad-hoc derivation in [18], we provide a detailed account into how the type of covariance matrices and the quality of initial frame labeling influences SAD accuracy, none which were addressed in [18]. In general, our SSL-based SAD provides a straightforward and simple approach to the SAD *without requiring any manual labeling effort at any stage*.

In the experimental part of this work, we first analyze the performance of the proposed semi-supervised GMM using both stand-alone SAD accuracy on two datasets, including telephony/microphone and radio-phone data from NIST SRE 2010 speaker recognition corpus [24] and the recent NIST OpenSAD challenge benchmark data [25]. We then demonstrate the benefit of integrating the proposed SAD into state-of-the-art automatic speaker verification (ASV) task including both text-independent and text-dependent scenarios, evaluated with the help of the NIST SRE 2010, RSR2015 and recent RedDots corpus.

Finally, as a key contribution of our work, we provide a novel analysis on the relative importance of miss and false alarm rates of the SAD to ASV. Due to the difficulty to relate the errors made by an SAD to a full ASV system consisting of

multiple front-end and back-end processing modules, SAD thresholds are usually determined using trial and error or *ad-hoc* rules that are not explicitly specified. There are, however, obvious benefits to define an explicit SAD segmentation objective to maximize ASV performance: it can lead to greatly reduced labor or optimization work required on new databases and, importantly, it can shed light into how one should trade-off SAD misses and false alarms in an ASV context in general. Having these broad goals in our mind, we present, for the first time, a methodology to derive an empirical cost function that interlinks SAD segmentation errors to the ASV errors.

2. The Role of SAD in Speaker Recognition

Study on the role of SADs in speaker recognition context is surprisingly limited [3, 18, 26]. In general, the speaker recognition community has paid relatively little attention to the problem as many on-the-shelf SADs perform more or less similarly with speech signals of relatively high signal-to-noise ratios (SNRs) on various standard corpora [3]. Typically, an energy-based SAD [27] performs quite well on earlier NIST evaluation corpora. However, the choice of the SAD becomes more critical for ASV with highly degraded speech, especially in the presence of additive noise [26]. A simple utterance-dependent threshold-based energy SAD produces unreliable speech frames in noisy conditions, leading to drastically increased speaker verification error rates. In recent years, we observe a growing interest in this topic especially after the NIST 2012 speaker recognition evaluation (SRE) campaign. The corpus for this evaluation consists of speech files severely distorted with various additive noise, and here, advanced SADs were helpful [28, 29]. Further, recent studies on DARPA RATS corpora having speech files collected from highly degraded radio channels also motivates the need for investigations in SAD algorithm for speaker recognition context [30].

Other than the impact of noise, it is also interesting to study the impact of recording duration to SAD performance. Particularly, this may be a crucial issue for the text-dependent speaker verification where a short-phrase is used for the

test. A limited number of studies have discussed the relative ASV performance for different SADs. In this work, we conduct more extensive study for different SADs on different aspects like text-variations, duration variability in clean and noisy conditions. Further, we also analyze the impact of SAD accuracy on the corresponding speaker recognition performance.

3. SAD with Recording-Specific GMMs

GMMs can be used in a supervised or unsupervised manner for efficient modeling of the speech and non-speech classes [31]. Given an arbitrary audio recording, represented by a sequence of short-term feature vectors $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\} \subset \mathbb{R}^d$, our goal is to train two GMMs, one to represent speech and the other one non-speech. To do so, we need at least partial labels for each of the observations $\mathbf{x}_t$. In [18], a small subset of $\mathcal{X}$ was selected using ranking of the respective frame energy values within the utterance, so that highest- and lowest-ranked frames were assumed to be correspond, respectively, speech and non-speech. Thus, some auxiliary SAD is required to produce initial segmentation for supervised training of class-specific GMMs. Arbitrary SAD providing real-valued scalar scores (*e.g.* log-likelihood ratios) can be used as such.

But acknowledging any imperfection of the initial SAD labels, we are traded-off to choose a small-enough subset of 'reliable' frames, yet large enough to enable numerically robust training of the two GMMs from limited data (in [18], the problem was addressed by using code-books trained by k-means, a highly restricted form of GMMs). To address these issues, we adopt semi-supervised learning to enable taking benefit of *all* the training vectors. Before presenting our approach, we provide a brief review of the closest similar methods.

3.1. Closest prior work on semi-supervised training

In our derivation shown below, we follow the classic approach to estimate parameters of statistical classifiers from labeled and unlabeled data using maximum likelihood criterion. Similar methods have independently been developed

in other applications such as image segmentation [21], audio classification [22] and text classification [32]; in the last study, the authors consider multinomial, rather than Gaussian distribution for observation modeling but the core approach remains otherwise the same. In fact, all these methods can be placed under the umbrella of the general EM framework formalized in the seminal paper [33]. Hence, the resulting algorithms presented in prior literature optimize similar objective functions defined up to the choice of observation model, leading naturally to similar update equations. A short history of semi-supervised training of generative classifiers can be found in [32]. Besides purely generative training strategy mentioned above, the authors of [34] proposed a hybrid generative/discriminative objective function. Our training process can be seen as a special case of their unified approach. In this study, we focus purely on generative training.

3.2. Semi-Supervised GMM

Let us define the terminology and notation. A *class* is the true class of any feature vector — either speech ($\ell = 1$) or nonspeech ($\ell = 2$). The number of classes is L (here, 2) and each of them is modeled using a GMM. We use w_k^ℓ, $\boldsymbol{\mu}_k^\ell$ and $\boldsymbol{\Sigma}_k^\ell$, to denote, respectively, the mixing weight, mean vector and covariance matrix of kth Gaussian in class ℓ. We assume the same number of Gaussians (K) per class to simplify notation but the resulting *expectation-maximization* (EM) update equations are trivial to modify if needed. Our derivation follows exactly the one presented in [21], though, in addition, we present special formulae for shared covariance updates in Subsection 3.3 that was not presented in [21].

Consider first a single feature vector, $\mathbf{x}$, with a *known* class label $y \in \{1, .., L\}$. The class-conditional density function is,

$$p(\mathbf{x}|y) = \sum_{k=1}^{K} w_k^y \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^y, \boldsymbol{\Sigma}_k^y). \tag{1}$$

But if the class label of $\mathbf{x}$ is *unknown*, we have a mixture density,

$$p(\mathbf{x}) = \sum_{\ell=1}^{L} \pi^{\ell} \sum_{k=1}^{K} w_k^{\ell} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k^{\ell}, \boldsymbol{\Sigma}_k^{\ell}), \tag{2}$$

where π^{ℓ} is the class prior. To draw a random $\mathbf{x} \sim p(\mathbf{x})$, we first select the class according to the class prior, a Gaussian from that class following the component prior and finally the Gaussian where $\mathbf{x}$ is then drawn.

In many speech processing systems, including universal background modeling for automatic speaker verification and language identification, large GMMs (often with 1024 or 2048 Gaussians) are used for modeling feature frames pooled from a large number of speakers and utterances. In contrast, the SAD solution sought in this study use recording-specific GMMs to model the speech and non-speech classes. To this end, consider now a training set — a set of frames from a single audio recording of a single session of one speaker — containing N vectors, $\mathbf{X}^a = \{\mathbf{x}_n^a\}_{n=1}^N$ with labels $\mathbf{y} = \{y_n\}_{n=1}^N$ plus M unlabeled vectors $\mathbf{X}^b = \{\mathbf{x}_m^b\}_{m=1}^M$. The union of $\mathbf{X}^a$ and $\mathbf{X}^b$, consisting of $N + M$ vectors, contains all the feature frames of a particular recording. Typically $N \ll M$ due to, for instance, labeling cost associated with human labor. We estimate the model parameters via maximum likelihood (ML). Assuming the training vectors are independent and identically distributed (*i.i.d.*), the logarithm of the joint density function for all observations is,

$$\log p(\mathbf{X}^a, \mathbf{X}^b) = \sum_{n=1}^{N} \log p(\mathbf{x}_n^a|y_n) + \sum_{m=1}^{M} \log p(\mathbf{x}_m^b). \tag{3}$$

Optimal model parameters can be found by maximizing this function. Let us consider the mean vector $\boldsymbol{\mu}_k^{\ell}$ and write down a condition which must be satisfied at the maximum of Eq. (3). Setting partial derivative of the function

with respect to $\boldsymbol{\mu}_k^\ell$ to zero, we obtain

$$\mathbf{0} = \sum_{n \in C^\ell} \underbrace{\frac{w_k^\ell \mathcal{N}(\mathbf{x}_n^a | \boldsymbol{\mu}_k^\ell, \boldsymbol{\Sigma}_k^\ell)}{\sum_j^K w_j^\ell \mathcal{N}(\mathbf{x}_n^a | \boldsymbol{\mu}_j^\ell, \boldsymbol{\Sigma}_j^\ell)}}_{\gamma_k^\ell(n)} {\boldsymbol{\Sigma}_k^\ell}^{-1} (\mathbf{x}_n^a - \boldsymbol{\mu}_k^\ell) +$$
$$\sum_{m=1}^{M} \underbrace{\frac{\pi^\ell w_k^\ell \mathcal{N}(\mathbf{x}_m^b | \boldsymbol{\mu}_k^\ell, \boldsymbol{\Sigma}_k^\ell)}{\sum_\ell^L \pi^\ell \sum_j^K w_j^\ell \mathcal{N}(\mathbf{x}_m^b | \boldsymbol{\mu}_j^\ell, \boldsymbol{\Sigma}_j^\ell)}}_{\xi_k^\ell(m)} {\boldsymbol{\Sigma}_k^\ell}^{-1} (\mathbf{x}_m^b - \boldsymbol{\mu}_k^\ell)$$

where C^ℓ is the set of indices such that $y_n = \ell$. After solving this equation with respect to $\boldsymbol{\mu}_k^\ell$ we obtain[1]

$$\boldsymbol{\mu}_k^\ell = \frac{\sum_n \gamma_k^\ell(n) \mathbf{x}_n^a + \sum_m \xi_k^\ell(m) \mathbf{x}_m^b}{\sum_n \gamma_k^\ell(n) + \sum_m \xi_k^\ell(m)} \tag{4}$$

It should be noted that $\gamma_k^\ell(n)$ and $\xi_k^\ell(m)$ depend on $\boldsymbol{\mu}_k^\ell$, so Eq. (4) is not a closed-form solution for likelihood maximization. However, it suggests an iterative optimization algorithm which alternates between computing $\gamma_k^\ell(n)$ and $\xi_k^\ell(m)$ and updating means according to (4). The same iterative scheme can be directly derived as an instance of the *expectation–maximization* (EM) algorithm [33], a general-purpose method for finding ML parameter estimates of probabilistic models. For further details, refer to [35, Chapter 9]. Similar derivations can be found in [21].

Similar to (4), the update equations for the remaining parameters are obtained:

$$\boldsymbol{\Sigma}_k^\ell = \frac{\sum_n \gamma_k^\ell(n) \boldsymbol{\delta}_n^a + \sum_m \xi_k^\ell(m) \boldsymbol{\delta}_m^b}{\sum_n \gamma_k^\ell(n) + \sum_m \xi_k^\ell(m)} \tag{5}$$

$$w_k^\ell = \frac{\sum_n \gamma_k^\ell(n) + \sum_m \xi_k^\ell(m)}{N_\ell + \sum_m \zeta^\ell(m)} \tag{6}$$

$$\pi_\ell = \frac{N_\ell + \sum_m \zeta^\ell(m)}{N + M}, \tag{7}$$

[1] $\gamma_k^\ell(n) = 0$ when $\mathbf{x}_n^a$ does not belong to class ℓ

where we introduce short-hands $\boldsymbol{\delta}_n^a = (\mathbf{x}_n^a - \boldsymbol{\mu}_k^\ell)(\mathbf{x}_n^a - \boldsymbol{\mu}_k^\ell)^\top$, $\boldsymbol{\delta}_m^b = (\mathbf{x}_m^b - \boldsymbol{\mu}_k^\ell)(\mathbf{x}_m^b - \boldsymbol{\mu}_k^\ell)^\top$ and $\zeta^\ell(m) \triangleq \sum_{k=1}^K \xi_k^\ell(m)$. We use empirical covariance matrix to initialize $\boldsymbol{\Sigma}_k^\ell$ and randomly sample $\boldsymbol{\mu}_k^\ell$ around empirical mean of the whole training set. All mixture weights are initialized to have equal value.

After training the two GMMs with the above iteration, we use *log-likelihood ratio* (LLR), $\Lambda(\mathbf{x}) = \log p(\mathbf{x}|\text{speech}) - \log p(\mathbf{x}|\text{nonspeech})$, as a speech activity indicator for any frame $\mathbf{x}$ in the same utterance. To make the hard SAD decision, we compare LLR with a threshold (which is, for the most part of this study, 0).

3.3. Model interpretation and practical issues

The method presents an interesting compromise between supervised and unsupervised training. On the one hand, if all the data is labeled ($M = 0$), the method boils down to training L class-specific GMMs independently of each other (as *e.g.* in [18]). On the other hand, if there are no labeled samples ($N = 0$) and each class is modelled as a single Gaussian ($K = 1$ for all ℓ), the method is exactly the same as unsupervised training (or clustering) with L-component GMMs [35]. In the general case of more than one Gaussian per class, it can be seen as a K-component GMM with weights $\pi_\ell w_k^\ell$ with the standard formulae to train a conventional GMM. Further, when class distribution is a single Gaussian (in this case $K = 1$ and hence all $w_k^l = 1$), equations (4), (7) and (5) reduce to formulae presented in [21].

Lack of training data (short speech utterances) or high-dimensional features might lead to numerical problems or inaccurately estimated models, making the choice of the covariance matrix type a relevant practical consideration. It may be either full (used as default type in this study), diagonal or spherical. Diagonal covariances are obtained from the Eq. (5) by retaining only the diagonal elements while spherical covariances can be obtained by averaging the elements of a diagonal covariance matrix. The number of covariance parameters per Gaussian in these three cases are $d(d+1)/2$, d and 1, respectively, where d is the feature dimensionality. To constrain the model further, we may also share the covariance matrices across the components of the same class. In this

case, Eq. (5) is modified as,

$$\boldsymbol{\Sigma}^{\ell} = \frac{\sum_{k=1}^{K}\sum_{n}\gamma_k^{\ell}(n)\boldsymbol{\delta}_n^a + \sum_{k=1}^{K}\sum_{m}\xi_k^{\ell}(m)\boldsymbol{\delta}_m^b}{N_{\ell} + \sum_{m}\zeta^{\ell}(m)} \tag{8}$$

In our experiments we did not estimate class mixing proportions π_l – they were set as $\pi_l = 0.5$. By default we used 10% of speech and non-speech frames obtained from auxiliary SAD to train corresponding GMMs. We provide details on initialization strategies in Section 5.3.

Figure 1 shows two stages of the proposed algorithm in the data space (waveform) and feature space (first two MFCCs).

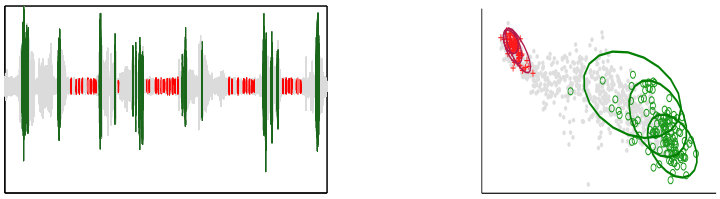
4. Standalone SAD Assessment: Experimental Set-up

Our experiments consists of three major parts. In the first part, consisting of Sections 4 and 5, we assess the proposed SADs in a standalone mode by evaluating their accuracy with respect to a reference speech/nonspeech segmentation. This involves studying the impacts of covariance matrix types, acoustic features and alternative ways to initialize the semisupervised SAD. The second part, described in Sections 6 and 7, consisting of disjoint data from the first part, is then devoted to automatic speaker verification (ASV) experiments, containing both text-independent and text-dependent scenarios. Finally, in Section 8, we analyze the relationship of SAD and ASV in more detail using novel methodology suggested in this study.

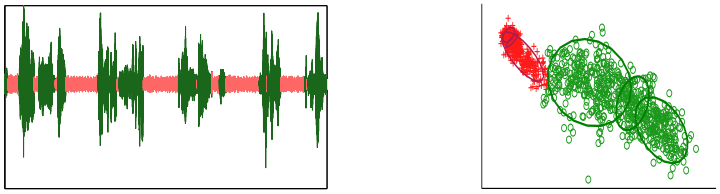
4.1. Datasets

We evaluate SAD performance on two different speech corpora. The first one, utilized in [18], consists of telephony speech degraded by digitally added noise to simulate noisy environments. The second one is a part of the development set in the recent NIST OpenSAD challenge [25], one of the rare benchmarks specifically targeted at evaluating SAD accuracy[2].

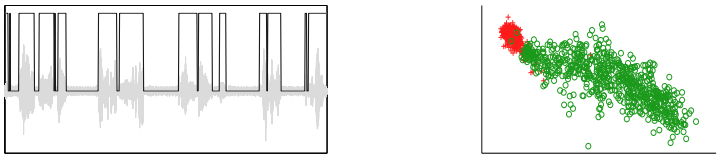
[2]NIST disclaimer: "NIST serves to coordinate the NIST OpenSAD evaluations in order to support speech activity detection research and to help advance the state-of-the-art in speech



(a) Supervised initialization



(b) Final result after 20 iterations



(c) Ground truth.

Figure 1: Semisupervised SAD in time (left) and MFCC (right) domains (first two cepstral coefficients are shown) for $K = 3$ Gaussians per class. Gray color in (a) represents unlabeled data, while green/red points are labeled speech/nonspeech frames. The ellipses represent the individual Gaussians in speech and nonspeech models. Best viewed in color.

SRE10: We utilize telephone utterances from the development set of the NIST 2010 speaker recognition evaluation (SRE) campaign. This development set, originally provided by NIST, contains 36 two-channel recordings with supplementary automatic speech recognition (ASR) transcripts with an average duration of 5 minutes per file. We manually audited both sides of the calls

activity detection technologies. NIST OpenSAD evaluations are not viewed as a competition: as such, results reported by NIST are not to be construed, or represented, as endorsements of any participant's system, or as official findings on the part of NIST or the U.S. Government". Web page: http://www.nist.gov/itl/iad/mig/opensad_15.cfm

($36 \times 2 = 72$ unique recordings) to ensure a controlled SAD development set, leading to rejection of 9 files without speech content[3]. This set of 63 files was then downgraded with controlled signal-to-noise ratios (SNRs) ranging from 0 dB to 20 dB using G729 audio weighting filter for speech level determination via *Filtering and Noise Adding* (FaNT) tool[4]. Eight noise files were selected from `FreeSound`[5], containing mostly device sounds in home environments (e.g. airconditioner and washing machine). To degrade an arbitrary file, a random long noise file was first selected with further random section selection within that file. SAD accuracy is evaluated by comparing the predicted SAD labels with a reference segmentation obtained from the ASR word-level transcripts. As noted in [18], this ground truth is not perfect as some speech missed by an SAD may originate from speech-internal pauses considered to be continuous chunks of speech according to the word-level ASR transcripts. Nonetheless, in our experience it suffices rather well for SAD optimization and complements our other evaluations detailed below.

OpenSAD: The recently conducted NIST OpenSAD evaluation was intended for advancing state-of-the-art in SAD for signals with highly degraded conditions. The OpenSAD data originates from one of the DARPA RATS (Robust Automatic Transcription of Speech) evaluation sets [36]. It consists of highly degraded recordings obtained by transmitting the source audio over several different noisy radio communication channels. A *push-to-talk* protocol was used in all the transmission channels except one. The data was provided along with ground-truth annotations produced by an automatic SAD and further audited by a human annotator. In our experiment we used the *dev-2* subset of the official *development* part[6]. We included six channels, namely {`B, D, E, F, G, H`}, in our evaluation set, resulting in 661 audio recordings with an average duration of 10 minutes. The same set of files was used as the development

[3] An earlier study [18] using the same data used all the 72 recordings.
[4] `http://dnt.kr.hsnr.de/download.html`
[5] `http://www.freesound.org`
[6] `LDC2015E97_NIST_OpenSAD15_Development`

set during the NIST OpenSAD Challenge. The proposed semi-supervised SAD was used as one of the sub-systems of the "HAPPY" team submission to the challenge [37].

4.2. Proposed SAD: the Choice of Acoustic Features

The proposed SAD in Section 3 is general and was described without any reference to the acoustic features $\{\mathbf{x}_t\}$ used for training the speech and non-speech GMMs. In this work, we study the role of features for speech/nonspeech discrimination since the proposed method allows easy plug-in of arbitrary multi-dimensional features, some of them potentially more robust in relative terms against degradations induced by noise, or in terms of selecting speech frames more relevant for speaker discrimination. In the prior work [18], conventional mel-frequency cepstral coefficients (MFCCs) were adopted without further elaboration. Before detailing the features selected for our study, we make a brief review of recent work on the role of features in the general SAD context.

There are several previous studies on the role of speech features in a *supervised* SAD context using the DARPA RATS corpus. For instance, in [30], mel filterbank outputs, MFCCs, perceptual linear prediction (PLP) coefficients, cochlear filter cepstral coefficients (CFCCs), frequency domain linear prediction (FDLP) features, cortical features, pitch contours and energy contours were studied. In another study [38] using the same RATS corpus, MFCCs were combined with different spectral information based on Gabor feature representation, voicing features, spectral flux measures, subband auto-correlation, multiband combo filter and F0 voicing measure to enhance SAD performance. Furthermore, rate-scale feature, based on spectro-temporal modulation filtering of the auditory was investigated along with PLP, FDLP and log-mel spectral features in [39]. In [40], robust front-end using contextual and discriminative information was investigated for SAD purposes. Voicing features [14] representing periodicity of the speech signal are also used in an integrated manner with other spectral features to boost SAD accuracy.

Studies on speech features in an *unsupervised* SAD context, however, are

very limited, and they use typically log-mel filterbank coefficients [41, 31]. This motivates us to re-assess some of the most representative features that are known to vary in their robustness. We provide a comparative evaluation of the GMM-based SADs using the following alternative feature sets, all adopted from public (open-source) implementations for reproducibility. The short-term framing settings are the same across all the compared SADs: speech is segmented into 20 ms frames every 10 ms, *i.e.*, with 50 % overlap.

Mel-frequency cepstral coefficient (MFCC): The standard MFCC features are computed by passing the signal through a triangular filterbank placed in non-linear Mel-frequency scale. The outputs of the filterbank energies are logarithmically compressed followed by discrete cosine transform (DCT) to obtain the final feature vector. We use 27 filters to extract the MFCCs and retain the lowest 12 coefficients (including the energy coefficient) as our features. We use the MFCC implementation provided in VQVAD package[7].

Power normalized cepstral coefficient (PNCC): PNCC features have yielded promising results in several speech processing tasks including speech [42] and speaker [43] recognition. The computational steps are similar to those of MFCCs, one of the differences being the use of gammatone filter for frequency integration. Additionally, it includes an environmental noise compensation scheme with the help of *asymmetric noise suppression* algorithm [42]. Furthermore, power function non-linearity is used instead of logarithm for energy compression. We adopt a publicly available implementation using its default parameter settings[8]. In specific, we extract 19-dimensional PNCCs using 32 gammatone filters.

Perceptual linear prediction (PLP) coefficients: PLP is another commonly adopted feature for speech/non-speech detection in RATS evalua-

[7] http://cs.uef.fi/pages/tkinnu/VQVAD/VQVAD.zip

[8] http://www.cs.cmu.edu/~./mharvill/RATS/software_releases/PNCC/PNCC_deployed_v6/

tion [30]. It emulates different processing blocks inside the human auditory system. First, the speech frames are processed by trapezoidal filterbank with center frequency spacing of 1 Bark, followed by equal loudness weighting of the filterbank outputs, cube-root compression and inverse discrete Fourier transform (IDFT) to obtain linear predictive coefficients further transformed to cepstral coefficients. We extract 13 PLP coefficients followed by RASTA filtering using the MATLAB implementation of [44]. The linear prediction based smoothing was not used, instead, the energy compressed filterbank output are directly converted into cepstrum using DCT.

Frequency domain linear prediction (FDLP): Another recently proposed feature, FDLP [45], exploits the dual properties of time and frequency by approximating the temporal envelope of a signal using linear prediction analysis in spectral, rather than time domain. We use the publicly available implementation[9]. We extract 13 features using a long-term analysis window of 10 s and all-pole model order for the temporal envelope is set at 33. We apply gain normalization on subband envelopes as it improves the robustness.

4.3. Proposed SAD: GMM vs SSGMM back-end

Through out our experiments, we consider two types of models trained using one of the above-described spectral feature sets. The proposed semi-supervised method will be referred to as **SSGMM**. In contrast, a special case of the proposed method, based on purely supervised training without any use of unlabeled data — *i.e.* $M = 0$ in equations (5) - (7) — is simply referred to as **GMM**.

4.4. Performance measures

Following standard evaluation methodology of speech activity detectors, we assess a performance of a SAD by comparing its predicted output (a binary label

[9] http://www.clsp.jhu.edu/~sriram/research/fdlp/fdlp_analyze.zip

for each speech frame) against a reference labeling. Let the boolean vectors $\mathbf{z}_u$ and $\hat{\mathbf{z}}_u$ denote, respectively, the ground truth and predicted SAD labels for an utterance u. We compute average *miss* and *false alarm* (FA) rates:

$$\mathrm{P}_{\mathrm{miss}} = \left(\frac{1}{N_{\mathrm{utt}}} \sum_{u=1}^{N_{\mathrm{utt}}} \frac{\#\{\mathbf{z}_u = 1 \wedge \hat{\mathbf{z}}_u = 0\}}{\#\{\mathbf{z}_u = 1\}} \right) \times 100\%,$$

$$\mathrm{P}_{\mathrm{fa}} = \left(\frac{1}{N_{\mathrm{utt}}} \sum_{u=1}^{N_{\mathrm{utt}}} \frac{\#\{\mathbf{z}_u = 0 \wedge \hat{\mathbf{z}}_u = 1\}}{\#\{\mathbf{z}_u = 0\}} \right) \times 100\%,$$

where $\#\{\cdot\}$ is a counting function which returns the number of non-zero elements in a boolean vector and N_{utt} is the number of utterances in a dataset. Having estimated the average miss and false alarm rates, SAD performance can also be summarized as a single scalar, *decision cost function* (DCF):

$$\mathrm{DCF}(\alpha) = \alpha \mathrm{P}_{\mathrm{miss}} + (1 - \alpha)\mathrm{P}_{\mathrm{FA}}, \tag{9}$$

where $\alpha \in [0, 1]$ is a fixed weight fixed in advance depending on the intended application and one's belief which error type is more costly. For instance, the official evaluation metric of the NIST OpenSAD evaluation [25] was DCF(0.75), indicating relatively higher penalty for missed speech compared to falsely included nonspeech.

5. Standalone SAD Assessment: Results

5.1. Effect of covariance matrix type

We begin by comparing different settings of covariance matrix structure for SSGMM SAD. As discussed above, the possible choices are full, diagonal and spherical covariances. Further, the covariance matrices can be distinct or shared across all the Gaussians within a class, resulting in 6 possibilities in total. We compare all these on the SRE10 dataset using the MFCC features for different number of Gaussians. As we do not focus on a particular trade-off between misses and false alarms at this stage, we use DCF(0.5) as our objective metric.

Since DCF(0.5) is a special case of *half total error rate* (HTER), we estimate the 95% confidence intervals of this quantity using the methodology of [46].

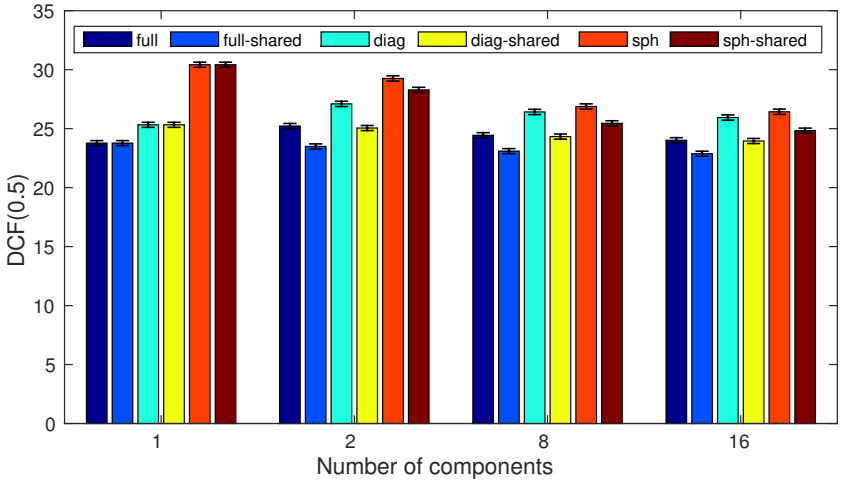


Figure 2: Comparison of models obtained by SSGMM SAD with different covariance matrix structure in terms of DCF(0.5) on the SRE10 set.

Figure 2 reveals that covariance sharing decreases the cost for all the three types of covariance matrices (full, diagonal, spherical) while the number of Gaussian components does not have a considerable effect on SAD performance. The results are shown for the SNR of 15 dB, but the relative order was similar for the other SNR levels, too. Further, Figure 3 shows the detection error trade-off (DET) for the different covariance matrix types. It confirms that the conclusions drawn from Figure 2 hold across a wider range of operating points. For the further stand-alone SAD experiments we choose the model with full non-shared covariance matrices and 8 Gaussians as an arbitrary but representative[10] case.

5.2. Comparison of features

Our next analysis concerns the choice of SAD features. Figure 4 shows a comparison of our four feature sets used by two SADs (SSGMM and GMM)

[10]Most open-source GMM code implementation include non-shared diagonal and full covariance matrix variants but covariance sharing is less common.

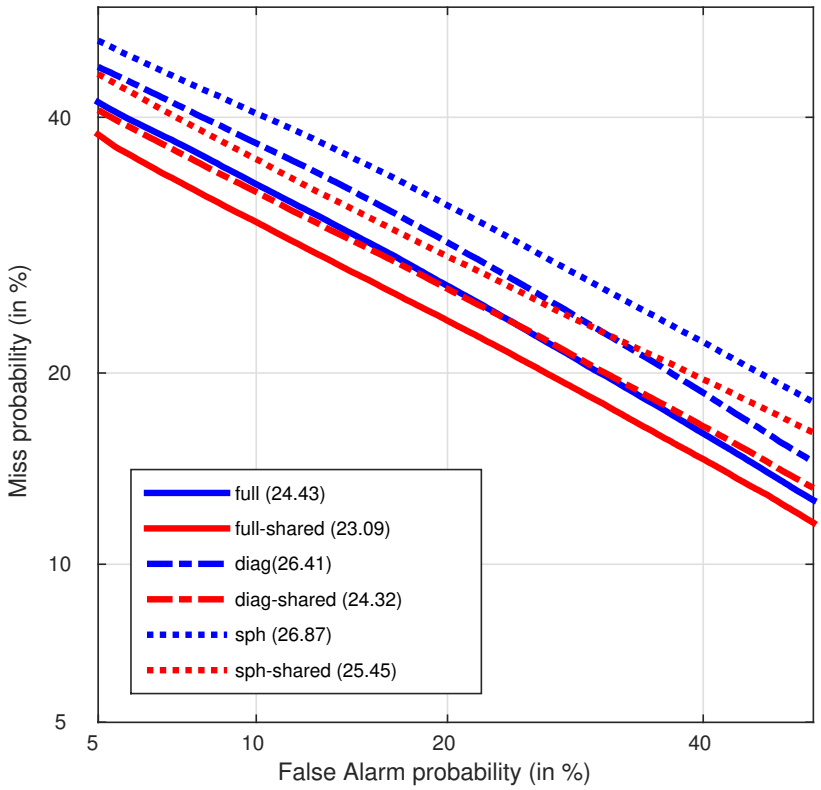
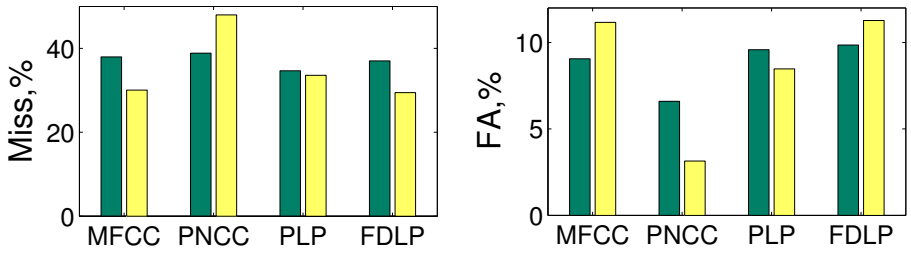


Figure 3: Comparison of models obtained by SSGMM SAD with different covariance matrix structure on the SRE10 set. The number of mixture components is 8. Each point on the curves represents *average* miss rate across files for a given false alarm rate. The corresponding values of DCF(0.5) from Fig. 2 are shown in brackets.

in terms of both miss and false alarm rates on the SRE10 set with an SNR of 15 dB. We make several observations. Firstly, PNCCs yield the lowest false alarm with a trade-off in highest miss rates. Secondly, all the other features (excluding PNCCs) have similar average miss rates with each other. This is likely due to the use of integrated noise reduction scheme in PNCC computation method [42]. Thirdly, PNCC has the lowest average false alarm rate. Finally, the performances of FDLP- and MFCC-based SADs are very close to each other. From these findings, the use of PNCCs might suit better applications where false alarms are more costly — such as automatic speaker verifica-



(a) (b)

Figure 4: Miss (a) and FA (b) rates for different SADs (left to right: GMM, SSGMM) and features on the SRE10 set.

tion; otherwise, MFCCs or PLP features might be a better choice. In all the remaining experiments of this study, we consider only the MFCC and PNCC features.

Table 1: Comparison of features on the NIST OpenSAD set. SAD performance on the NIST OpenSAD set is shown as (Miss % / FA %).

SAD	Miss, %	FA, %
GMM (MFCC)	27.22	16.06
SSGMM (MFCC)	22.23	14.31
GMM (PNCC)	26.88	14.69
SSGMM (PNCC)	37.49	4.21

Table 1 shows a comparison of different SADs based on MFCC and PNCC features on our other SAD dataset, NIST OpenSAD. For the GMM (without semi-supervised training), MFCCs and PNCCs perform similarly. For the semi-supervised variant, however, the false alarm rate is considerably lower but with considerably increased miss rate. This observation agrees with the results on the SRE10 data. Figure 5 shows DET curves to compare performance of two SADs for the case of PNCC and MFCC features. Each row of Table 1 corresponds to a point on these curves. From these results we can presume that SSGMM gives lower error rate on the NIST OpenSAD set using both MFCC and PNCC features.

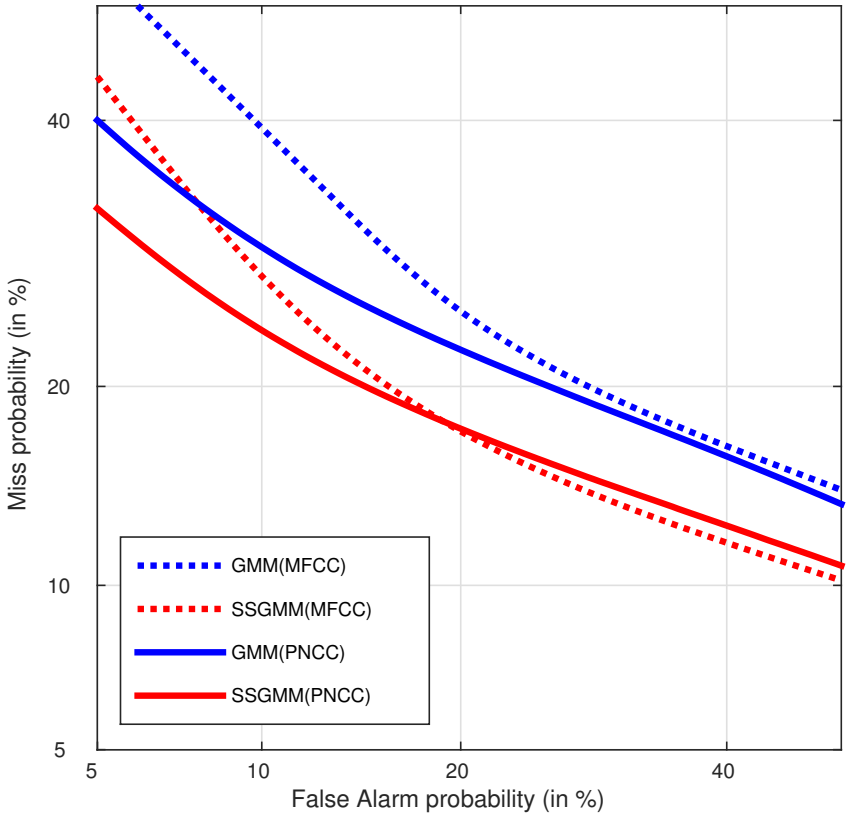


Figure 5: SAD performance on the NIST OpenSAD for PNCC and MFCC features. Each point on the curves represents *average* Miss rate across files for a given FA rate. Each DET curve corresponds to a row of Table 1.

5.3. Effect of initialization

In [18], an energy-based SAD was used to obtain an initial segmentation for training the speech and non-speech models. Short-term signal energy, however, is easily impacted by the presence of background noise, which motivates a more detailed look into the role of initialization accuracy in the GMM-based SADs. We compare three alternatives. The first one is the same energy SAD as in [18] that uses 10% of frames with the highest and lowest energies within an utterance to train the models. In the second case, we additionally consider fundamental frequency (F0) information along with energy, referred to as Energy+F0. Speech

frames with valid F0 value (*i.e.* voiced frames) are only considered for the initialization of speech model. On the other hand, for non-speech model, we consider only the speech frames having no valid F0 values. For F0 extraction, we adopt a cross-correlation method from the `Snack Sound Toolkit` [47].

Specific care is taken to ensure that the contrastive initialization schemes use the same number of selected speech frames. In specific, if the F0 detector has detected a sufficient amount of voiced frames, we use the voiced frames with the highest energy values. But when there are not enough detected voiced frames, we augment them with the highest energy frames taken from the remaining part of an utterance. For instance, if the target is to train the utterance-specific speech model using 10% of frames but only 6% of them are voiced, we choose the rest (4%) from the remaining frames in the recording corresponding to the highest energy values. Therefore, if no voiced frames were detected, the Energy+F0 initialization scheme collapses to the first one.

Our third strategy uses *oracle* initialization with the help of ground-truth labels. In this case we select a random subset of frames from each class. The purpose of our oracle initialization analysis is two-fold. Firstly, it simulates human-assisted SAD initialization; even if the applications concerned in this study involve *automatic* speaker verification experiments, there are applications, such as forensics, that involve at least partial human supervision. Secondly, it provides an experimental bound on the performance achievable by a perfectly initialized SAD (assuming the ground truth labels are correct).

Table 2: Effect of initialization method (Energy, Energy + F0) on the NIST OpenSAD set, (Miss % / FA %).

SAD used	Energy (10%)	Energy + F0 (10%)
GMM	34.99 / 24.27	27.22 / 16.06
SSGMM	**31.04** / **23.45**	**22.23** / **14.31**

Table 2 shows the miss and false alarm rates for different initializations on the OpenSAD data for the MFCC features. As is obvious, the use of F0 to create speech model helps in reducing both miss and false alarm rates. As the F0 detector itself is characterized by very small false alarm rates, our speech

Table 3: Effect of amount of training data(%) with estimated labels. SAD performance on the NIST OpenSAD set is shown as (Miss % / FA %).

SAD used	Energy + F0 (1%)	Energy + F0 (10%)	Energy + F0 (20%)
GMM	23.74 / 20.66	27.22 / 16.06	26.15 / 17.69
SSGMM	**20.44** / **14.79**	**22.23** / **14.31**	**24.06** / **16.03**

Table 4: Effect of amount of training data (%) with oracle labels. SAD performance on the NIST OpenSAD set is shown as (Miss % / FA %).

SAD used	Oracle (1%)	Oracle (10%)	Oracle (100%)
GMM	18.08 / 7.21	18.15 / 7.39	18.10 / 7.06
SSGMM	17.06 / 11.57	16.94 / 10.88	18.10 / 7.06

model becomes trained from data having less mis-labeled exemplars. Table 3 shows further the SAD performance dependency on the amount of data used for initialization (1 %, 10 % and 20 %). Interestingly, using less frames for initialization leads to lower error rates, especially the miss rate. In general, we expect a trade-off between the amount and quality of the initialization. Here, 1 % of speech or nonspeech data consists of about 6 seconds of data (the audio files are 10 minutes long).

Further, when the ground truth labels are used (Table 4), the performance is stable with different amounts initialization data. These results indicate that larger amounts of imperfectly labeled data degrades the speech and nonspeech models. The last column in Table 4 shows the bounds for the performance that can be achieved, as in corresponds to the case of training and evaluating the speech and nonspeech models on the same data. The results of GMM with and without semisupervision agree with the use of maximum amount of accurately labeled data. By comparing the oracle results of Table 4 to those obtained in Tables 2 and 3, we see that the best obtained miss rate (20.44 %) is not too far from the oracle (18.10 %) but there is a 2-fold gap in the best false alarm rate (14.31 %) in comparison to that of the oracle (7.06 %).

Figure 6 further shows DET curves corresponding to the different entries in Tables 2, 3 and 4. We make several interesting observations. First, the quality (accuracy) of the initialization has apparently a profound impact on SAD

performance. Second, SSGMM has lower error rate than GMM for imperfect initialization but it performs slightly worse when oracle labels are used. This is an expected result, explained as follows. In the case of oracle labels, both speech and non-speech models created by the GMM method are guaranteed to be "pure" since each model is trained using data only from the corresponding class. In contrast, by using unlabeled data, the SSGMM method is bound to be corrupted from wrongly-assigned labels to the unlabeled feature vectors. This negative effect was dominated by the positive effect of using more data to create more accurate models in the case of imperfect initialization. Our main conclusion, therefore, is that SSGMM is preferable when the auxiliary SAD used to get initial labels is not very accurate. When the initialization is close to perfect, however, the proposed SSGMM may not provide considerable performance gains over the conventional GMM.

6. Application to Speaker Verification: Experimental Set-up

The proposed speech activity detector is, at least in principle, application-independent. In this study, we focus on an automatic speaker verification application, including evaluation across varied clean and noisy conditions. We conduct extensive experiments in both text-dependent and text-independent scenarios on multiple speech corpora to confirm the consistency of results. In contrast to the text-independent scenario containing large amounts of telephony data, our text-dependent speaker verification samples are short smartphone recordings.

6.1. Text-Independent Speaker Verification Set-up

Dataset: For the text-independent ASV experiments, we select a subset of the male trials from **NIST SRE 2010**[11] consisting of telephone quality speech. In specific, we report our findings on the normal vocal effort telephone speech

[11] http://www.nist.gov/itl/iad/mig/upload/NIST_SRE10_evalplan-r6.pdf

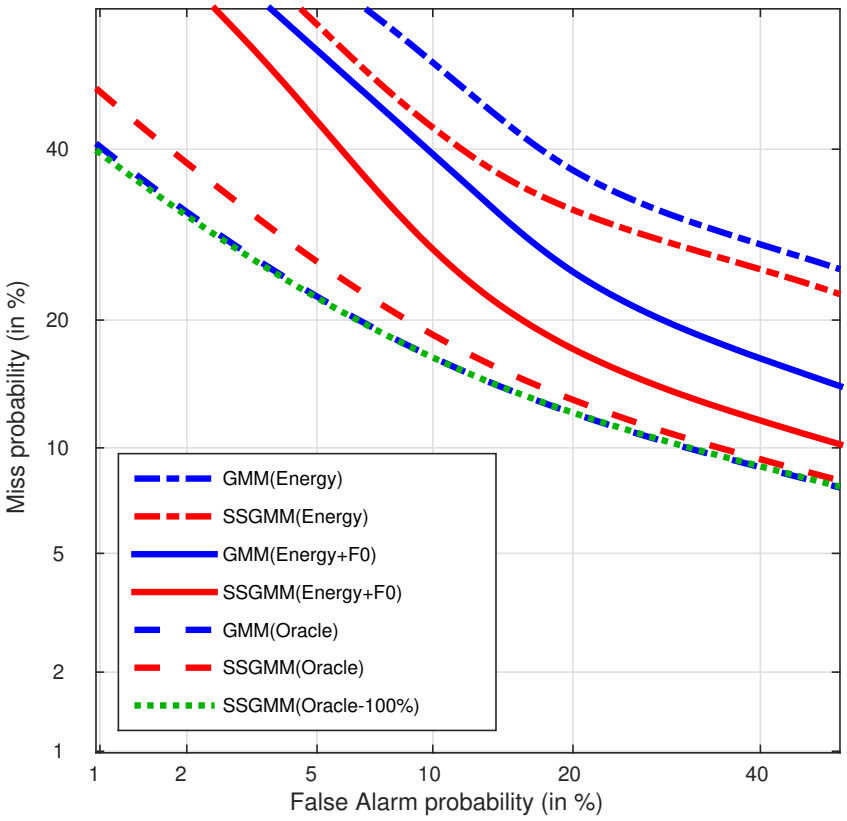


Figure 6: SAD performance on the NIST OpenSAD with different initialization strategies. Each point on the curves represents *average* miss rate across files for a given false alarm rate. The amount of initialization data is equal to 10% in all the cases, except those explicitly specified. In the case where 100% of oracle labels are used, SSGMM and GMM are by definition equivalent (this corresponds to $M = 0$ in equations (5) - (7)). Therefore, we have arbitrarily labeled the thick dotted green line as 'SSGMM' but it could have been equivalently labeled as 'GMM' (if both curves corresponding to Oracle-100% were shown, they would be completely overlapping).

condition (CC5) which has more male trials than the other telephone conditions of SRE10. Another reason to select the male subset is to limit the number of simulations and to focus more on the analysis for different SAD configurations. Additionally, experiments are also conducted with digitally added noise on speech signals. We choose three different levels of noise: 10 dB, 6 dB and 0 dB, and use similar noise adding procedure as in the standalone SAD evaluation.

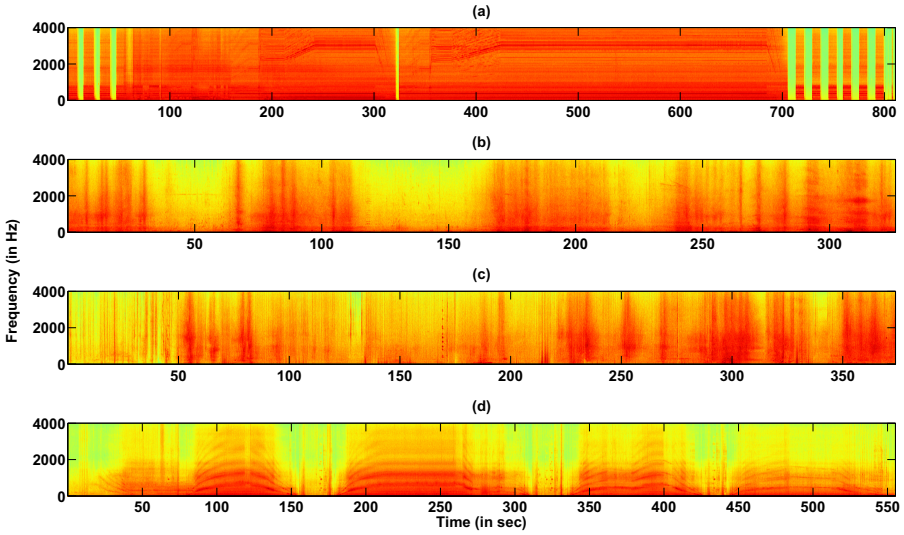


Figure 7: Spectrograms of noise signals used for corrupting the speech samples for the ASV experiment. As can be seen, the noises are highly non-stationary ones. For instance, the uppermost signal represents a washing machine noise and the lowest one train (underground) noise.

The noise samples for the ASV part contain both stationary and non-stationary noise samples as illustrated in Fig. 7.

Classifier back-end: Our ASV system back-end for the text-independent experiments uses i-vector [48] utterance descriptors with *probabilistic linear discriminant analysis* (PLDA) [49] scoring. In specific, We use the *simplified* PLDA model described in [50]. We train gender-dependent UBMs with 512 mixture components using the EM algorithm with speech data from NIST SRE 2004—2006, FISHER, and Switchboard corpora. Then, we train a total variability matrix (i-vector extractor) with 400 factors using five EM iterations from 19082 files from the same databases. Prior to PLDA training or scoring, the i-vectors are pre-processed using *linear discriminant analysis* (LDA) to reduce their dimension to 200, followed by radial Gaussianization [51]. For LDA and PLDA, we re-use the same utterances as for training the i-vector extractor. The dimensionality of the eigenvoice subspace in PLDA is set to 150 and 20 EM iterations are used for estimating the PLDA hyper-parameters.

6.2. Text-Dependent Speaker Verification Set-up

Dataset: For the text-dependent experiments, we choose RSR2015 [52] and recent quarter four (Q4) release of **RedDots** corpus[12]. The RSR2015 consists of microphone speech collected in clean and matched environmental condition whereas RedDots corpus is collected from speakers using different smartphones as well from variable environmental condition. The RedDots database is collected from speakers using different smart-phones and acoustic environments. Speech samples are collected using crowd-sourcing from native and native English speakers of diverse accents around the globe. We choose the first part (Part 1) male subset of both datasets for our experiments. In this subset, the speakers pronounce a set of *fixed pass-phrases* identical to all the speakers. This condition is also a popular choice for user authentication in commercially available voice biometrics[13]. The number of trials for three corpora are summarized in Table 5. We have used male sections only because they have more numbers of trials in most of the datasets used in our experiments.

Table 5: Summary of trials for the corpora used in speaker verification experiments.

Database Name	#target	#non-target
NIST SRE2010	353	13707
RSR2015	10244	573664
RedDots	3242	120086

Classifier back-end: Our text-dependent ASV system uses the classic Gaussian mixture model with universal background model (GMM-UBM) [53]. In our experiments with RedDots corpora, we have found it to outperform several other ASV techniques including i-vectors and hidden Markov models [54] (for similar findings, see [55]). We train a UBM with 512 Gaussians with diagonal covariances from 4380 male utterances of the TIMIT corpus. TIMIT is a good choice for UBM model compared to NIST SRE or Switchboard cor-

[12] https://sites.google.com/site/thereddotsproject/reddots-2015-quarter-4-release

[13] http://www.nuance.com/ucmct/groups/imaging/@web-enus/documents/collateral/nucc1021vocalpasswordv9proddes.pdf

pora as the evaluation speech files are recorded at 16 kHz similar to TIMIT. In [56, 54] results with TIMIT background were found comparable to other systems using utterances from other data such as RSR2015 or LibriSpeech used in [31, 55, 57]. The target speaker models are trained using *maximum-a-posteriori* (MAP) adaptation of the Gaussian means as detailed in [53], using adaptation relevance factor of 4, as optimized in [54].

6.3. Feature Extraction and Contrastive Speech Activity Detectors

Both the text-independent and -dependent ASV systems described above use the same MFCC features extracted from 20 ms Hamming-windowed frames every 10 ms. We use 20 filters in mel filterbank. We retain 19 base coefficients after DCT, discarding the DC-coefficient. The speech features are then processed by RASTA filtering [58] to suppress the effects of linear, slowly-varying channel effects. Then 57-dimensional MFCCs are formed after augmenting the dynamic coefficients. Here delta and double-delta features are computed using differentiator method across three adjacent coefficients. Finally, *cepstral mean and variance normalization* (CMVN) is carried out after discarding the non-speech frames with a speech activity detector.

As the focal point of this study is the speech activity detector, our contrastive results presented in Section 7 consists always of otherwise equivalent ASV systems except that the enrollment and the test samples are processed with different SADs. However, the system off-line components, namely, UBM, T-matrix, LDA and PLDA are pre-trained with the same fixed SAD; as these datasets are relatively clean and trained from very large number of speech frames, we use bi-Gaussian energy SAD for this purpose (see below for the details). Even if further performance gain might be expected by using "matched" or further optimized SADs for the off-line components, there are two reasons we have decided to *not* do this. Firstly, in *forensic automatic speaker recognition* (FASR) systems, and in certain commercial applications, the application user has typically access only to enrollment and test samples while the internal system parameters are fixed in advance and optimized by the system vendor. The second

reason is computational: experimentation overhead is lower with fixed SAD for the off-line components, allowing us to study more SAD variants. The same experimentation strategy was adopted in [18]. We summarize our experimental design as follows.

SAD for the enrollment and test files: one of the contrastive SADs listed below.

SAD for the off-line components: UBM, T-matrix, LDA and PLDA are trained from features extracted with fixed, bi-Gaussian energy SAD. Hence, the system back-end remains fixed.

Acoustic features: same in all the cases (19 MFCCs with deltas and double deltas), CMVN after SAD processing.

We consider the following contrastive SADs for processing the enrollment and test utterances:

No SAD: features are processed without any speech activity detection. This serves as a reference point, especially for the text-dependent ASV set-up containing short amount of speech, with a potential quality-amount trade-off of selected speech frames for ASV scoring.

Clean labels: Clean SAD labels are obtained from the original uncorrupted speech recordings using bi-Gaussian energy SAD; thus, performance of the clean labels and bi-Gaussian SAD agree on the original data but will in general differ for noise-added data.

Bi-Gaussian: In bi-Gaussian based speech activity detection method, the log-energies of speech frames are fitted with GMM of two components [41]. The mode with a higher mean is assumed to correspond to speech whereas the other mode with the lower mean is assumed to corresponds to non-speech [59]. In a previous comparison of SADs for ASV task, bi-Gaussian

SAD outperformed other methods in noisy condition [26]. We use our own implementation as made available in [14].

Sohn: Sohn's SAD [7] uses a robust decision rule derived from the generalized likelihood ratio test considering geometric mean likelihood ratios over all the frequency bins. It uses the noise statistics using a noise estimation approach. We use the implementation provided in voicebox toolbox[15].

rSAD: rSAD is another unsupervised SAD [60]. Here, the speech signal is first filtered using a high-pass filter and then high-energy segments are detected using *a posteriori* SNR weighted energy difference. Within the high energy segments, frames without a valid pitch value are marked as noise. Then, a denoised signal is generated by using a modified minimum statistics based noise estimation method and setting the high-energy noise segments to zero. Then chunk of speech frames are also verified if their signal power is considerably greater than the corresponding noise power. Finally, *a posteriori* SNR weighted energy difference measure is applied to the denoised signal, and the frames containing pitch are treated speech. The rSAD produced promising performance in recent NIST OpenSad challenge [37]. We use the implementation by the original authors[16].

GMM: Proposed GMM-based SAD described in Section 3 without semi-supervised training.

SSGMM: Proposed GMM-based SAD described in Section 3 with semi-supervised training.

6.4. Performance measure

We assess speaker verification performance in terms of *equal error rate* (EER) which corresponds to the operating point (decision threshold) at which the miss

[14] http://cs.joensuu.fi/~sahid/codes/bgVAD1.0.zip
[15] http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html
[16] http://kom.aau.dk/~zt/online/rVAD/index.htm

and false alarm rates of the ASV system coincide. In practice, we compute the EER values using BOSARIS toolkit with ROC convex hull (ROCCH) method[17].

7. Application to Speaker Verification: Results

7.1. Text-Independent Speaker Verification

In our first set of experiments, we evaluate different SADs in text-independent task on the original NIST SRE 2010 data without added noise (matched condition) and for the mismatched condition where the test data is distorted with digitally added noises in 3 SNR levels: 10 dB, 6 dB and 0 dB. The results are shown in Table 6. For the GMM and proposed semi-supervised GMM, the results are also shown for the best configurations found in Section 5 with MFCC and PNCC features. The number of Gaussian is selected as 2 and 8 with full (non-shared) covariances. In all cases, we use 50 speech frames (0.5 seconds as our frame rate is 100 fps) to initialize the speech and non-speech models. This amount was chosen according to the initialization scheme based on F0 detection and signal energy computation.

The results with "no SAD" denotes the performance without using any SAD whereas the results with clean labels indicate the results with a perfect SAD. From the results with no SAD and clean labels, we infer that the ASV performance can be considerably improved by using ground-truth regarding speech/non-speech class. We then compare the performances with baseline bi-Gaussian SADs and other two standard SADs: Sohn and rSAD. The results indicate that use of SAD can considerably improve ASV performance. The performances of Sohn and rSAD are relatively better than baseline method in noisy conditions.

Comparing GMM and SSGMM, the latter yields lower EERs in most cases. This can be explained noting that SSGMM uses all the available frames to enhance the models while GMM relies only on the small amount of initializa-

[17] https://sites.google.com/site/bosaristoolkit/

Table 6: ASV performance in terms of EER (%) on NIST SRE 2010 in clean and noisy condition using different SADs. For the proposed GMM and SSGMM methods, we indicate the features (MFCC, PNCC) and the number of Gaussian components (K) used for modeling speech and nonspeech classes.

SAD Method	Matched	10 dB	6 dB	0 dB
No SAD	12.53	18.62	21.68	28.32
Clean labels	4.25	7.07	9.56	14.91
Bi-Gaussian	**4.25**	11.86	14.23	23.49
Sohn [7]	6.16	**7.50**	**9.42**	**13.20**
rSAD [60]	4.70	7.81	9.73	14.27
GMM (MFCC, $K = 8$)	11.10	13.10	14.37	18.69
GMM (MFCC, $K = 2$)	5.59	9.86	12.25	16.80
GMM (PNCC, $K = 8$)	7.63	12.31	13.18	17.22
GMM (PNCC, $K = 2$)	5.38	7.93	10.00	**13.87**
SSGMM (MFCC, $K = 8$)	**4.36**	7.58	10.13	15.37
SSGMM (MFCC, $K = 2$)	4.41	**6.81**	**9.43**	15.23
SSGMM (PNCC, $K = 8$)	6.01	8.12	10.21	15.66
SSGMM (PNCC, $K = 2$)	6.12	7.48	9.17	14.18

tion data which might be not enough for fitting accurate models. Comparing the models with $K = 8$ or $K = 2$ Gaussians, the latter yields slightly better performance. The proposed SAD techniques systematically outperform the energy-based bi-Gaussian SAD for noisy conditions. On the other hand, the performance of proposed SSGMM with MFCC features and two Gaussians is better other than existing SADS in higher SNRs (matched and 10 dB) and is slightly poor for lower SNRs (6 dB and 0 dB). Interestingly, the clean SAD is not showing the best performance for noisy condition. This might be due to the fact that a SAD algorithm also discards unreliable speech frames with very low segmental SNR but 'clean labels' includes those noisy frames and they introduce errors in ASV scoring process.

Next we study the effect of initialization length on the accuracy of the ASV system, as illustrated in Fig. 8. We observe that while the accuracy of GMM without semi-supervised training improves with longer data, the SSGMM is relatively insensitive to the amount of initialization data. This is expected and

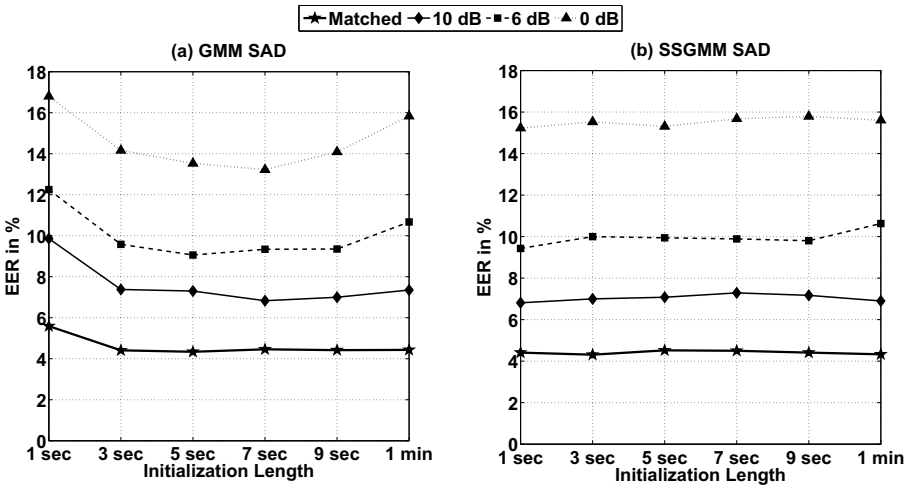


Figure 8: Effect of initialization data amount for GMM SAD and SSGMM SAD using MFCC and PNCC features with two Gaussians ($K = 2$) using full covariance. Performances in terms of % of EER are shown for matched and noisy condition for amount of initialization as 1s, 3s, 5s, 7s, 9s and 1m.

favorable as it enables operating the SAD using different amounts of supervision.

7.2. Text-Dependent Speaker Verification: Results

Table 7 shows the results for the RedDots corpus. As before, we used 0.5 seconds of audio data to initialize the GMM models. Since the speech segments are short in duration, we have used shared covariance here as the total amount of speech is not adequate to estimate the covariances. We can see no considerable benefit of using SSGMM in terms of ASV system performance. This observation is supported by the intuition that the semi-supervised training can be helpful if the overall amount of data is relatively large. Otherwise, neither GMM, nor SSGMM method is able to fit accurate speech and non-speech models. We can see that the GMM-based systems has lower accuracy, which also can be explained by the short file durations in these datasets.

We have also observed the similar trend for RSR2015 in clean and noisy conditions. The results are shown in Table 8. Here also, the proposed SADs as well as Sohn and rSAD method give higher EER than simple energy-based SAD.

Table 7: ASV performance in terms of EER (%) on RedDots using no SAD and different proposed and existing SAD techniques.

SAD Method	EER (%)
No SAD	4.99
Clean labels	3.01
Bi-Gaussian	3.01
Sohn [7]	2.81
rSAD [60]	3.37
GMM (MFCC, full-2)	3.81
GMM (PNCC, full-2)	3.64
SSGMM (MFCC, full-2)	3.87
SSGMM (PNCC, full-2)	4.04

Using no SAD at all outperforms all the compared SADs except energy-based Bi-Gaussian SAD for the clean condition. These results confirm the limitations of the proposed SADs for short text-dependent ASV tasks.

Table 8: ASV performance in terms of EER (%) on RSR2015 in clean and noisy condition using no SAD and different proposed and existing SAD techniques.

SAD Method	Matched	10 dB	6 dB	0 dB
No SAD	2.18	7.71	12.21	22.50
Clean labels	2.08	5.00	7.85	16.64
Bi-Gaussian	2.08	5.07	7.91	17.30
Sohn [7]	2.35	5.80	8.89	17.91
rSAD [60]	3.09	6.84	10.47	19.94
GMM (MFCC, 2)	3.21	6.71	9.89	18.80
GMM (PNCC, 2)	**2.59**	5.80	8.72	17.35
SSGMM (MFCC, 2)	2.88	6.35	9.34	18.24
SSGMM (PNCC, 2)	2.62	**5.66**	**8.45**	**16.79**

7.3. Shared features for SAD and ASV

So far in this study, we have used separate features SAD and ASV tasks. In the next experiment, we use the *same* acoustic front-end for both SAD and ASV. To model speech and non-speech classes, we use the same 19-dimensional

Table 9: ASV performance in terms of EER (%) on NIST SRE 2010 in clean and noisy condition with shared feature. MFCC features used for ASV task are adopted here for SAD.

SAD Method	Matched	10 dB	6 dB	0 dB
GMM (8)	7.49	11.65	13.50	17.89
GMM (2)	5.45	8.07	10.82	14.89
SSGMM (8)	4.81	8.01	10.43	15.50
SSGMM (2)	**4.66**	**7.04**	**9.10**	**13.67**

base MFCCs extracted with 20 filters as used in ASV task (as discussed in Section 6.3). One obvious advantage of this feature sharing scheme is that it does not require additional computational overhead for SAD feature extraction. As shown in Table 9, the ASV results with this approach indicate that SSGMM outperforms GMM. We also notice that for GMM-SAD, the ASV performance is better than the performance obtained by MFCC feature with different configuration as reported in Table 6.

8. Impact of SAD Errors to Speaker Verification Accuracy

Throughout our experiments, we have compared different SADs using their default settings, *i.e.*, the SADs were not specially calibrated for ASV tasks. Nonetheless, as a binary classifier involving the use of a detection threshold, any SAD in practice will have to trade-off between speech misses (speech frames declared as non-speech by the SAD) and false alarms (non-speech frames declared as speech by the SAD). One should expect the relative importance of these two types errors to depend on the application, type and level of background noise, duration of speech utterances and other factors. As an example, the evaluation metric for the OpenSAD challenge expressed in Eq. 9, with parameter $\alpha = 0.75$ expresses a belief that speech misses are three times more costly than falsely detected non-speech frames. Hence, the threshold for the SAD should be optimized to detect a large number of speech frames. But in ASV, especially under noisy conditions, we might have a different preference so as to avoid including unusable frames with high amount of noise into speaker enrollment or scoring.

With this background, our final analysis in this Section addresses the question how one should weight the false alarms and misses in the ASV context. By knowing — even approximately — what the SAD segmentation metric should be in order to maximize ASV accuracy, the acquired knowledge would be useful for optimizing other SADs outside the specific databases or classifiers used in this study.

We begin by visualizing the distribution of the test file SAD errors represented by the corresponding $\mathrm{P_{miss}}$ and $\mathrm{P_{fa}}$ in the SRE 2010 data. Figure 9 illustrates this for one SNR level (10 dB). To obtain this graph, we varied the decision threshold of the SSGMM SAD such that, for each recording, a fixed fraction of frames in the range 5%-100% was labeled as speech. The step was equal to 5%, so that each SRE 2010 test file corresponds to a total of 20 points in this plot. To obtain the SAD false alarm and miss rates in each case, we use the segmentation produced by the bi-Gaussian SAD from the original uncorrupted NIST file as a reference. The resulting scatter plot consists of the union of 20 copies of the test set corresponding to different threshold levels and the three SNR levels (10 dB, 6 dB, 0 dB). We then computed the average miss and false alarm rates across the files within each of these 20 subsets, shown as circles in Fig. 9.

We then measured the EER *of the ASV system* for each case, *i.e.*, using the same trial list but different SAD labels. Given the small number of trials, we further estimate 95% confidence intervals of the EER following the methodology of [46]. In specific, we compute parametric confidence intervals as $\mathrm{EER} \pm \sigma \cdot Z_{\alpha/2}$, where $Z_{\alpha/2} = 1.96$, $\sigma = 0.5\sqrt{\mathrm{EER}(1-\mathrm{EER})(n_+ + n_-)/(n_+ n_-)}$ and n_+, n_- are the number of target and non-target trials, respectively.

The results, shown in Figure 10, illustrates the dependency of the ASV EER on the logarithm of the miss to false alarm rates ratio of the SAD for the three different SNRs. The best performance corresponds to the point around $\mathrm{P_{miss}}/\mathrm{P_{fa}} \approx 5$ (since $\log(5) \approx 1.6$ – the point corresponding to the lowest EER) and is relatively stable to the noise level. Thus, for our data and SADs, we conclude that higher miss rate is less critical than false alarm rate, hence false

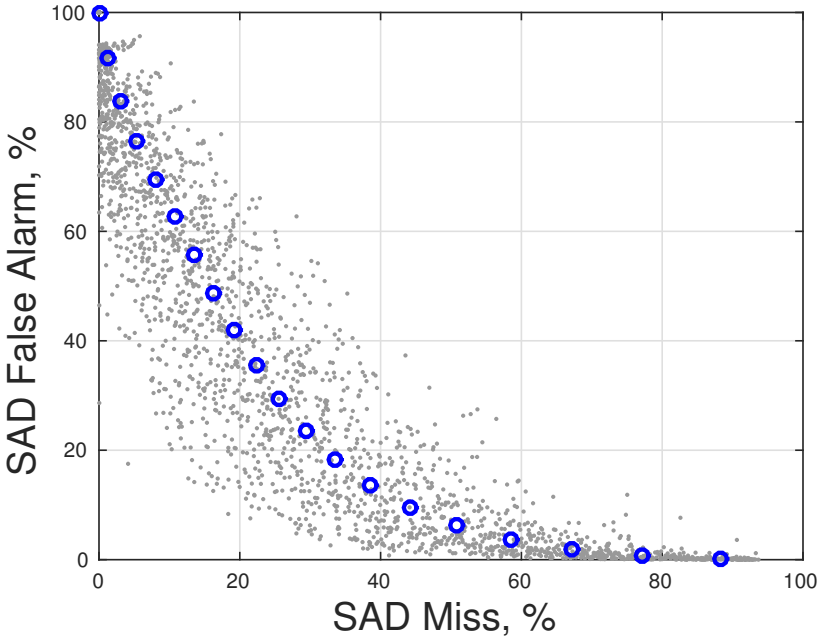


Figure 9: Distribution of the test file SAD errors of the GMM SAD. Circles correspond to the average errors computed for each copy of the test set. See text for the details.

alarms should be penalized roughly 4-5 times higher to get the best performance in the application to speaker verification. It coincides with intuition that "noise-poisoned" speaker model can be worse than the model estimated from smaller, but more pure speech data. Naturally, these observations should be re-examined using other SADs and datasets.

Interestingly, the proposed SSGMM method yields wider region of close-to-the-optimal ASV error rates. One possible explanation comes from comparing the objective functions for training GMM and SSGMM. SSGMM is trained by maximizing (3) while only the first term of (3) is used as the training objective for GMM. The second term in equation (3) can be seen as a regularizer which enforces a more smooth decision boundary, therefore, improves generalization. This might also explain why SSGMM was found to be less sensitive to the amount of initialization data, as we have seen in Figure 8. Hence, moving the decision boundary of SSGMM, by changing the decision threshold around the

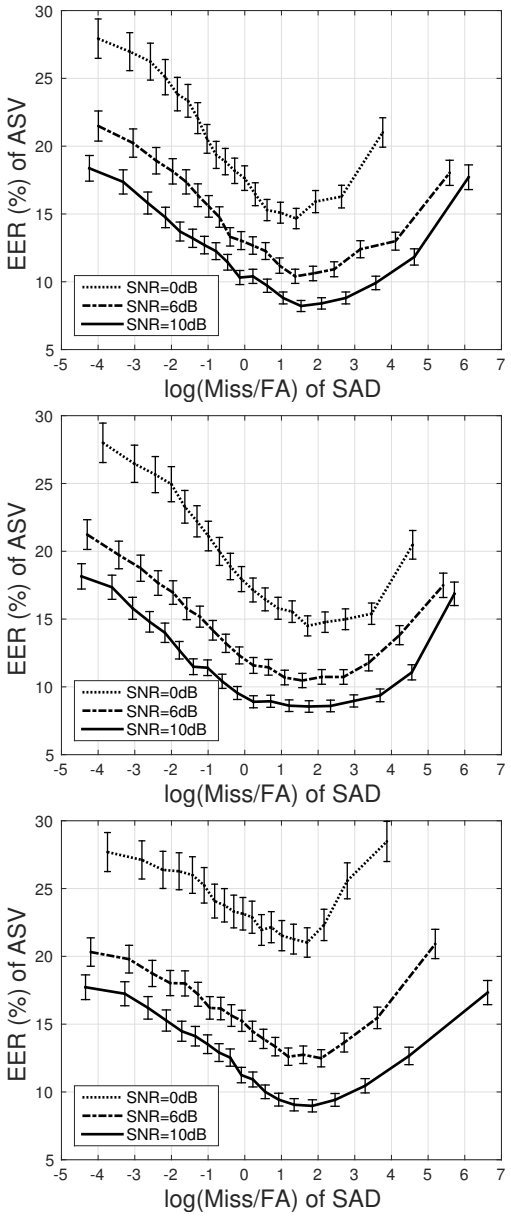


Figure 10: Estimated EER as a function of the logarithm of the miss to FA rates ratio for different SADs: GMM (top), SSGMM (center), BiGaussian (bottom). Horizontal axis represents the trade-off between two types of SAD errors which can tuned to achieve the best ASV performance in terms of EER. We can see that the optimal trade-off is relatively stable with respect to noise levels and almost the same for all three SADs.

region of the optimal values, has less impact than in the case of GMM.

9. Conclusion

We have introduced a simple and general-purpose Gaussian mixture model (GMM) based SAD trained from scratch for every audio recording. To train the speech and non-speech GMMs, a small fraction of labeled data from the specific recording is required, but otherwise the SAD segmentation process is completely automatic. Depending on the application scheme, the initial labels could be produced by a human annotator or by another simpler SAD. We have extensively evaluated the proposed SAD using three different datasets varying in their data qualities.

First, in order to find the optimal configuration in terms of standalone SAD performance, we evaluated our method on two datasets with very different noise characteristics. The main design parameters of the method include the number of Gaussians, the type of the covariance matrices, and the initialization method. The number of Gaussians was found to have little impact but the choice of covariance matrix structure has some impact. We found the SAD with full covariance matrices to outperform the diagonal and spherical covariances, and further, covariance sharing within each class (speech and non-speech) was helpful.

Experiments with initial labeling revealed that accurate initialization is very important. A combination of energy and F0 outperforms simpler energy-based initialization (used in [18]) on the extremely noisy NIST OpenSAD data, as one might expect. Further, our results suggest that larger amounts of imperfectly labeled data degrades the models, leading to drop in accuracy. In turn, with oracle initialization (using known reliable labels), SAD accuracy is stable to changing the amount of data used to train models. These results are expected. Comparing the baseline GMM and the proposed SSGMM models, the latter is recommend especially when there is no guarantee on the correctness of the initial labels, while the former can be more accurate with reliable initial labels.

We further compared four different features widely used in different recognition tasks: MFCC, PLP, FDLP, and PNCC. The first three features had comparable error rates while the use of PNCC features yielded higher miss and much lower false alarm rates. While the choice of best SAD features is expected to depend on a specific application and data, our experiments demonstrate that all the features produced reasonable results. Therefore, it is straightforward to integrate different spectral features into the proposed SAD modeling scheme without requiring major changes in the back-end parameter configuration.

Besides standalone evaluation of SADs, we assessed the performance of ASV systems with different SADs integrated to them. These experiments included two different datasets involving both text-independent and text-dependent scenarios. Concerning the text-independent case, the proposed GMM-based SADs yielded comparable performance to the systems based on several other unsupervised SADs. In specific, they outperform the most commonly used, energy-based (in our case, with bi-Gaussian threshold selection) by a wide margin under the noisiest conditions as indicated by the results in Table 6. The performance is comparable and sometimes better to the other two unsupervised SADs, Sohn and rSAD. Concerning the text-dependent results with relatively short duration data, presented in Table 7, our results are, however, negative: the simpler baseline methods outperformed the proposed methods.

Several conclusions can be drawn from these ASV experiments: (1) PNCC features seem preferable for noisier conditions while MFCCs can be better for relatively cleaner data; (2) the proposed SAD is best suited for long and noisy data conditions, such as the NIST datasets, while the simpler DSP-based unsupervised SADs seem sufficient for shorter data conditions; and (3) false alarms of the SADs should be penalized higher than misses, *i.e.*, decision threshold should be adjusted in a way to make SAD more selective in order to reach lower EER.

Overall, comparing the two alternative ways to train the GMMs for ASV, with and without semi-supervision, the latter was found helpful in the case of long recordings, but none of them was found a good choice for our text-

dependent scenario involving short utterances in RedDots and RSR2015. This observation is consistent with the core idea of the GMM training process: it relies on the availability of relatively large amounts of data for fitting accurate models — on the order of a few minutes, rather than only a few seconds.

We observed that using no SAD at all in text-independent task leads to considerable ASV performance degradation, as one could expect. In contrast, in the experiments on text-dependent ASV, the relative degradation between the best SAD and no SAD was much smaller. This might be explained by the short durations in the text-dependent case because such limited amount of data ($\sim$ 10 sec for enrolment and $\sim$ 3 sec for test) are not sufficient for training target speaker models and decision making and the use of SAD further reduces the amount of data.

Finally, utilizing the NIST SRE 2010 data, we took a detailed look into how the SAD miss and false alarm errors impact ASV performance. Our novel analysis revealed that best ASV performance corresponds to an SAD segmentation cost function where, approximately, false alarms should penalized 4 to 5 times over the misses. What we find most interesting is that this choice, reflected by the minima region in Fig. 10, *does not depend on neither the SAD method nor the SNR*. While we do not claim generality of such observation beyond the specific methods and data presented here, we find the result encouraging for future studies on SAD optimization for ASV (or other recognition) tasks. Further, our analysis reveals an interesting property of the proposed semi-supervised SAD: it yields comparatively more stable ASV error rates over a wider range of SAD decision points especially under low noise condition in comparison to the other two SAD methods compared. This leads us to conclude that the SAD calibration might be relatively easy for the proposed method, while for instance the success of the energy-based SAD can be critically dependent on hitting just the 'correct' SAD operating point.

Our SAD naturally allows initialization with a portion of oracle labels provided by human or more accurate but computationally expensive SAD. These suggest several possible directions to improve the method, specifically, combin-

ing different features, using alternative models for speech/nonspeech and using other SADs for initialization.

10. Acknowledgements

This paper reflects some results from the OCTAVE Project (#647850), funded by the Research European Agency (REA) of the European Commission, in its framework programme Horizon 2020. The views expressed in this paper are those of the authors and do not engage any official position of the European Commission. Part of the study was also funded by Academy of Finland (projects 283256 and 288558) and the Government of the Russian Federation, Grant 074-U01.

[1] J. Ramirez, J. M. Gorriz, J. C. Segura, Voice Activity Detection. Fundamentals and Speech Recognition System Robustness, InTech, 2007.

[2] A. Kindoz, A. M. Kondoz, Digital Speech; Coding for Low Bit Rate Communication Systems, 1st Edition, John Wiley & Sons, Inc., New York, NY, USA, 1994.

[3] M.-W. Mak, H.-B. Yu, A study of voice activity detection techniques for NIST speaker recognition evaluations, Computer Speech & Language 28 (1) (2014) 295 – 313.

[4] A. Benyassine, E. Schlomot, H. Su, ITU-T recommendation G729 Annex B: A silence compression scheme for use with G729 optimized for v.70 digital simultaneous voice and data applications, IEEE Communications Magazine 35 (1997) 64–73.

[5] R. Tucker, Voice activity detection using a periodicity measure, IEE Proc.-I (Communications, Speech & Vision) 139 (4) (1992) 377–380.

[6] J. Ramirez, J. Segura, C. Benitez, A. D. L. Torre, A. Rubio, Efficient voice activity detection algorithms using long-term speech information, Speech Communication 42 (2004) 3–4.

[7] J. Sohn, N. Kim, W. Sung, A statistical model-based voice activity detection, IEEE Signal Processing Letters 6 (1999) 1–3.

[8] J. Shin, J.-H. Chang, N. Kim, Voice activity detection based on statistical models and machine learning approaches, Comput. Speech Lang. 24 (3) (2010) 515–530.

[9] J. Shin, J.-H. Chang, N. Kim, Voice activity detection based on a family of parametric distributions, Pattern Recogn. Lett. 28 (11) (2007) 1295–1299.

[10] R. Ng, M. Nicolao, O. Saz, M. Hasan, B. Chettri, M. Doulaty, T. Lee, T. Hain, The Sheffield language recognition system in NIST LRE 2015, in: Proc. Odyssey 2016: The Speaker and Language Recognition Workshop, 2016.

[11] O. Plchot, P. Matejka, R. Fer, O. Glembek, O. Novotn, J. Pesan, K. Vesel, L. Ondel, M. Karafiat, F. Grezl, S. Kesiraju, L. Burget, N. Brummer, A. Swart, S. Cumani, S. H. Mallidi, R. Li, BAT system description for NIST LRE 2015, in: Proc. Odyssey 2016: The Speaker and Language Recognition Workshop, 2016.

[12] J. Wu, X. Zhang, Maximum margin clustering based statistical VAD with multiple observation compound feature, IEEE Signal Process. Lett. 18 (5) (2011) 283–286.

[13] X. Zhang, J. Wu, Deep belief networks based voice activity detection, IEEE Transactions on Audio, Speech & Language Processing 21 (4) (2013) 697–710.

[14] S. Thomas, G. Saon, M. Segbroeck, S. Narayanan, Improvements to the IBM speech activity detection system for the DARPA RATS program, in: Proc. ICASSP, South Brisbane, Australia, 2015, pp. 4500–4504.

[15] X. Zhang, J. Wu, Transfer learning for voice activity detection: A denoising deep neural network perspective, CoRR abs/1303.2104.

[16] F. Germain, D. Sun, G. Mysore, Speaker and noise independent voice activity detection, in: Proc. Interspeech, Lyon, France, 2013, pp. 732–736.

[17] M. Huijbregts, C. Wooters, R. Ordelman, Filtering the unknown: Speech activity detection in heterogeneous video collections, in: Proc. Interspeech, Antwerp, Belgium, 2007, pp. 2925–2928.

[18] T. Kinnunen, P. Rajan, A practical, self-adaptive voice activity detector for speaker verification with noisy telephone and microphone data, in: Proc. ICASSP, Vancouver, Canada, 2013, pp. 7229–7233.

[19] X. Zhu, Semi-supervised learning literature survey, Tech. Rep. 1530, http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf (2005).

[20] K. Nigam, A. Mccallum, T. Mitchell, Semi-supervised text classification using EM, in: Semi-Supervised Learning, MIT Press, 2006, pp. 33–56.

[21] A. Martinez-Uso, F. Pla, J. Sotoca, A semi-supervised Gaussian mixture model for image segmentation, in: Proc. of ICPR, Washington DC, USA, 2010, pp. 2941–2944.

[22] P. Moreno, S. Agarwal, An experimental study of EM-based algorithms for semi-supervised learning in audio classification, in: Proc. of the ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data, 2003.

[23] A. Diment, T. Heittola, T. Virtanen, Semi-supervised learning for musical instrument recognition, in: Proc. EUSIPCO, Marrakech, Morocco, 2013.

[24] 2010 NIST Speaker Recognition Evaluation, http://www.itl.nist.gov/iad/mig//tests/sre/2010/.

[25] NIST Open Speech-Activity-Detection Evaluation (OpenSAD), http://www.nist.gov/itl/iad/mig/opensad_15.cfm.

[26] M. Sahidullah, G. Saha, Comparison of Speech Activity Detection Techniques for Speaker Recognition, ArXiv e-printsarXiv:1210.0297.

[27] T. Kinnunen, H. Li, An overview of text-independent speaker recognition: from features to supervectors, Speech Communication 52 (1) (2010) 12–40.

[28] L. Ferrer, M. McLaren, N. Scheffer, Y. Lei, M. Graciarena, V. Mitra, A noise-robust system for NIST 2012 speaker recognition evaluation, in: Proc. Interspeech, 2013.

[29] J. Villalba, E. Lleida, A. Ortega, A. Miguel, The I3A speaker recognition system for NIST SRE12: Post-evaluation analysis, in: Proc. Interspeech, 2013, pp. 3689–3693.

[30] O. Plchot, S. Matsoukas, P. Matejka, N. Dehak, J. Ma, S. Cumani, O. Glembek, H. Hermansky, S. H. Mallidi, N. Mesgarani, R. Schwartz, M. Soufifar, Z. H. Tan, S. Thomas, B. Zhang, X. Zhou, Developing a speaker identification system for the DARPA RATS project, in: Proc. ICASSP, 2013, pp. 6768–6772.

[31] M. Alam, P. Kenny, P. Ouellet, T. Stafylakis, P. Dumouchel, Supervised/unsupervised voice activity detectors for text-dependent speaker recognition on the RSR2015 corpus, in: Proc. the Odyssey speaker and language recognition workshop (Odyssey2014), 2014, pp. 123–130.

[32] K. Nigam, A. K. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, Machine Learning 39 (2-3) (2000) 103–134.

[33] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the royal statistical society. Series B (methodological) 39 (1977) 1–38.

[34] J.-T. Huang, M. Hasegawa-Johnson, On semi-supervised learning of Gaussian mixture models for phonetic classification, in: Proc. NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing, SemiSupLearn '09, 2009, pp. 75–83.

[35] C. Bishop, Pattern Recognition and Machine Learning, Springer Science+Business Media, LLC, New York, 2006.

[36] K. Walker, S. Strassel, The RATS radio traffic collection system, in: Proc. of Odyssey 2012: The Speaker and Language Recognition Workshop, 2012.

[37] T. Kinnunen, A. Sholokhov, E. Khoury, D. Thomsen, M. Sahidullah, Z.-H. Tan, HAPPY team entry to NIST OpenSAD challenge: A fusion of short-term unsupervised and segment i-vector based speech activity detectors, in: Proc. Interspeech, 2016.

[38] M. Graciarena, A. Alwan, D. Ellis, H. Franco, L. Ferrer, J. Hansen, A. Janin, B. Lee, Y. Lei, V. Mitra, N. Morgan, S. Sadjadi, T. Tsai, N. Scheffer, L. Tan, B. Williams, All for one: feature combination for highly channel-degraded speech activity detection, in: Proc. Interspeech, 2013, pp. 709–713.

[39] G. Saon, S. Thomas, H. Soltau, S. Ganapathy, B. Kingsbury, The IBM speech activity detection system for the DARPA RATS program., in: Proc. Interspeech, 2013, pp. 3497–3501.

[40] M. Segbroeck, A. Tsiartas, S. Narayanan, A robust frontend for VAD: exploiting contextual, discriminative and spectral cues of human voice, in: Proc. Interspeech, 2013.

[41] D. Ying, Y. Yan, J. Dang, F. K. Soong, Voice activity detection based on an unsupervised learning framework, IEEE Trans. Audio, Speech and Lang. Proc. 19 (8) (2011) 2624–2633.

[42] C. Kim, R. Stern, Power-normalized cepstral coefficients (PNCC) for robust speech recognition, in: Proc. ICASSP, Kyoto, Japan, 2012, pp. 4101–4104.

[43] E. Ambikairajah, J. Kua, V. Sethu, H. Li, PNCC-ivector-SRC based speaker verification, in: Proc. APSIPA, 2012, pp. 1–7.

[44] D. Ellis, PLP and RASTA (and MFCC, and inversion) in MATLAB, http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/, accessed: 2016-03-22.

[45] M. Athineos, D. Ellis, Autoregressive modeling of temporal envelopes, IEEE Transactions on Signal Processing 15 (11) (2007) 5237–5245.

[46] S. Bengio, J. Mariéthoz, A statistical significance test for person authentication, in: Proc. Odyssey 2004: The Speaker and Language Recognition Workshop, 2004.

[47] The Snack sound toolkit, http://www.speech.kth.se/snack/, accessed: 2016-03-22.

[48] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, P. Ouellet, Front-end factor analysis for speaker verification, Audio, Speech, and Language Processing, IEEE Transactions on 19 (4) (2011) 788–798.

[49] P. Li, Y. Fu, U. Mohammed, J. H. Elder, S. Prince, Probabilistic models for inference about identity, IEEE Trans. on Pattern Analysis and Machine Intelligence 34 (1) (2012) 144–157.

[50] P. Kenny, Bayesian speaker verification with heavy-tailed priors, in: Proc. Odyssey 2010: The Speaker and Language Recognition Workshop, 2010, p. 14.

[51] D. Garcia-Romero, C. Espy-Wilson, Analysis of i-vector length normalization in speaker recognition systems, in: Proc. Interspeech, 2011.

[52] L. Larcher, K. Lee, B. Ma, H. Li, Text-dependent speaker verification: Classifiers, databases and RSR2015, Speech Communication 60 (2014) 56–77.

[53] D. Reynolds, T. Quatieri, R. Dunn, Speaker verification using adapted Gaussian mixture models, Digital Signal Processing 10 (1) (2000) 19–41.

[54] H. Delgado, M. Todisco, M. Sahidullah, A. Sarkar, N. Evans, T. Kinnunen, Z.-H. Tan, Further optimisations of constant Q cepstral processing for integrated utterance verification and text-dependent speaker verification, in: Proc. IEEE Workshop on Spoken Language Technology (SLT) 2016, 2016.

[55] H. Zeinali, H. Sameti, L. Burget, J. Cernocky, N. Maghsoodi, P. Matejka, i-vector/HMM based text-dependent speaker verification system for RedDots challenge, in: Proc. Interspeech, 2016.

[56] M. Sahidullah., T. Kinnunen, Local spectral variability features for speaker verification, Digital Signal Processing 50 (2016) 1 – 11.

[57] M. Alam, P. Kenny, V. Gupta, Tandem features for text-dependent speaker verification on the RedDots corpus, in: Proc. Interspeech, 2016.

[58] H. Hermansky, N. Morgan, RASTA processing of speech, IEEE Trans. on Speech and Audio Processing 2 (4) (1994) 578–589.

[59] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, D. Reynolds, A tutorial on text-independent speaker verification, EURASIP Journal on Applied Signal Processing 2004 (4) (2004) 430–451.

[60] Z.-H. Tan, B. Lindberg, Low-complexity variable frame rate analysis for speech recognition and voice activity detection, IEEE Journal of Selected Topics in Signal Processing 4 (5) (2010) 798–807.