Available online at www.sciencedirect.com

ScienceDirect

Computer Speech & Language 58 (2019) 231–246

www.elsevier.com/locate/csl

ALTIS: A new algorithm for adaptive long-term SNR estimation in multi-talker babble

Roozbeh Soleymani[a,b,*], Ivan W. Selesnick[a], David M. Landsberger[b]

[a] *Department of Electrical and Computer Engineering, New York University Tandon School of Engineering, 2 Metrotech Center, Brooklyn, NY 11201, USA*

[b] *Department of Otolaryngology, New York University School of Medicine, 550 1st Avenue, STE NBV 5E5, New York, NY 10016, USA*

Received 30 August 2018; received in revised form 7 April 2019; accepted 3 May 2019
Available online 10 May 2019

Abstract

We introduce a real-time capable algorithm which estimates the long-term signal to noise ratio (SNR) of the speech in multi-talker babble noise. In real-time applications, long-term SNR is calculated over a sufficiently long moving frame of the noisy speech ending at the current time. The algorithm performs the real-time long-term SNR estimation by averaging "speech-likeness" values of multiple consecutive short-frames of the noisy speech which collectively form a long-frame with an adaptive length. The algorithm is calibrated to be insensitive to short-term fluctuations and transient changes in speech or noise level. However, it quickly responds to non-transient changes in long-term SNR by adjusting the duration of the long-frame on which the long-term SNR is measured. This ability is obtained by employing an event detector and adaptive frame duration. The event detector identifies non-transient changes of the long-term SNR and optimizes the duration of the long-frame accordingly. The algorithm was trained and tested for randomly generated speech samples corrupted with multi-talker babble. In addition to its ability to provide an adaptive long-term SNR estimation in a dynamic noisy situation, the evaluation results show that the algorithm outperforms the existing overall SNR estimation methods in multi-talker babble over a wide range of number of talkers and SNRs. The relatively low computational cost and the ability to update the estimated long-term SNR several times per second make this algorithm capable of operating in real-time speech processing applications.

Keywords: Multi-talker babble; Long-term SNR; Adaptive SNR; Real-time SNR; Signal-to-noise ratio

1. Introduction

It is essential to have knowledge about the presence and intensity of target speech and background noise when analyzing different segments of a noisy speech sample. Hence, estimating the Signal-to-Noise Ratio (SNR) of noisy speech signals has multiple applications in speech processing, enhancement and recognition (e.g., Ephraim and Malah, 1985; Hirsch and Ehricher, 1995; Morales et al., 2011; Scalart and Filho, 1996; Sohn et al., 1999; Tchorz

* Corresponding author at: Department of Electrical and Computer Engineering, New York University Tandon School of Engineering, 2 Metrotech Center, Brooklyn, NY 11201, USA.
E-mail address: rs4462@nyu.edu (R. Soleymani).

https://doi.org/10.1016/j.csl.2019.05.001
0885-2308/

and Kollmeier, 2003). SNR estimation has been the subject of many previous studies and different categories of SNR estimation algorithms have been introduced for different applications.

One category which has many applications in real time denoising algorithms is the "instantaneous short-term SNR estimation". These algorithms estimate the SNR of relatively short frames of the noisy speech. In recent years, many instantaneous SNR estimation algorithms have been proposed (e.g., Plapous et al., 2006; Elshamy et al., 2017; Sun et al., 2014; Lun and Hsung, 2010). Many of these algorithms use a variety of techniques to improve earlier well-known works including decision-directed (DD) algorithm (Ephraim and Malah, 1984), spectral subtraction (Lim and Oppenheim, 1979), noise power spectral density estimation (Martin, 2001) and noisy speech sub-banding (e.g., Nemer et al., 1999). These algorithms are designed to be highly sensitive to the short-term variations of the signal and noise. Even with a relatively fixed target speech and background noise level, instantaneous short-term SNR estimation methods are likely to yield highly variant SNR values for short segments of the noisy speech. This is mainly due to the inherent temporal fluctuations in both the target speech and background noise. Even though it may be desirable to track these short-term activities for certain denoising applications, instantaneous short-term SNR estimation provides very little insight into the long terms variations of speech and noise. Moreover, these algorithms usually assume a stationary or quasi-stationary behavior for the background noise. Hence de-noising techniques which employ these algorithms generally have a lower performance when the background noise is a non-stationary signal such as multi-talker babble noise (Hu and Loizou, 2007).

Another category of SNR estimation algorithms estimates the overall SNR of the noisy speech over the entire noisy speech. We will refer to this category as "Overall SNR estimation" algorithms. Kim and Stern (2008) used maximum likelihood estimation to find the shaping parameter of a Gamma distribution which models the noisy speech. The obtained value of the shaping parameter was used to estimate the overall SNR of the noisy speech. This algorithm, which assumes the foreground speech and the background noise can be modeled by a Gamma and a Gaussian distribution respectively, performs relatively well in white noise and music background. Papadopoulos et al. (2016) employed features which are sensitive to the presence of speech in noise to estimate the energy of speech and noise in different regions of the noisy speech. These energy values were used to train noise dependent regression models which provide an estimation of the overall SNR. The algorithm was trained with various noise types and different noise-specific regression models were obtained. In the case of unknown noise type, a Deep Neural Network (DNN) using Mel-Frequency Cepstral Coefficients (MFCCs) was trained to determine the type of the noise. Narayanan and Wang (2012) used ideal binary masking to labels time/frequency units of the noisy speech as being either speech or noise dominated and then estimated the long-term SNR based on the energy of each class. These algorithms are less susceptible to fluctuation of the SNR over short intervals. However, because these algorithms need the entire noisy signal, they are inappropriate for real-time applications. Like the first category, these algorithms usually have a lower accuracy when the background noise is non-stationary (Narayanan and Wang 2012).

We developed a new approach to estimate "Real-Time Long-Term SNR" which is a real-time measure of SNR that is independent of the short-term speech or noise activity. This SNR estimate should only change when there is a non-transient change in the intensity of speech or background noise. In order to achieve this, SNR should be measured over a sufficiently long moving-frame ending at the current point in time. In this work, we developed and evaluated the "Adaptive Long-Term SNR estimation algorithm" (ALTIS) which has the following properties:

- ALTIS estimates the long-term SNR which is estimated over longer frames of the noisy speech than instantaneous short-term SNR estimation methods. Using longer frames reduces the sensitivity of the SNR estimation to short-term speech activity.
- ALTIS processes real-time signals in short frames and estimates the long-term SNR over a moving long-frame with an adaptive duration which consists of multiple consecutive short frames. This moving long-frame ends at the current time and has an adaptive duration which is determined based on the events (i.e., non-transient variations of speech or noise intensity) in the noisy speech detected by an event detector. ALTIS updates the estimated long-term SNR after receiving every new short frame from the input.
- ALTIS is designed to perform well with multi-talker babble. Multi-talker babble is one of the most challenging non-stationary noises which is very common place in real life situations.

The estimated long-term SNR is highly affected by the duration of the long-frame over which the SNR is calculated. To have a more accurate estimation, the long-frame must be sufficiently long such that the short-term speech activities

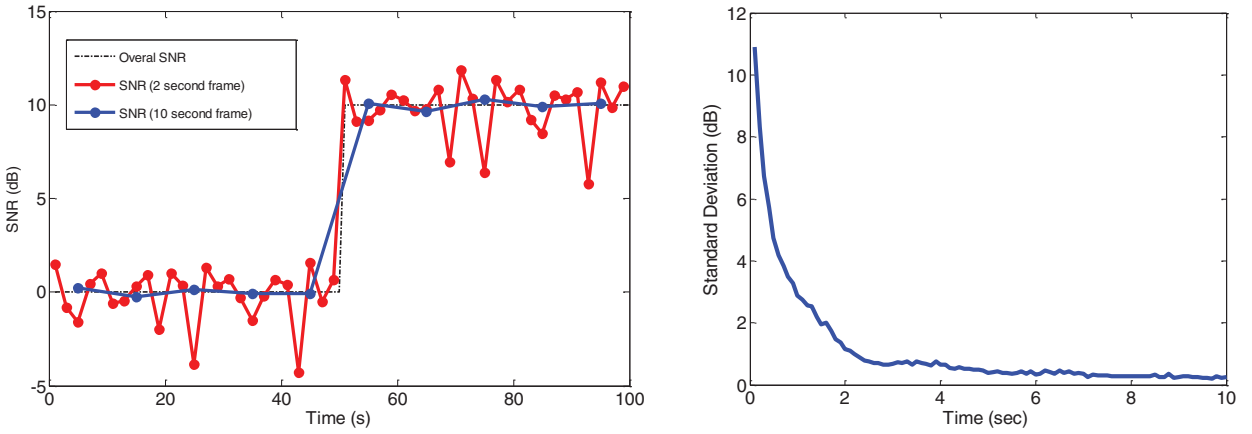

Fig. 1. The results of an ideal SNR estimator with frame lengths of 2 and 10 s. The noisy speech consists of 50 s of noisy speech with overall SNR = 0 dB and 50 s of noisy speech with overall SNR = 10 dB and the background noise is 10 talker babble (left). Standard deviation of the estimated long-term SNR in dB as a function of the long-frame duration in seconds in speech samples corrupted by 10 talker babble (right).

including speech gaps and naturally high or low energy phonemes of the speech do not affect the estimated long-term SNR. Even with an ideal SNR estimator (i.e., assuming prior knowledge of clean speech and noise) and a noisy speech with a fixed long-term SNR, using a "long-frame" with an insufficient duration would lead to an estimated SNR that oscillates around the actual long-term SNR. Increasing the long-frame length will decrease the variance of this oscillation (see Fig. 1 – right). Conversely, choosing overly-long durations for a long-frame will produce long-term SNR estimations that would be insensitive to the changes in the long-term SNR within the long-frames. Furthermore, longer frame durations lead to a slower response to changes in SNR as shown in Fig. 1 (left). To minimize the trade-off between detection accuracy and detection agility, ALTIS uses an adaptive long-frame with a duration that varies based on the noise situation.

The motivation for the development of ALTIS was for implementation in SEDA, which is a real-time wavelet based babble noise reduction algorithm developed by Soleymani et al. (2018). However, the ALTIS algorithm is inherently independent of SEDA and is likely to be useful for other similar applications. We therefore have documented and evaluated ALTIS independently. SEDA uses the long-term SNR estimation to determine a priori probability of observing speech or babble dominated short frames, as well as to adjust the denoising aggressiveness in a wavelet domain.

2. Algorithm

Like most real-time algorithms, the proposed algorithm receives incoming short-frames (each a few tens of milliseconds in duration) of noisy speech as input. First, several features which are sensitive to the level of noise and speech are extracted from every incoming short-frame of the noisy input signal. Then the algorithm employs a DNN to estimate the "speech-likeness" value of each short-frame of the incoming signal based on its normalized extracted features. Then an event detector uses the speech-likeness values of the last few short-frames to detect any possible transitions in the long-term SNR. Subsequently, based on the results of the event detector, the algorithm determines an optimal number of consecutive short frames to form a long-frame over which the long-term SNR is estimated. Finally, a regression model is used to estimate the SNR of the long-frame (i.e., long-term SNR) based on the mean speech-likeness value of its short-frames. Fig. 2 shows the block diagram of the algorithm. More details about different components of the algorithm are given in following sub-sections.

2.1. Feature extraction

For every incoming short-frame, a feature vector of $\overline{\mathrm{F}}_i = [f_1, \mathrm{f}_2, \ldots f_{16}]$ consisting of 16 features is formed. The selected features consist of 12 MFCC coefficients and four additional features taken from Soleymani et al. (2018) that are sensitive to the level of noise in speech. Considering the decreased robustness of MFCC features in noise, adding these four features, significantly increases the performance of the classifier. The selected features are as follows:

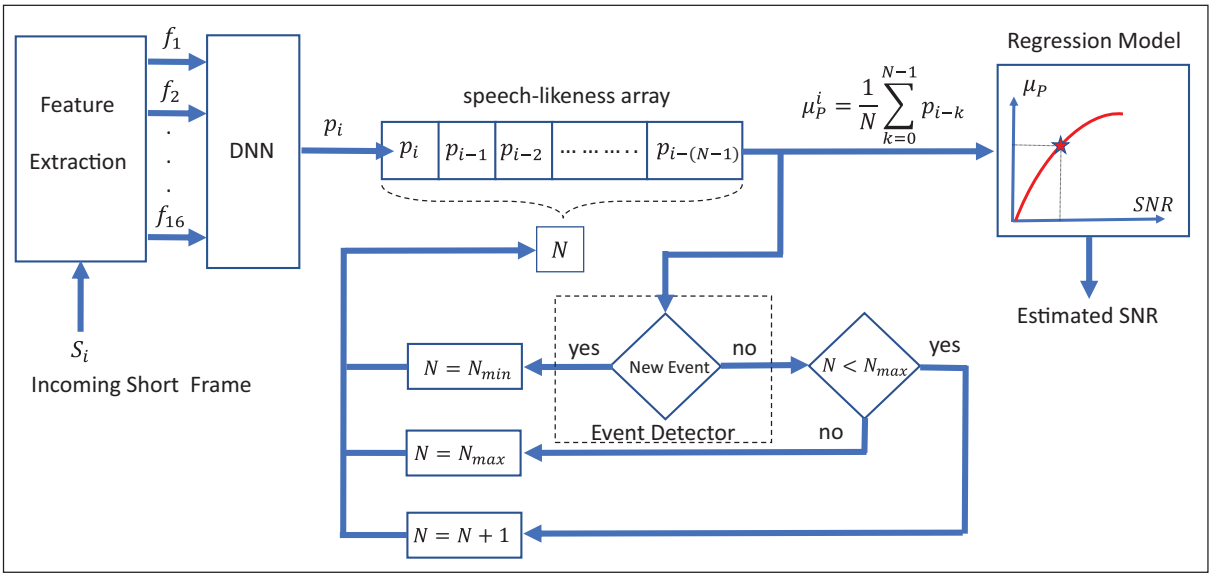

Fig. 2. ALTIS Overall Block Diagram. S_i: a new incoming short frame, f_1 to f_{16}: features extracted from S_i (details in Section 2.1), DNN: Deep Neural Network classifier (details in Section 2.2), p_i: speech-likeness value of S_i (details in Section 2.3), N: number of short-frames in a long-frame over which the long-term SNR is estimated (details in Section 2.5).

- **MFCC Features (f_1 to f_{12}):**

To obtain MFCC coefficients for an incoming short-frame, first a pre-emphasizing high-pass filtering is performed to reduce the concentration of energy in lower frequencies. In this work we used a first order pre-emphasis filter with a filter coefficient $\alpha = 0.97$ (Young et al., 2006). Then we used a Hamming window to enhance the feature quality by reducing the effect of the discontinuities at the borders of the frame. In this work, we selected a 22 channel Mel-scale triangular filter bank which covers frequencies from 200 Hz to 6000 Hz. Usually MFCC values have a wide range of variance which makes it difficult to compare them. To solve this problem, we use a Cepstral Lifter as follows (Young et al., 2006):

$$c_i' = \left(1 + \frac{L}{2}\sin\frac{\pi i}{L}\right) c_i \tag{1}$$

where L is cepstral sine lifter parameter (we used $L = 22$), c_i and c_i' are the cepstral coefficients before and after lifting respectively. Finally, f_1 to f_{12} were obtained as: $f_i = c_i' \ \ (i = 1, 2, \ldots 12)$. Please note that we do not use 0th Mel-Frequency Cepstral Coefficient c_0' in this work.

- **Amplitude Entropy (f_{13}):**

In developing the SEDA algorithm (Soleymani et al., 2018), we determined that the amplitude entropy of short-frames of noisy speech decreases by increasing the noise level in the short-frame. For a short-frame denoted with S, the value of feature f_{13} can be obtained as follows:

$$f_{13} = -\frac{1}{l}\sum_{k=1}^{N_b} h(k) log_{10}\left(\frac{h(k)}{l}\right) \tag{2}$$

Where l is the short-frame's length, h is the amplitude histogram of S and N_b is the number of bins. The main parameter that affects the quality of this feature is the histogram bin width (bw). To find the value of bw that maximizes the quality of this feature we used the Fisher score (Tang et al., 2014; Gu et al., 2012; Duda et al., 2001) that is widely used to estimate the quality of features and can be calculated as follows:

$$\frac{\sum_{k=1}^{C_n} n_k(\mu_k-\mu)^2}{\sum_{j=1}^{C_n} n_k\sigma_k^2} \tag{3}$$

where C_n is the number of classes, μ and μ_k are the mean values of the feature in all classes and in class k, respectively, σ_k is the standard deviation of the feature in class k and n_k is the number of samples in class k. When f_{13} is calculated for speech in multi-talker babble, the histogram bin width which maximizes the Fisher quality score of this feature is $bw = 0.05M$ where M is the long-term maximum of the noisy amplitude (Soleymani et al., 2018). Note that l is a constant and does not affect the feature quality. Hence, we can simplify (2) to: $f_{13} = -\sum_{k=1}^{N_b} h(k)log_{10}(h(k))$.

- **post-thresholding to pre-thresholding RMS ratio** (f_{14})**:**

Hard threshold of a noisy speech frame $\mathrm{S} = [s_1, s_2, \ldots. \ s_l,]$, with threshold level τ can be defined as: $T(\mathrm{S}, \tau) = \{s_1^\tau, s_2^\tau, \ldots, s_l^\tau\}$ where: $s_i^\tau = \begin{cases} 0, & |s_i| \le \tau \\ s_i, & |s_i| > \tau \end{cases}$.

Using hard thresholding, the value of f_{14} for a noisy speech short-frame S can be obtained as follows:

$$f_{14} = \frac{\| T\left(S\ ,\ \frac{1}{l}\mathrm{K}\| S\|_1\right)\|_2}{\| S\|_2} \tag{4}$$

Our experiments show that if the thresholding level in (4) is selected carefully (i.e., above the noise base level), samples originating from the target speech are more likely to survive the hard thresholding. Hence the value of this feature increases by increasing the signal to noise ratio in the noisy speech frame S. Selecting $K = 3$ maximizes the Fisher quality score of this feature. (Soleymani et al., 2018).

- **Temporal Envelope Variance** (f_{15})**:**

For a short frame $S = [s_1, s_2, \ldots. s_l,]$ the envelope e_S can be obtained as follows:

$$e_S(n) = \frac{1}{l_e}\sum_{k=-\frac{l_e}{2}}^{\frac{l_e}{2}-1} \left|S(k+nh)\right| w(k) \tag{5}$$

where l_e is the length of the moving average window w, and h is the hop value. The value of f_{15} can be calculated as:

$$f_{15} = \frac{1}{n_w}\sum_{n=1}^{n_w}\left(\frac{e_S(n)}{max(e_S)} - \frac{1}{n_w}\sum_{n=1}^{n_w}\frac{e_S(n)}{max(e_S)}\right)^2 \tag{6}$$

where n_w is the total number of windows in a short-frame. In this work, we used non-overlapping rectangular windows with $h = l_e = 50$ to maximize the feature's quality score. The value of f_{15} decreases with increasing the noise (Soleymani et al., 2018).

- **Envelope Mean-Crossing rate** (f_{16})**:**

Using (5) the value of f_{16} can be calculated as follows:

$$f_{16} = \frac{1}{2n_w}\sum_{\mathrm{k}=2}^{n_w}\left|\mathrm{sign}\left(e_S(k) - \frac{1}{n_w}\sum_{n=1}^{n_w} e_S(n)\right) - \mathrm{sign}\left(e_S(\mathrm{k}-1) - \frac{1}{n_w}\sum_{n=1}^{n_w} e_S(n)\right)\right| \tag{7}$$

where sign(x) is the sign function of x. The value of this feature increases with increasing the noise.

Features numbered 13–16 were previously used for classification of speech dominated and noise dominated frames in SEDA. These four features are directly extracted from the incoming short-frames of the signal before pre-emphasizing or applying the Hamming window. For a further discussion of using each of these features on noisy speech frames, please see Soleymani et al. (2018).

2.2. DNN based speech-noise classifier

At this stage, using a DNN we design a classifier which can accurately discriminate single talker (i.e., clean) speech and multi-talker babble (4 to 20 talkers). The DNN was trained with short-frames (frame duration = 128 ms) of clean speech and multi-talker babble. We used a database 2100 sentences, including 720 male speakers and 720 female speaker IEEE standard sentences (IEEE Subcommittee, 1969), 260 male speaker HINT sentences (Nilsson et al., 1994) and 400 male speaker SPIN sentences (Bilger et al., 1984) to create clean speech and multi-talker babble. Babble samples were generated with a random number (between 4 and 20) of sentences spoken by either or both genders.

Human speech naturally has low-energy gaps which consist of silence or background noise. We used a simple energy-based gap detector to identify and remove these gaps from the single talker training material. This prevents the classifier from being trained by the non-speech gaps labeled as speech. We created one hour of randomly generated multi-talker babble and one hour of clean speech for training the classifier (i.e., 28,125 short frames for each class). In this work we used the sampling rate of 16,000 samples per second (i.e., frame length of 2048 samples).

After creating and labeling the babble and clean speech frames, we extracted the previously discussed features from each frame and created a training dataset. The obtained feature matrix $M_{train} = (f_{ij})_{n_s \times n_f}$, consisted of n_s feature vectors (i.e., number of short-frames) with $n_f = 16$ features per vector where, f_{ij} is the jth feature of the ith short frame. Another feature matrix M_{test} was created for testing the classifier. To ensure that the classifier is tested with unseen data, multi-talker babble and clean speech samples which were used for training and testing the classifier were generated from different sentence lists with different speakers.

Because the cepstral liftering described in Section 2.1 is only applied to the original MFCC features, at this stage we perform a mean and variance normalization on final DNN input features including 12 selected MFCCs and four additional non-MFCC features. The values of the feature matrix M_{train} were normalized as follows:

$$\widehat{M}_{train} = \left(\widehat{f}_{\mathrm{ij}}\right)_{n_s \times n_f} \text{ where, } \widehat{f}_{\mathrm{ij}} = \frac{f_{\mathrm{ij}} - \mu_j}{\sigma_j}, \mu_j = \frac{1}{n_s}\sum_{i=1}^{n_s} f_{\mathrm{ij}}, \sigma_j = \sqrt{\frac{1}{n_s}\sum_{i=1}^{n_s}\left(f_{ij} - \mu_j\right)^2} \tag{8}$$

The same normalization process was repeated for test feature matrix M_{test}. Note that the test feature matrix was normalized under real-time assumption in which feature means and variances used for normalization of each feature vector (i.e., each row of M_{test}) were calculated based on the previous feature vectors and were updated after receiving a new short-frame. Then each of these normalized test feature vectors were separately used as an input to the DNN based classifier. Using the normalized feature matrix $\widehat{M}_{train}$, we trained a Deep Neural Network (DNN) with 16

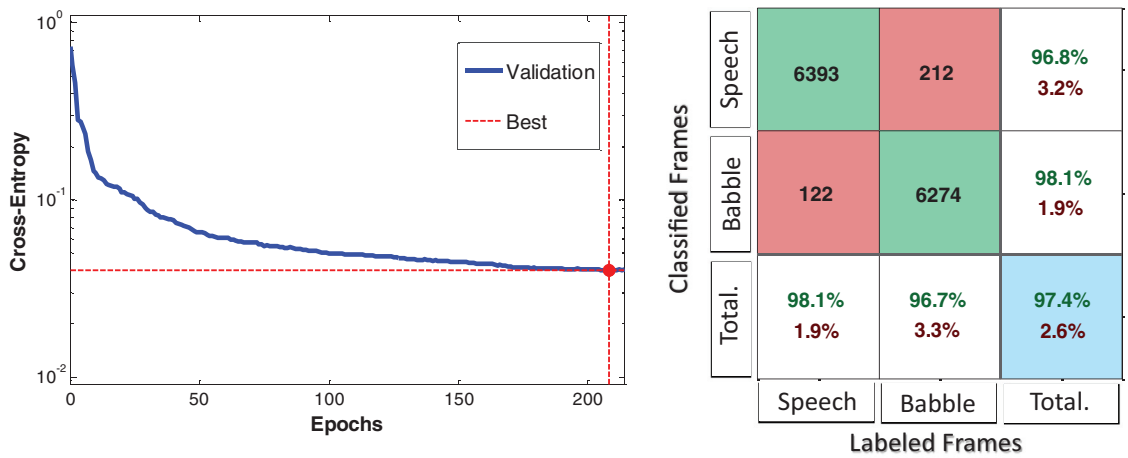

Fig. 3. Classifier cross-entropy vs. number of epochs. Dashed line shows the best performance which is taken from the epoch with the lowest validation error (left). Classification Confusion Matrix. Green and red cells show the number of correct and incorrect classifications respectively. White cells show the percentage of correct and incorrect classification for each case (speech or noise). Blue cell shows the overall percentage of correct and incorrect classification (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

inputs (number of features per frame), two hidden layers each with 16 nodes and two outputs (number of classes). DNN was trained using the Scaled Conjugate Gradient Algorithm (Moller 1993).

The number of epochs in training is the number of times that the entire training data set is presented to the neural network. The neural network is updated after each epoch. The classifier performance with the number of epochs that minimizes the cross-entropy validation error (i.e., 208) was calculated (left panel Fig. 3). Performance of the classifier is presented in a confusion matrix (right panel Fig. 3). Clean speech, multi-talker babble and total detection performances of the classifier are 98.1%, 96.7% and 97.4%, respectively. The additional improvement provided by increasing the number of layers or nodes was not large enough to justify the corresponding additional computational requirements.

2.3. "Speech-likeness" estimation

The classifier described in Section 2.2 outputs a value $0 \leq p_i \leq 1$ which is a measure of the likelihood that the frame is either clean speech or multi-talker babble. The higher the value of p_i, the more likely the frame is clean speech. However, if a frame containing combined single-talker target speech and background multi-talker babble is presented to the classifier, the appropriate interpretation of the output value of p_i will be different. Because we already know that the frame is neither clean speech nor noise alone, p_i can be interpreted as an estimate of the similarity of the frame to clean speech. In this case, we refer to this value as a "speech-likeness" estimation. Our experiments demonstrate that the average p_i value of a sufficiently large number of consecutive short-frames of the noisy speech can be mapped to an accurate estimate of the SNR value of the long-frame which is the result of concatenation of the above mentioned consecutive short-frames.

As mentioned in the introduction, in a real-time application, the estimation of the long-term SNR can be defined as measuring the SNR over a long-frame with the duration of t_L which ends at the current time (t_i) and starts at $t_i - t_L$. To minimize the latency, audio signals in real time algorithms are usually received and processed in short frames (i.e., a few milliseconds in duration). Assuming S_i is the latest short-frame of the noisy speech which has been received from the input at time t_i, we can define the long-frame F_L^i as the concatenation of the last N short-frames as follows:

$$F_L^i = \left[S_{i-(N-1)}, S_{i-(N-2)} \ldots, S_{i-1}, S_i\right], N = \frac{t_L}{t_S} \tag{9}$$

where F_L^i is the long frame of the noisy speech which ends at t_i, and t_L and t_S are durations of the long and short frames respectively. For the long frame of F_L^i we define array P_L^i and its mean μ_P^i as follows:

$$P_L^i = \left[p_{i-(N-1)}, p_{i-(N-2)} \ldots, p_{i-1}, p_i\right], \mu_P^i = \frac{1}{N}\sum_{k=0}^{N-1} p_{i-k} \tag{10}$$

where p_i is the "speech-likeness" value of S_i obtained from the classifier. Our experiments show that if N is sufficiently large, the value of μ_P^i provides an accurate estimate of the SNR in the long-frame F_L^i. Assuming a fixed duration for long-frames, in the event of receiving a new frame of S_{i+1}, the new P_L^{i+1} can be created by adding the speech-likeness value of the new short frame to the end of the old P_L^i and discarding the speech-likeness value of the oldest short frame from its beginning. The μ_{PL} for the new long frame will be simply calculated as:

$$\mu_P^{i+1} = \frac{N\mu_P^i + p_{i+1} - p_{i-(N-1)}}{N} \tag{11}$$

We do not need to buffer the entire audio signal within the long-frame to compute its SNR. The only necessary information is the array of P_L which contains the speech-likeness values for the last N short-frames. We will discuss the process of estimating the long-term SNR from μ_P in the next section. In practice, we use a long-frame with a length that will vary based on the events in the noisy speech. A modified version of Eq. (11) for an adaptive long-frame will be given in Section 2.5.

Fig. 4 shows the scatter plots, means and standard deviations of μ_P values for a large number of long-frames of noisy speech with SNR values ranging from −10 dB to 20 dB for two different long-frame durations. As can be seen, the standard deviation of the μ_P values decrease with an increase in the duration of the long-frame and as we will discuss, this will lead to a more accurate SNR estimation.

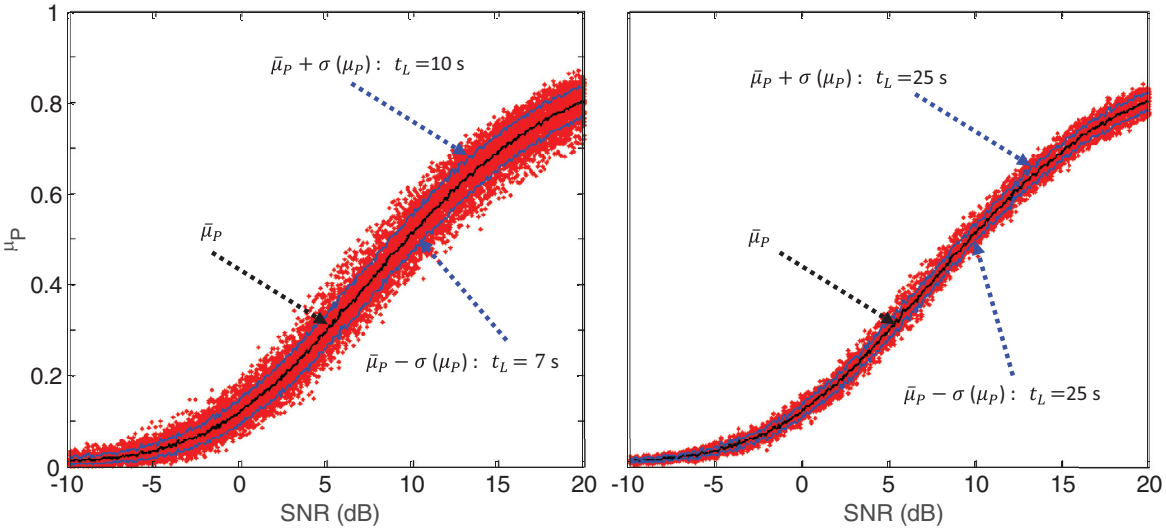

Fig. 4. Scatter plot of μ_P values (red dots), the mean μ_P (black curve) and ± 1 standard deviation (blue curves) in different SNRs for long-frame durations of 7 s (left) and 25 s (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

2.4. Long-term SNR estimation using regression models

Our previous experiments showed that the value of μ_P changes consistently as a function of the long-term SNR. To estimate the value of the long-term SNR of a long-frame from the value of its μ_P we need a regression model which describes the relationship between the two values. To obtain this regression model, in this work, we used Ezy-Fit toolbox (Moisy, 2016) that is designed based on a variation of the Nelder-Mead method (Nelder and Mead, 1965). This method uses a numerical non-linear optimization algorithm to minimize the differences between the estimated model and the distribution of the actual μ_P values with respect to different parameters of the model (Mathews and Fink, 2004; Lagarias et al., 1998). Our experiments suggest that the resulting regression models are independent of the long-frame duration but vary with the number of talkers in babble. Hence different models were obtained for different numbers of talkers in babble (grey curves in Fig. 6).

Our experiments show that for different numbers of talkers, the resulting functions are always well approximated by the sum of two, three or four Gaussian functions with different parameters. For example, the blue dashed line in Fig. 5 and Eq. (12) show the Gaussian function which is the best fit for the average μ_P values of long-frames of noisy speech corrupted with 10 talker babble.

$$\mu_P\,(snr) = 1.276e^{\frac{-(snr-9.268)^2}{908.147}} - 1.026e^{\frac{-(snr+0.774)^2}{381.874}} \tag{12}$$

The problem with having different regression models for different number of talkers is that in practice the number of talkers in babble is not always known. To solve this problem, we create a talker independent regression model which is estimated using a large number of long, noisy speech frames with a randomly selected number of talkers (between 4 and 20).

$$\mu_P\,(snr) = \begin{cases} 0.051e^{\frac{-(snr+87.305)^2}{12602}} + 0.457e^{\frac{-(snr-11.914)^2}{103.059}}, & snr \leq 5dB \\ 1.9363e^{\frac{-(snr+22.358)^2}{3685.3}} + 1.8723e^{\frac{-(snr+9.97)^2}{574.673}}, & snr > 5dB \end{cases} \tag{13}$$

The resulting "number of talker independent" model will be approximately the average of the individual models for different number of talkers. Eq. (13) and the red dashed line in Fig. 6 show the best fit function for the relationship between μ_P and long-term SNR of noisy speech, given the number of talkers in babble is unknown. As expected

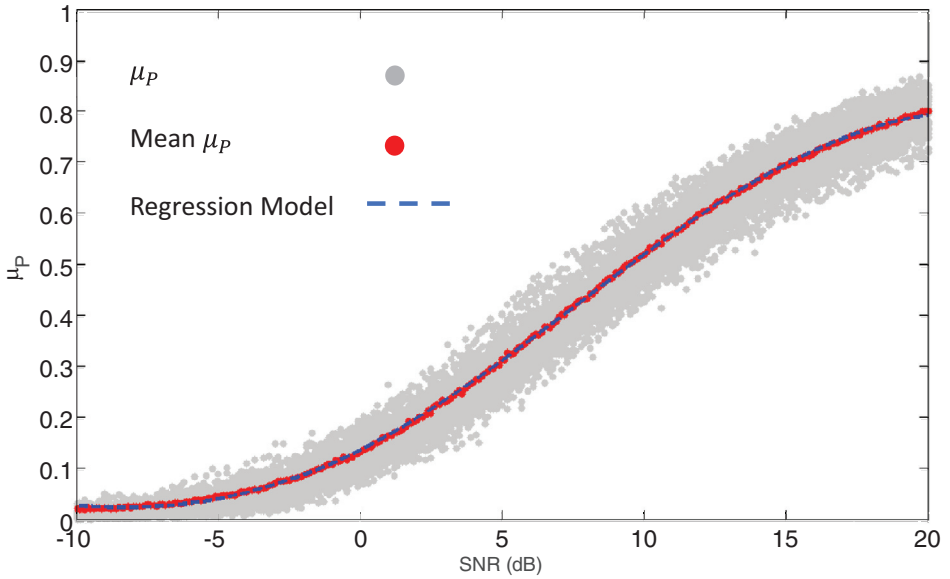

Fig. 5. The regression model of long-term SNR (dashed blue line) plotted as a function of μ_P for 10-talker babble. The grey dots represent the average speech-likeness values (μ_P) of long-frames with random durations between 10 and 30 s. The mean of average speech-likeness values ($\overline{\mu}_P$) at each SNR are represented by red dots. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

using the "number of talker independent" model will introduce some SNR estimation bias mainly when the number of talkers is very small (i.e., less than 6 talkers) and generally performs slightly poorer than the "number of talker specific model". Using the obtained regression models, the estimated SNR value of a long-frame can be found quickly based on its μ_P. Because the resulting regression models do not have a closed form solution for SNR, we use a lookup table search to find the estimated value of SNR. Note that because in realistic situations the SNR value varies within a relatively narrow range (e.g., between -10 and +20) and it is estimated with limited precision (one or two decimal places) the lookup table of μ_P values for all possible estimated SNR values is not very large.

2.5. Event detector and adaptive long-frame

As discussed in previous sections, an increased duration of a long-frame (i.e., the more short-frames used to estimate the long-frame average speech-likeness value μ_P) reduces the variance of μ_P and therefore the improves the accuracy of the SNR estimation. However, in real-time applications, when the actual long-term SNR changes, using very long frames for long-term SNR estimation leads to long and undesirable estimation transition times (see Fig. 8 plot 2).

If the long-term SNR of noisy speech changes from one value (SNR_1) to another (SNR_2) and the transition starts at the time t_1 and ends at the time t_2, the transition time will be $t_t = t_2 - t_1$. When the long-term SNR changes, we expect the estimated long-term SNR to change accordingly and we define the "estimation transition time" as the duration that the estimated long-term SNR changes from one stable value ($\sim SNR_1$) to another ($\sim SNR_2$). Although both true and estimation transition times are caused by the change of SNR, the estimation transition time is always longer than the true transition time. It is desirable for estimation transition times to be as close as possible to true transition times for applications such as de-noising.

The difference between the true and estimation transition time decreases with decreasing duration of the long-frame. Fig. 7 shows the effect of long-term frame duration on estimation transition time. As seen in this figure at $t < t_1$ before the SNR changes, the long frame (marked with 1) contains only the noisy speech with long-term SNR value of SNR_1. Exactly at the time $t = t_1$ the actual SNR transition begins and continues until $t = t_1 + t_t$. The estimation transition period also starts at $t = t_1$ when the actual SNR transition begins but it continuous until $t = t_1 + t_t + t_L$ when the entire transition period is out of the long-term frame (marked with 6). As illustrated in Fig. 7, during the estimation transition period, the long-term frame (marked with 3, 4, 5) contains noisy signals with various overall

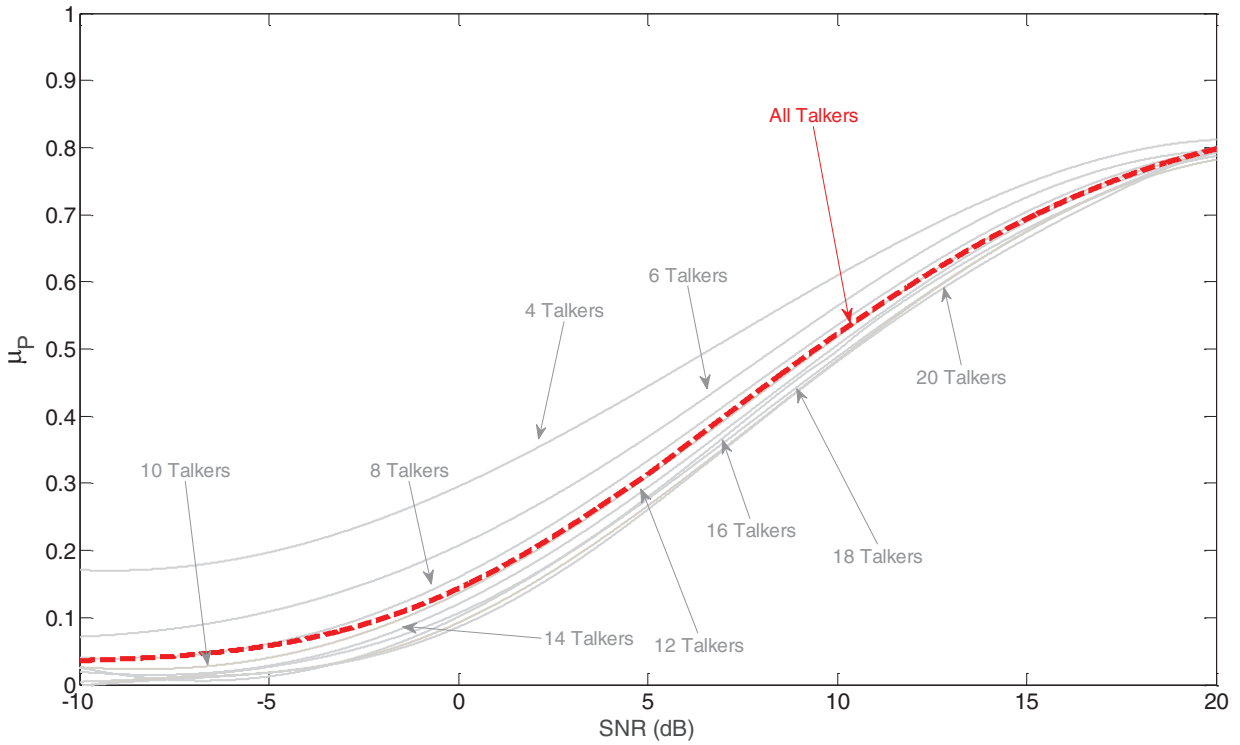

Fig. 6. Regression models for different numbers of talkers (grey curves) and the regression model generated using a random number of talkers (between 4 and 20; dashed red curve). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

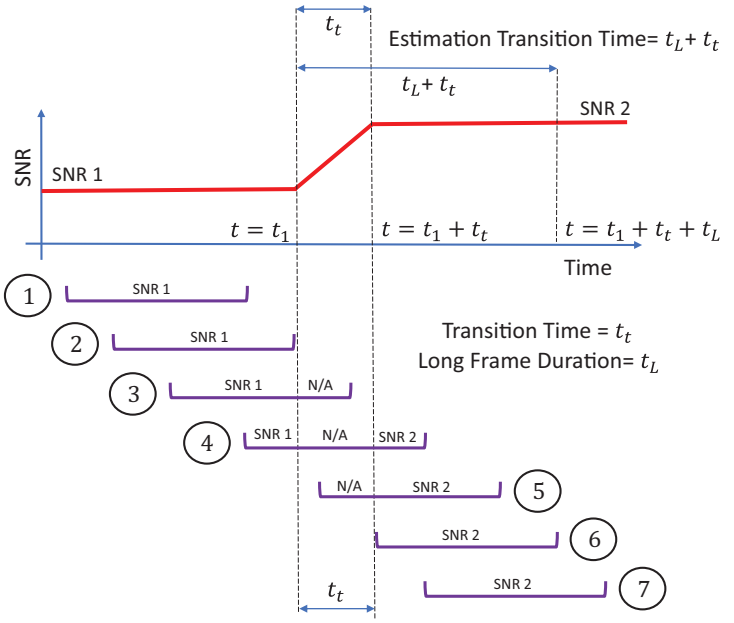

Fig. 7. Actual and estimation transition times. The SNR variation with time is presented in red. A moving long-frames at each time point is shown in purple. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

SNR values (i.e., SNR1, SNR2 and the momentarily changing SNR of actual transition period). Because the duration of each portion (SNR1, SNR2 and actual transition) in the long-term frame constantly changes during this period (from stages 2 to stage 6 in Fig. 7), the overall detected SNR also is subject to change until the entire long-term frame is filled with noisy signal with long-term SNR value of SNR2 and this will happen at $t = t_1 + t_t + t_L$. Hence the total estimation transition time will be $t_E = t_L + t_t$.

To reduce the estimation transition time, we should use a shorter long-frame when the actual transition happens. To take advantage of higher detection accuracy of longer frames and a shorter transition time of shorter frames we use an event detector to adjust the duration of the long frame in different situations. First, we should set a minimum and a maximum duration for the long-frame. As shown in Fig. 1 (right plot), even an ideal long-term SNR estimate with relatively short long-frames (i.e., shorter than 5 s) yields a highly variant result in a fixed long-term SNR. Hence, we select the lower limit to be $t_{L_{\min}} = 5.12$ s which is 40 times longer than selected duration of the short-frames (128 ms). We select the upper limit to be $t_{L_{max}} = 48$ s which is 375 times longer than selected duration of the short frames (128 ms). The duration of the adaptive long-frame will change between these two values as below:

$$N_{min} = \frac{t_{L_{min}}}{t_s} = \frac{5120 \text{ ms}}{128 \text{ ms}} = 40, N_{max} = \frac{t_{L_{max}}}{t_s} = \frac{48000 \text{ ms}}{128 \text{ ms}} = 375, N_{min} \leq N \leq N_{max} \tag{14}$$

where N_{min}, N_{max}, N are the number of short-frames (128 ms) in long-frames with minimum, maximum and adjustable durations respectively.

We determine the adjustable long-frame duration by monitoring the noisy signal's behavior and detecting changes in the long-term SNR. As long as the long-term SNR is constant and no event is detected, the long-frame duration gradually increases. The long-frame duration will stop increasing when it reaches its maximum level of $t_{L_{max}}$. In the case of detecting a new event, the long-frame's duration immediately drops to the minimum level of $t_{L_{min}}$ to reduce the estimation transition time. If the actual SNR reaches a new stable level after the change, the long frame duration will start to increase again.

We define an event as a change in the long-term SNR of the noisy speech from one stable level to another. This change should be in one direction (i.e., up or down, and not just a temporary increase in variability) and must remain at the new SNR for a certain duration before the long-term SNR changes again. Short term oscillatory variations of the SNR (i.e., instantaneous SNR) around one stable level which usually are the result of natural transient fluctuations in speech or babble activity levels should not be detected as event.

We define an event window to have a duration of $t_e \geq 2t_{L_{min}}$. This event window is updated once every t_s (i.e., the duration of the short-frame) by adding a newly received short frame to the end of the event window and discarding the oldest short-frame which keeps the length of the event window constant. Now we divide the event window into two equally long parts with the duration of $\frac{t_e}{2} \geq t_{L_{min}}$ and calculate the absolute value of the difference between mean speech-likeness values of the two parts. A significant difference between the mean speech-likeness values of the two parts is interpreted as a change of long-term SNR and therefore is considered a new event. The length of each part (i.e., each half of the event window) should be long enough (i.e., equal or longer than minimum window length) to ensure that the detected difference is not the result of natural fluctuation of speech and babble.

In practice, event detection does not impose any extra computational cost to the algorithm. All the speech-likeness values required for the event detection have been already calculated for long-term SNR detection and we can use the already available data for the event detection. The mean speech-likeness difference between the two halves of the event-window can be calculated as:

$$\Delta_{\mu}^{e} = \left| \frac{\sum_{k=0}^{\frac{N_e}{2}-1} p_{i-k} - \sum_{k=\frac{N_e}{2}}^{N_e-1} p_{i-k}}{N_e} \right| \tag{15}$$

where Δ_{μ}^{e} is the mean speech-likeness difference between the two halves of the event-window, $N_e = \frac{t_e}{t_s}$ is the number of short-frames in the event-window and $P_e = [p_i, p_{i-1} \ldots, p_{i-(N_e-1)}]$ is the speech-likeness array of the event-window at t_i. Because speech-likeness values vary between zero and one, we will always have $0 \leq \Delta_{\mu}^{e} \leq 1$.

If the current time is t_i, the first half of the event window contains the speech likeness values of short-frames between $t_i - 2t_{tr}$ and $t_i - t_{tr}$ and the second half contains the speech likeness values of short-frames between $t_i - t_{tr}$

and t_i. If the long-term SNRs of these two consecutive time frames are different, the values of Δ_μ^e will become greater than an event threshold value of $0 < \alpha < 1$ and a new event will be detected.

The value of α should be selected carefully so that the natural variances of Δ_μ^e are not detected as a new event. One way to choose α is to measure natural variance of Δ_μ^e in constant long-term SNR and set: $\alpha \gg \sigma(\Delta_\mu^e)$. Our experiments show that the added mean absolute error due to the false positive event detection, is less than the added error resulting from a false negative event detection. Hence, we should deliberately tune the value of α to make the false negative detections of an event less likely than the false positive detection. Our selected value is $\alpha = 0.2$.

Assuming N_{old} is the long-frame length (i.e., number of short-frames in the long-frame) at t_i, using the event detector we calculate N_{new} which is the number of short-frames in the long-frame upon receiving of the next short frame (at $t_i + t_s$) as:

$$N_{new} = \begin{cases} N_{max}: & \Delta_\mu^e < \alpha\ , \quad N_{old} = N_{max} \\ N_{old}+1: & \Delta_\mu^e < \alpha\ , \quad N_{old} < N_{max} \\ N_{min}: & \Delta_\mu^e \geq \alpha \end{cases} \tag{16}$$

where N_{min} and N_{max} are the minimum and maximum number of short-frames in adaptive long-frame of the noisy speech, respectively (see Eq. (14)).

Using adjustable long-frame duration, we can update Eq. (11) which calculates the value of μ_P^{i+1} based on μ_P^i. For a long-frame F_L^i we can write:

$$\mu_P^i = \frac{1}{N}\sum_{k=0}^{N_{old}-1} p_{i-k}, P_L^i = \left[p_{i-(N_{old}-1)}, p_{i-(N_{old}-2)} \cdots, p_{i-1}, p_i\right]$$

where P_L^i and μ_P^i and N_{old} are the speech-likeness array of F_L^i, its average and its length (number of short-frames in the long-frame) respectively. The speech likeness array of the next long-frame F_L^{i+1} is denoted with P_L^{i+1} and can be written as:

$$P_L^{i+1} = \left[p_{i+1-(N_{new}-1)}, p_{i+1-(N_{new}-2)} \cdots, p_i,\ p_{i+1}\right]$$

where N_{new} is the number of short-frames in F_L^{i+1}. Having μ_P^i the value of μ_P^{i+1} for adjustable long-frame duration, can be obtained as follows:

$$\mu_{PL}^{i+1} = \begin{cases} \dfrac{N_{max}\mu_{PL}^i + p_{i+1} - p_{i-(N_{max}-1)}}{N_{max}}: & N_{new} = N_{old} = N_{max} \\ \dfrac{N_{min}\mu_{PL}^i + p_{i+1} - p_{i-(N_{min}-1)}}{N_{min}}: & N_{new} = N_{old} = N_{min} \\ \dfrac{N_{old}\mu_{PL}^i + p_{i+1}}{N_{new}}: & N_{new} = N_{old}+1 \\ \dfrac{1}{N_{min}}\sum_{k=-1}^{N_{min}-2} p_{i-k}: & N_{new} = N_{min} < N_{old} \end{cases} \tag{17}$$

Using Eq. (17), the value of μ_{PL} can be updated after receiving a new short-frame. An estimated long-term SNR will be obtained for every updated value of μ_{PL} using the regression models discussed in Section 2.4. Fig. 8 shows the benefit of using an adjustable long-frame relative to a fixed length long-frame. The center panel of Fig. 8 illustrates that a long frame with a larger duration (e.g., 35 s) yields accurate results but is slow to respond to changes in SNR. The right panel of Fig. 8 illustrates that a long frame with a shorter duration (e.g., 7 s) yields fast response to changes in SNR but is inaccurate. However, the left panel of Fig. 8 shows that using adaptive frame-length simultaneously reduces the estimation transition time and increases the accuracy.

3. Performance evaluation

We have evaluated the following two variations of the ALTIS algorithm:

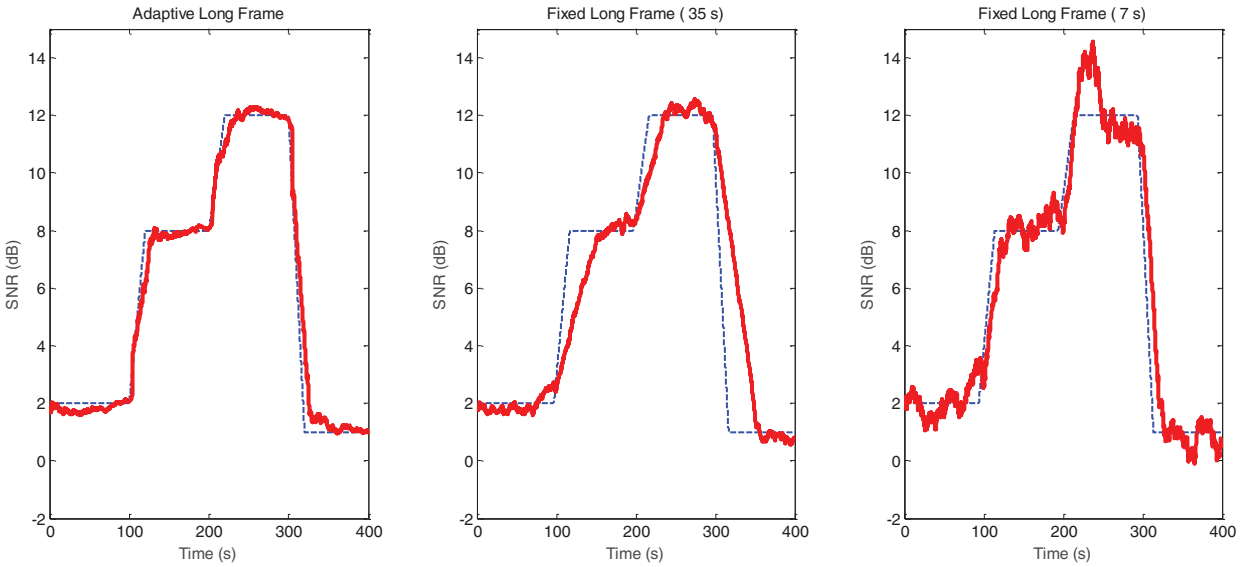

Fig. 8. The actual (dashed blue lines) and detected long-term SNR (red line) in a dynamic SNR situation using the algorithm. The left plot shows the detection results with an adaptive long-frame duration. The center plot shows detection results with a fixed long-frame duration of $t_s = 35\ (s)$ which is accurate but slow to adjust. The right plot shows detection results with a fixed long-frame duration of $t_s = 7\ (s)$ which is fast to adjust but inaccurate. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

I ALTIS using "number of talker independent" regression models for unknown number of talkers in babble (ALTIS-NTI).

II ALTIS using "number of talker specific" regression models for known number of talkers in babble (ALTIS-NTS).

First, each variation of the algorithm was tested at fixed SNRs. Second, each variation was tested in a dynamic SNR situation where the SNR randomly changed over time. All tests were performed using the real-time version of ALTIS with 128 ms non-overlapping frames, meaning that the algorithm did not have access to the entire noisy speech and received the noisy speech frame by frame, and estimated the long-term SNR once every 128 ms. The selected metric to measure the ALTIS performance was "Mean Absolute Error" of the detected long-term SNR.

3.1. Performance in fixed SNR

Both variations of the ALTIS algorithm were tested with a fixed SNR. For the number of talker specific case, regression models were selected based on the number of talkers in the background babble. In both cases the performance was evaluated for noisy speech samples corrupted by multi-talker babble with all integer SNRs between −5 dB and 15 dB. For each SNR, 100 noisy speech samples were created, each having a duration of 100 s. To ensure that speech and babble are evenly distributed throughout the 100 s duration of the noisy sample, each 100 s test sample was created by concatenation of 10 shorter segment with the same SNR (each segment 10 s long).

The number talkers were randomized (between 4 and 20 talkers) for each segment. The proportion of female and male speakers in the babble was also randomized. Every 128 ms, the algorithm generates a new SNR estimation. The mean absolute error between the known SNR and the estimated SNR was calculated for all of the 100 s test samples. The process was repeated for 21 SNR levels with 100 samples. Fig. 9 (top plot) shows the average mean absolute error as a function of SNR for both ALTIS-NTI and ALTIS-NTS.

Our experiments show that the ALTIS performance quickly degrades in SNRs below −5 dB mainly due to the fact that in those SNRs the babble is so strong that makes it difficult for the algorithm to differentiate between the target speech and background babble speeches.

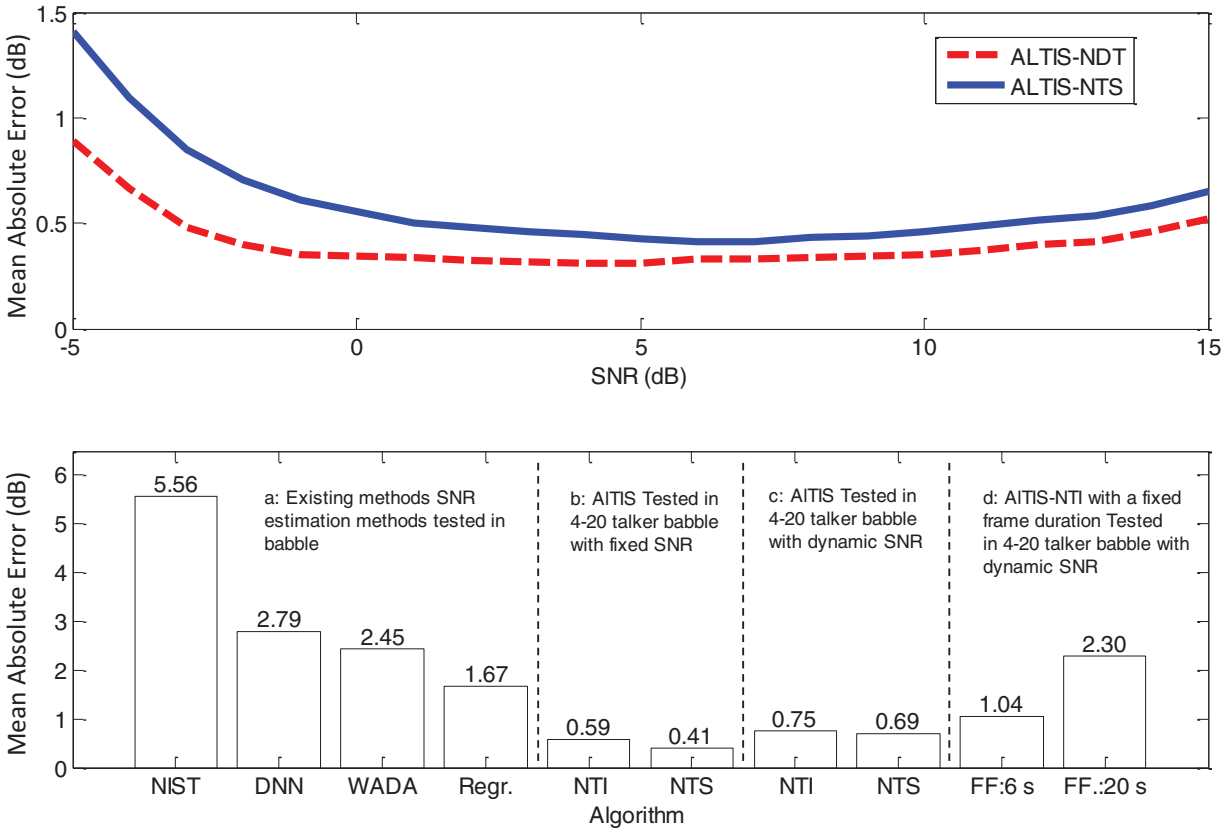

Fig. 9. Average mean absolute error of the long-term SNR estimation for two variations of ALTIS in fixed SNRs (top plot). Average mean absolute error of four existing overall SNR estimation methods (i.e., NIST, DNN, WADA and "Regr") taken from Papadopoulos et al. (2016) in multi-talker babble with unknown number of talkers (bottom plot section a). Average mean absolute error of ALTIS-NTI and ALTIS-NTS in babble with fixed SNR and 4−20 number of talkers (bottom plot section b). Average mean absolute error of ALTIS-NTI and ALTIS-NTS in babble with dynamic SNR and 4−20 number of talkers (bottom plot section c). Average mean absolute error of ALTIS-NTI with fixed frame duration of 6 s (FF:6 s) and fixed frame duration of 20 s (FF:20 s) in babble with dynamic SNR and 4−20 number of talkers (bottom plot section d).

3.2. Performance in dynamic SNR

We repeated the evaluation tests in a dynamic babble situation where the actual long-term SNR changes with time. In this case, for both conditions (known and unknown number of talkers) we created 100 noisy speech samples. Each noisy speech sample was 5 min long consisting of 5 1-min long segments. Each segment had a randomly selected SNR between −5 dB and 15 dB.

Additionally, ALTIS-NTI was evaluated with a dynamic SNR without the event detection and adaptive frame length. Instead, a fixed frame length was implemented (either 6 or 20 s). The fixed frame-length variant was evaluated to determine the effectiveness of the event-based adaptive frame-length implementation. As expected, test results show that in dynamic noisy situation the performance is significantly better when an adaptive long-frame is used (see Fig. 9 bottom plot).

To the best of our knowledge there is no other algorithm for "Adaptive Long-Term SNR" estimation with a reported performance in multi-talker babble. Hence, we compared the performance of ALTIS with average results of four existing "overall SNR" estimation methods in multi-talker babble extracted from Papadopoulos et al. (2016) (see Fig. 9 bottom plot). While these algorithms perform well in many stationary noises, their performances in multi-talker babble is generally poor and ALTIS outperforms all four algorithms when the background noise is multi-talker babble.

Note that ALTIS is capable of working in real-time and dynamic noisy conditions and updates its long-term SNR estimate once every few tens of milliseconds whereas other existing overall SNR estimation algorithms only provide one SNR estimate for an entire duration of an available noisy signal. In addition to ALTIS that is specifically trained for multi-talker babble, in this evaluation, *Regr.* algorithm (Papadopoulos et al., 2016) also directly uses a babble-specific regression model. This might partly explain the relatively large gap between the performances of ALTIS and *Regr* with the other three general purpose algorithms (i.e., WADA, DNN and NIST). Moreover, the properties of the babble noise used for testing the algorithms reported in Papadopoulos et al. (2016) are not sufficiently documented. Hence, it is unclear how appropriate the comparison between the test results of ALTIS and other existing algorithms is.

4. Discussion

In the present manuscript, we introduced ALTIS which is an algorithm capable of providing an adaptive and real-time estimate of the long-term SNR when speech is corrupted by multi-talker babble. It was specifically trained and tested with multi-talker babble noise as the algorithm was originally designed to be implemented as a component of the SEDA (Soleymani et al., 2018) babble noise reduction algorithm. However, ALTIS could be trained with other types of noise and function independently or as a component of another algorithm. SEDA uses a priori information provided by ALTIS to increase the performance of its babble/speech dominated short frame classifier (i.e., lower/higher long-term SNR indicates a higher/lower probability of observing noise dominated short frames). It also uses the estimation of the long-term SNR to adjust thresholding levels in a wavelet domain. Similar algorithms such as those that classify noisy speech short frames (or time/frequency tiles) as being either speech or noise dominated (e.g., algorithms that employ binary masking) or perform wavelet based denoising for non-stationary noise might also benefit from the long-term SNR estimated by ALTIS.

The non-stationary nature of the multi-talker babble and its spectral similarities with the target speech makes it one of the most challenging noises to separate from a target speaker. The high performance of ALTIS with multi-talker babble suggests that it is likely to work well for other types of noise if ALTIS is retrained for the corresponding noise types. If ALTIS is to be used for estimating the long-term SNR in other types of noise, first, it needs to be trained separately for each type of noise. Then a noise type classifier should be employed to select the appropriate model which is specifically trained for the detected type of noise.

ALTIS extracts features from incoming short-frames of the noisy speech with the duration of 128 ms to estimate the long-term SNR over long-frames consisting of multiple consecutive short-frames. The selected short frame duration (i.e., 128 ms) is due to the fact that the classifier features exhibit suboptimal performance in shorter frames. Even using 50% overlapping short-frames, the chosen duration of 128 ms for short-frames will produce a minimum latency of 64 ms. However, the 64 ms latency of ALTIS will not be added to the latency of a real-time algorithm that uses ALTIS. The real-time algorithm can maintain a latency well below 64 ms while using ALTIS. This is due to the fact that ALTIS and the corresponding real-time algorithm work in parallel and do not need to share the same frame duration. Furthermore, the 64 ms latency inherent in ALTIS is unlikely to affect its usefulness for real-time applications, because ALTIS measures the long-term SNR over a moving long-frame which is substantially longer than this latency. Possible changes of SNR within this 64 ms will not affect the measured SNR over a moving long-frame with a duration of 5.12–48 s. Hence, we can always safely assume that the output of ALTIS shows the long-term SNR at the current time with a negligible error due to the 64 ms delay. It is worth noting that although ALTIS was implemented and evaluated with a 128 ms frame duration, it could easily be trained and implemented with an alternate frame duration. However, for the previously discussed reasons, it is unlikely that it would be beneficial to alter the frame duration.

ALTIS was trained and evaluated with sampling rate of 16,000 samples per second (i.e., frame length of 2048 samples) as this is the standard sampling rate for cochlear implant (CI) devices and this algorithm is intended to be used in a cochlear implant speech enhancement algorithm. All classifier features were optimized for the sampling rate of 16,000 samples per second. If ALTIS is to be used in other sampling rates, all classifier features, DDN classifier and regression models need to be reoptimized and retrained.

ALTIS was evaluated with both known and unknown number of talkers. Knowing the number of talkers did improve the mean absolute error of SNR estimation by 0.06 dB in a dynamic SNR situation. However, the mean absolute error of the realistic condition of when the number of talkers is unknown was only 0.75 dB in a dynamic SNR situation.

Acknowledgments

Support for this research was provided by the National Institutes of Health/National Institute on Deafness and Other Communication Disorders (R01 DC012152; PI: Landsberger) as well as an NYU School of Medicine Applied Research Support Fund internal grant.

References

Bilger, R.C., Nuetzel, J.M., Rabinowitz, W.M., Rzeczkowski, C., 1984. Standardization of a test of speech perception in noise. J. Speech Hear Res. 27, 32–48.

Duda, R.O., Hart, P.E., Stork, D.G., 2001. Pattern Classification, 2nd ed. Wiley, New York.

Elshamy, S., Madhu, N., Tirry, W., Fingscheidt, T., 2017. Instantaneous a priori SNR estimation by cepstral excitation manipulation. IEEE/ACM Trans. Audio Speech Lang. Process. 25, 1592–1605.

Ephraim, Y., Malah, D., 1984. Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator. IEEE Trans. Acoust. Speech Signal Process. 32, 1109–1121.

Ephraim, Y., Malah, D., 1985. Speech enhancement using a minimum mean-square error log-spectral amplitude estimator. IEEE Trans. Acoust. Speech Signal Process. 33, 443–445.

Gu, Q., Li, Z., Han, J., 2012. Generalized fisher score for feature selection. arXiv preprint https://arxiv.org/abs/1202.3725.

Hirsch, H.G., Ehricher, C., 1995. Noise estimation techniques for robust speech recognition. In: Proceedings of the IEEE International Conference of Acoustics Speech Signal Processing, pp. 153–156.

Hu, Y., Loizou, P., 2007. Subjective evaluation and comparison of speech enhancement algorithms. Speech Commun. 49, 588–601.

IEEE Recommended Practice for Speech Quality Measurements, in IEEE No 297-1969, pp. 1−24, June 11, 1969. doi:10.1109/IEEESTD.1969.7405210.

Kim, C., Stern, R.M., 2008. Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis. Interspeech 2598–2601.

Lagarias, J.C., Reeds, J.A., Wright, M.H., Wright, P.E., 1998. Convergence properties of the nelder-mead simplex method in low dimensions. SIAM J. Optim. 9, 112–147.

Lim, J.S., Oppenheim, A.V., 1979. Enhancement and bandwidth compression of noisy speech. Proc. IEEE 67, 1586–1604.

Lun, D.P.K., Hsung, T.C., 2010. Improved wavelet based a-priori SNR estimation for speech enhancement. In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems, pp. 2382–2385.

Martin, R., 2001. Noise power spectral density estimation based on optimal smoothing and minimum statistics. IEEE Trans. Speech Audio Process. 9, 504–512.

Mathews, J.H., Fink, K.D., 2004. Numerical Methods Using Matlab, 4 ed. Pearson.

Moisy F.2016., EzyFit Curve Fitting Toobox for Matlab. [Online]. http://www.fast.u-psud.fr/ezyfit.

Morales-Cordovilla, J., Ma, N., Sanchez, V., Carmona, J., Peinado, A., Barker, J., 2011. A pitch based noise estimation technique for robust speech recognition with missing data. In: Proceedings of the IEEE International Conference of Acoustics Speech Signal Processing, pp. 4808–4811.

Moller, M.F., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural Netw. 6, 525–533.

Narayanan, A, Wang, D., 2012. A CASA-based system for long-term SNR estimation. IEEE Trans. Audio Speech Lang. Process. 20, 2518–2527.

Nelder, J.A., Mead, R., 1965. A simplex method for function minimization. Comput. J. 7, 308–313.

Nemer, E., Goubran, R., Mahmoud, S., 1999. SNR estimation of speech signals using subbands and fourth-order statistics. IEEE Signal Process. Lett. 6, 171–174.

Nilsson, M., Soli, S.D., Sullivan, J.A., 1994. Development of the hearing in noise test for the measurement of speech reception thresholds in quiet and in noise. J. Acoust. Soc. Am. 95, 1085–1099.

Papadopoulos, P., Tsiartas, A., Narayanan, S., 2016. Long-term SNR estimation of speech signals in known and unknown channel conditions. IEEE/ACM Trans. Audio Speech Lang. Process. 24, 2495–2506.

Plapous, C., Marro, C., Scalart, P., 2006. Improved signal-to-noise ratio estimation for speech enhancement. IEEE Trans. Audio Speech Lang. Process. 14, 2098–2108.

Scalart, P., Filho, J.V., 1996. Speech enhancement based on a priori signal to noise estimation. In: Proceedings of the IEEE International Conference of Acoustics Speech Signal Processing, USApp. 629–632.

Sohn, J., Kim, N.S., Sung, W., 1999. A statistical model-based voice activity detection. IEEE Signal Process. Lett. 6 (1), 1–3.

Soleymani, R., Selesnick, I, Landsberger, D., 2018. SEDA: a tunable Q-factor wavelet-based noise reduction algorithm for multi-talker babble. Speech Commun. 96, 102–115.

Sun, H., Ou, S., Liu, R., Gao, Y., 2014. A variable momentum factor algorithm for a priori SNR estimation in speech enhancement. In: Proceedings of the 2014 Seventh International Congress on Image and Signal Processing, pp. 888–892.

Tang, J., Aleyani, S., Liu, H., 2014. Feature selection for classification: a review. Data Classification: Algorithms and Applications. CRC Press.

Tchorz, J., Kollmeier, B., 2003. SNR estimation based on amplitude modulation analysis with applications to noise suppression. IEEE Trans. Speech Audio Process. 11 (3), 184–192.

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P., 2006. The HTK Book (for HTK Version 3.2) Engineering Department. Cambridge University.