INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Global SNR Estimation of Speech Signals for Unknown Noise Conditions using Noise Adapted Non-linear Regression

Pavlos Papadopoulos, Ruchir Travadi, Shrikanth Narayanan

University of Southern California, Signal Analysis and Interpretation Lab, USA
ppapadop@usc.edu, travadi@usc.edu, shri@sipi.usc.edu

Abstract

The performance of speech technologies deteriorates in the presence of noise. Additionally, we need these technologies to be able to operate across a variety of noise levels and conditions. SNR estimation can guide the design and operation of such technologies or can be used as a pre-processing tool in database creation (e.g. identify/discard noisy signals). We propose a new method to estimate the global SNR of a speech signal when prior information about the noise that corrupts the signal, and speech boundaries within the signal, are not available. To achieve this goal, we train a neural network that performs non-linear regression to estimate the SNR. We use energy ratios as features, as well as ivectors to provide information about the noise that corrupts the signal. We compare our method against others in the literature, using the Mean Absolute Error (MAE) metric, and show that our method outperforms them consistently.

Index Terms: signal-to-noise-ratio, i-vectors, neural networks

1. Introduction and Prior Work

Speech applications operating in noisy real life conditions experience performance deterioration, since they are often unable to predict how the type and level of noise will alter the properties of the original speech signal. Signal to Noise Ratio (SNR), a fundamental construct in signal processing, is defined as the ratio of signal power to noise power expressed in decibels (dB) and provides information about the level of noise present in the original signal. Accurate SNR estimation can aid the design of algorithms and systems that compensate for the effects of noise, such as robust automatic speech recognition [1, 2], speech enhancement [3, 4, 5], and noise suppression [6]. However, estimating SNR can be a challenging task, since we do not know how a specific type of noise affects the properties of the original speech signal.

Broadly speaking, SNR estimation falls into two categories: Instantaneous SNR, that focuses on a frame level decision, and Global SNR that deals with the entire signal. In this work, we address the problem of Global SNR estimation which is defined as:

$$\begin{aligned}\text{SNR} =& 20\log_{10}\frac{\sqrt{\frac{1}{M}\sum_{m=1}^{M}s^2[m]}}{\sqrt{\frac{1}{M}\sum_{m=1}^{M}n^2[m]}}\\ =& 10\log_{10}\frac{E(s)}{E(n)}\end{aligned}$$

where $s[\cdot]$, $n[\cdot]$ are the speech and noise signals respectively. Accurate SNR estimation can assist the development of SNR-specific speech and speaker recognition systems [7, 8], as well as other speech processing tasks. Hence, there has been a renewed effort on robust global SNR estimation [9, 10, 11].

Most proposed methods for global SNR estimation typically assume that:

- Background noise is stationary
- Noise and Speech sources are independent
- Noise and Speech are zero-mean signals
- Speech boundaries in the signal are known

However, recent demands of speech technology systems being widely deployed under real-life conditions (e.g., mobile applications with varying environmental conditions) have resulted in many SNR estimation efforts moving away from the stationarity assumption [12, 13]. Furthermore, prior knowledge of speech boundaries in the signal is not always feasible. Speech Activity Detection (SAD) systems could be used to extract speech regions, but they are usually tailored to handle specific channel-conditions. The authors in [10] study such effects of SAD on SNR estimation.

Another approach is the NIST SNR measurement [14], which models the noise using a sequential Gaussian mixture estimation approach. Then, it builds a short-time energy histogram to estimate the signal and noise energy distributions, and makes a decision based on those distributions.

In [9], the authors assume that the amplitude of the speech and noise signals follow Gamma and Gaussian distributions, respectively. They claim that different levels of noise affect the shaping parameter of the Gamma distribution and perform Maximum Likelihood (ML) estimation to estimate that parameter, which determines their SNR estimation. Their system works well when their assumptions are met, but fails when noise has impulsive characteristics [18].

Other strategies include estimation of the Ideal Binary Mask (IBM)[15], which identifies speech and noise regions under a time-frequency representation. In [16] a system is presented that estimates the SNR using a binary mask on the voiced speech frames. Their estimates are accurate when SNR is close to 0dB but biased under other conditions. This problem is rectified in [11] where the authors propose a method based on computational auditory scene analysis, where IBM is estimated in both voiced and unvoiced regions.

In [17] the authors developed a procedure to estimate the SNR in unknown noise conditions based on Mel-frequency cepstral coefficients (MFCCs) and the K-Nearest Neighbour algorithm (KNN). Although this technique performed well when the unknown noise had similar characteristics with at least one of those used to train the KNN, it suffered from poor generalization properties due to the sensitivity of MFCCs to noise. These shortcomings are alleviated in [18] where the authors use a different feature set, and build noise-specific regression models to estimate the SNR. Moreover, they train a neural network that acts as a noise type classifier and is used to select the appropriate noise type from a noise bank, when dealing with unknown noise scenarios.

http://dx.doi.org/10.21437/Interspeech.2017-230

In this work, we propose a new method to estimate SNR which is not dependent on specific noise conditions. To that end, we perform a non-linear regression (to allow for more flexibility in the estimation procedure) by employing a neural network, that accepts a feature set based on energy ratios. These features are able to capture the proper information in the signal for accurate SNR estimations under known noise conditions [18]. However, training such a network for multiple noise conditions is a challenging task, since the input features are dependent on noise and this information is not represented in the features. We use ivectors [19] to perform channel adaptation on neural networks, inspired by previous work on speaker adaptation [20]. Since ivectors contain both speaker and channel information, we follow a similar approach to adapt the network to specific channel conditions, by appending ivectors to our original feature set. Furthermore, we make no assumptions regarding speech boundaries in the signal.

The rest of this paper is organized as follows. In Section 2 we give a brief overview of the Total Variability Model and ivector extraction. In Section 3 we present the features and give details about the network we built to estimate the SNR. In Section 4 we describe our dataset and present our results. Finally, in Section 5 we draw our conclusions.

2. Total Variability Model

The Total Variability Model (TVM) [19] is a popular framework for obtaining a fixed-dimensional vector-space representation, also known as an ivector, in order to capture differences in feature space distributions across variable length sequences.

2.1. Motivation

The Total Variability Model is more commonly used in applications such as speaker recognition [19] and language identification [21, 22], where ivectors are used to capture speaker or language variability across utterances, respectively. However, in applications where such variability is undesirable, the ivector representation can also be used as an appended input to a discriminative system, in order to enable it to adapt to the source of variability represented by the ivector. For example, appending ivectors to the input while training an acoustic model for speech recognition has been found to improve speaker independence of the recognition system [20, 23]. Similarly, in our case, the motivation for using ivectors is to enable the SNR estimation system to be able to predict the SNR robustly, while staying independent of the variable noise conditions present in the utterance.

2.2. Model Formulation

Let $\mathbf{X} = \{\mathbf{X}_u\}_{u=1}^{U}$ be the collection of acoustic feature vectors in a dataset comprising of U utterances, where $\mathbf{X}_u = \{\boldsymbol{x}_{ut}\}_{t=1}^{T_u}$ denotes the feature vector sequence of length T_u from a specific utterance u. Let D be the dimensionality of each feature vector: $\boldsymbol{x}_{ut} \in \mathbb{R}^D$. In the Total Variability Model (TVM), it is assumed that with every utterance u, there is an associated vector $\boldsymbol{w}_u \in \mathbb{R}^K$ (K being a design parameter) known as the *ivector* for that utterance. The conditional distribution of $\boldsymbol{x}_{ut}$ given $\boldsymbol{w}_u$ is a Gaussian Mixture Model of C components with parameters $\{p_c, \boldsymbol{\mu}_{uc} = \boldsymbol{\mu}_c + \mathbf{T}_c\boldsymbol{w}_u, \boldsymbol{\Sigma}_c\}_{c=1}^{C}$ where $p_c \in \mathbb{R}, \boldsymbol{\mu}_c \in \mathbb{R}^D, \mathbf{T}_c \in \mathbb{R}^{D\times K}$ and $\boldsymbol{\Sigma}_\mathbf{c} \in \mathbb{R}^{D\times D}$. The prior distribution for $\boldsymbol{w}_u$ is assumed to be standard normal:

$$f(\boldsymbol{w}_u) = \mathcal{N}(\mathbf{0}, \mathbf{I})$$

Let $\boldsymbol{M}_0, \boldsymbol{M}_u \in \mathbb{R}^{CD}$ denote vectors consisting of stacked global and utterance-specific component means $\boldsymbol{\mu}_c$ and $\boldsymbol{\mu}_{uc}$ respectively. Then, the TVM can be summarized as:

$$\boldsymbol{M}_u = \boldsymbol{M}_0 + \mathbf{T}\boldsymbol{w}_u$$

where $\mathbf{T} \in \mathbb{R}^{CD\times K}$ is given as: $\mathbf{T} = \left[\begin{array}{c:c:c} \mathbf{T}_1^\mathsf{T} & \dots & \mathbf{T}_C^\mathsf{T} \end{array}\right]^\mathsf{T}$

2.3. Parameter Estimation and Ivector Extraction

TVM parameters are usually estimated using the Expectation Maximization algorithm [19]. However, we chose to estimate them using randomized Singular Value Decomposition (SVD) [24] since it is much faster compared to EM. For extracting the ivector for an utterance, we first obtain statistics $\boldsymbol{N}_u, \boldsymbol{F}_u$:

$$N_{uc} = \sum_{t=1}^{T_u} \gamma_{utc} \qquad \boldsymbol{N}_u = [\, N_{u1} \;\; \dots \;\; N_{uC} \,]$$

$$\boldsymbol{F}_{uc} = \frac{1}{N_{uc}} \sum_{t=1}^{T_u} \gamma_{utc}\, \boldsymbol{x}_{ut} \qquad \boldsymbol{F}_u = \left[\begin{array}{c:c:c} \boldsymbol{F}_{u1}^\mathsf{T} & \dots & \boldsymbol{F}_{uC}^\mathsf{T} \end{array}\right]^\mathsf{T}$$

where γ_{utc} are component posteriors obtained from the Universal Background Model (UBM). Let $\boldsymbol{\Sigma}_c^{-1} = \boldsymbol{L}_c\boldsymbol{L}_c^\mathsf{T}$ be the Cholesky decomposition of $\boldsymbol{\Sigma}_c^{-1}$. Then, the statistics are normalized as:

$$\widetilde{\boldsymbol{F}}_{uc} = \sqrt{N}_{uc}\boldsymbol{L}_c^\mathsf{T}\boldsymbol{F}_{uc} \qquad \widetilde{\boldsymbol{F}}_u = \left[\begin{array}{c:c:c} \widetilde{\boldsymbol{F}}_{u1}^\mathsf{T} & \dots & \widetilde{\boldsymbol{F}}_{uC}^\mathsf{T} \end{array}\right]^\mathsf{T}$$

Then, as in [24], the ivector for an utterance is extracted from its normalized statistics as below:

$$\boldsymbol{w}_u^* = \frac{1}{\sqrt{T_u}} \left(\frac{1}{T_u}\boldsymbol{I} + \widetilde{\boldsymbol{T}}^\mathsf{T}\widetilde{\boldsymbol{T}}\right)^{-1} \widetilde{\boldsymbol{T}}^\mathsf{T}\widetilde{\boldsymbol{F}}_u$$

where $\widetilde{\boldsymbol{T}}$ is a normalized version of the matrix $\boldsymbol{T}$ estimated during the randomized SVD algorithm.

3. Feature and Network Description

In this section we describe the features and the neural network architecture we used to estimate SNR.

3.1. Features

The feature set we use is similar to the one described in [18] and is comprised of energy ratios, calculated based on different feature sets, Long-Term Energy (LTE), Long-term Signal Variability (LTSV) [25], Pitch, and Voicing Probability. In the case of LTE calculating energy ratios is straightforward. Given a signal y its LTE is defined as:

$$\mathcal{E}_y(n) = \frac{1}{|F|} \sum_{f_j \in F} S_y(n, f_j)$$

where $S_y(n, f_j)$ is the spectrum at frame n and frequency bin f_j, F is the set of frequency bins, and $|F|$ is the cardinality of F. In our experiments, the spectrum is calculated using a 25ms hamming window with a 10ms shift and 256 frequency bins. Then, we apply a simple moving average window of length m on the long-term energy to eliminate abrupt transitions. Thus, we acquire a smoothed version of LTE $E(y) = \mathbb{S}_m(\mathcal{E}_y)$ with $\mathbb{S}_m(\cdot)$ being the smoothing operator. We try smoothing windows of different lengths in an effort to maintain the original information in the signal, but also get robust measurements. For every window length we compute an energy measurement, $E(x)$, in the following manner: First, we calculate

the smoothed long-term energy in each time frame and sort the values, then we pick two percentile values (e.g. 90% and 95%) that correspond to percentage values of the total energy, and calculate the average LTE of the frames that fall in that region. The reason we chose percentile values is that signals can be of arbitrary length and speech boundaries are unknown. We repeat the same procedure for two different percentile values (e.g. 10% and 15%). Finally, for every triplet of smoothing window length, and energy measurements we construct an energy ratio as:

$$l_m^{a-b,c-d} = 10\log_{10}\frac{E^{c-d}(x) - E^{a-b}(x)}{E^{a-b}(x)} \quad (1)$$

where m stands for the length of the smoothing window, a, b, c, d are the percentile values, and $E^{c-d}(x)$, $E^{a-b}(x)$ are energy measurements based on their respective percentile values. The smoothing window length ranges from 5 to 30 frames with a step of 5.

The second feature we use to create energy ratios is LTSV [25], which is a method of determining the degree of non-stationarity in a signal by measuring the entropy of the normalized short-time spectrum at every frequency over consecutive frames. To construct these energy ratios we take the following steps. First, we apply a moving average smoothing window on both the LTSV values and the LTE of the signal (to smooth regions of abrupt transitions) and then we sort the smoothed LTSV. We choose a window defined by two percentile values (e.g. 90% and 95%) of the highest LTSV values and find all the frames that fall in that range. Finally, we compute the energy of the LTSV frames of that window and take a measurement of $E(x)$ based on those. Using the same method, we compute the energy in a different window of LTSV values. Hence, we create energy ratios as:

$$v_{m,k,R}^{a-b,c-d} = 10\log_{10}\frac{E(\xi^{c-d}(x)) - E(\xi^{a-b}(x))}{E(\xi^{a-b}(x))} \quad (2)$$

where m, k, are the lengths of the smoothing windows for the energy and LTSV respectively, the set a, b, c, d are the percentile values that define the windows and R is the analysis window of LTSV. The expression $E(\xi^{c-d}(x))$ stands for the energy measurement through the LTSV feature. In other words, it is the estimation of energy based on the frames that fall into the $c\%$-$d\%$ region of the sorted LTSV. We tried three different lengths of windows for energy smoothing (parameter m): 10, 20, and 30 frames. Different sets of LTSV features were extracted using three different analysis windows (parameter R), i.e., 10, 15, and 20 frames. We additionally processed every LTSV set by applying a smoothing window. We tried six different window lengths (parameter k), from 5 to 30 with a step of 5 frames.

In a similar fashion, we calculate energy ratios based on pitch and voicing probability. The ratios based on pitch can be expressed as:

$$p_{m,k}^{a-b,c-d} = 10\log_{10}\frac{E(f^{c-d}(x)) - E(f^{a-b}(x))}{E(f^{a-b}(x))} \quad (3)$$

while those based on voicing probability can be calculated as:

$$c_{m,k}^{a-b,c-d} = 10\log_{10}\frac{E(g^{c-d}(x)) - E(g^{a-b}(x))}{E(g^{a-b}(x))} \quad (4)$$

where in both cases m, k, are the lengths of the smoothing windows for the energy and pitch/voicing respectively. For energy smoothing we used a moving average window of 10 frames (parameter m), while for pitch and voicing we tried six different window lengths(parameter k) ranging from 5 frames to 30 frames with a step of 5.

For all the feature sets, the percentile values a, b, c, d that define the measurement windows are presented in Table 1. Through this parametrization we created 312 energy ratios. Finally, once we extract these ratios for every utterance, we extract the ivector for that utterance, as described in Section 2. We extract 400-dimensional ivectors using a 512 Gaussians Component UBM trained on 13-dimensional MFCCs. Finally, we append the ivectors to the energy ratios. Thus, the input stream to the neural network has a dimensionality of 712.

Table 1: *Percentile values that define measurement windows in equations (1)-(4)*

	a	**b**	**c**	**d**
Long Term Energy and Long Term Signal Variability	85%	95%	5%	15%
	80%	90%	10%	20%
	5%	15%	85%	95%
	10%	20%	80%	90%
Pitch and Voicing Probability	85%	95%	5%	15%
	80%	90%	10%	20%
	75%	85%	15%	25%
	5%	15%	85%	95%
	10%	20%	80%	90%
	15%	25%	75%	85%

3.2. Neural Network for SNR Estimation

In order to estimate the SNR of noisy signals we implemented a feed-forward neural in TensorFlow [26]. The network consists of 4 hidden layers. Every layer has 1024 neurons with RELU (rectified linear unit) activations [27]. A RELU activation is defined as:

$$f(y) = \max(y, 0)$$

A major benefit of RELU activations over sigmoid is the constant value of the gradient, which occurs when $y = Wx + b$ is greater than 0. In contrast, the sigmoid gradient goes to 0 as the absolute value of x increases, which results in the vanishing gradient problem.

Usually, the Mean Square Error (MSE) cost function is used in regression settings. However, we opted for the Mean Absolute Error (MAE) cost function, since we achieved better SNR estimates in our experiments without affecting training time. The Mean Absolute Error (MAE) cost function is defined as:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|$$

where N is the number of data points, and $\hat{y}_i$, y_i are the estimated SNR value and ground truth respectively, for data point i. Parameter optimization was performed using the Adam (Adaptive Moment) optimizer [28] with $l = 10^{-5}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$, where l is the learning rate and β_1, β_2 are hyperparameters controlling the exponential decay rates of the moving averages of the gradient and the squared gradient. Gradient descent operated on mini-batches of 128 utterances for 20

epochs. Moreover, we utilized two GPUs (Graphics Processing Unit) to train the network following a synchronous synchronization strategy [29] through gradient averaging. Finally, the input layer had a dimensionality of 712 (our feature dimensionality), while the output layer of the network was a linear layer producing the SNR estimate.

4. Experimental Results

To test the validity of our approach we created a "noisy" speech dataset by combining clean speech utterances from TIMIT and noises from the DEMAND database [30]. The DEMAND database contains 18 different noises, each having a duration of 5 minutes, drawn from urban environments. We used 2000 clean speech utterances, and for each utterance we added one of the 18 different types of noise at 9 different SNR levels, from -5dB to 15dB with a step of 2.5. Moreover, in each of these utterances we added silence periods randomly selected to be between 2 and 4 seconds to create signals with unknown speech boundaries. Thus, for each of the 18 noise types we have 18000 noisy utterances of both male and female speakers (2000 utterances $\times$ 9 SNR levels), resulting in 324000 noisy utterances, subsets of which will serve as out training set. The test set is created in a similar fashion, using 100 clean speech utterances (50 male and 50 female) from TIMIT, ensuring that there is no overlap in the training and testing utterances. Hence our test set consists of 900 utterances per noise type.

In our first set of experiments we check the predictive capability of our features independently. To that end, we performed three experiments. In the first, we used just the energy ratios to train the network that provides SNR estimates, in the second we used only the ivectors, and in the last, their combination. In all of these experiments we used the complete training and testing sets. The results are summarized in Table 2. Obviously this set of experiments does not refer to the unknown noise scenario, however, we can draw some useful conclusions. We observe that both energy ratios and ivectors (to a lesser extend) contain valuable information for SNR estimation. However, the combination of the ratios with the ivectors provides the lowest MAE, since ivectors capture the channel conditions, enabling the network to fine tune the SNR estimates for different types of noise.

Table 2: *SNR Mean Absolute Error for different feature sets.*

	ratios	ivectors	combination
MAE	2.807	3.190	1.546

Next, we test our method under unknown noise conditions. To achieve this goal, we follow a leave-one-out strategy. We exclude all the files in the training set that have been corrupted by a particular type of noise (e.g. Park noise). We train a UBM with the remaining set of 306000 noisy utterances, and extract ivectors. The UBM is built using 512 Gaussian components, while the extracted ivectors have a dimensionality of 400. Then, we extract the energy ratio features, combine them with the ivectors, and train the network. Our test set consists of utterances corrupted by the type of noise that was excluded from the training set (e.g. Park noise). Using this approach, we ensure that our model will operate on utterances altered by some noise for which we have no prior information. Finally, we repeat this procedure for different types of noise.

We compare our method (Channel Adapted DNN) with WADA (Waveform Amplitude Distribution Analysis) [9] and DNN selection [18] for 8 different types of noise[1]. For each noise type we calculate the average MAE across different SNR levels and present the results in Table 3.

Table 3: *SNR Mean Absolute Error across different estimation methods and averages for 9 different SNR levels.*

	WADA	**DNN Selection**	**Ch. Ad. DNN**
KITCHEN	4.663	2.976	**2.835**
LIV. ROOM	3.641	2.413	**1.358**
METRO	7.126	4.761	**2.902**
PARK	5.644	3.356	**2.116**
STATION	3.121	1.732	**1.141**
TRAFFIC	4.567	3.599	**1.936**
RESTAURANT	3.345	2.454	**1.918**
CAFE	3.766	2.691	**1.106**

We observe that our method consistently outperforms the other approaches and achieves low MAE for all the noise types we tested against. Furthermore, our results indicate that the ivectors hold information regarding the type of noise that is altering the signal, something that other applications (e.g., speech enhancement) can take advantage of. Our method is dependent on the size of the noise pool at our disposal, since we exploit similarities between noise conditions. For example, if the UBM and the network were trained with instances drawn from just one noise type, we believe that performance would drop significantly. However, our method achieves low MAE for many challenging noise conditions encountered in real life.

5. Conclusions and Future Research

We proposed a method for estimating the global SNR operating on signals with unknown speech boundaries, that is able to generalize across different noise conditions. We compared our method against the state-of-the-art in the literature, and found that our performance was consistently better. Our method can be considered as "noise-independent" since it does not make explicit assumptions about the type of noise that alters the original speech signal, instead it uses ivectors to model the noise type and adapt the final SNR estimation. This "noise-independence" property is the reason of the enhanced performance of our method, since we do not force our system to deal with specific family of noises. However, our method depends on the availability of noises. Training the UBM and the network from a small noise pool will result in a model that is not able to generalize across different noise conditions. Moreover, the noise pool must contain diverse noise types. If the noise pool only contains examples of stationary types of noise, our model will not be able to handle noises with impulsive characteristics. To overcome this shortcomings, we plan to explore if different models can capture the noise type without relying on features to provide that information (e.g., recurrent networks).

[1]We also compared against NIST SNR, but it failed to provide reasonable SNR estimates (e.g. MAE was over 30dB in most cases), thus we opted not to include it in our comparison.

6. References

[1] H. G. Hirsch and C. Ehricher, "Noise Estimation Techniques for Robust Speech Recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1995.

[2] J. Morales-Cordovilla, N. Ma, V. Sanchez, J. Carmona, A. Peinado, and J. Barker, "A Pitch Based Noise Estimation Technique for Robust Speech Recognition with Missing Data," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 4808–4811.

[3] Y. Ephraim and D. Malah, "Speech Enhancement Using a Minimum-Mean Square Error Short-Time Spectral Amplitude Estimator," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[4] C. Plapous, C. Marro, and P. Scalart, "Improved Signal to Noise Ratio Estimation for Speech Enhancement," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 6, pp. 2098–2108, 2006.

[5] Y. Ren and M. T. Johnson, "An Improved SNR Estimator for Speech Enhancement," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2008, pp. 4901–4904.

[6] J. Tchorz and B. Kollmeier, "SNR Estimation Based on Amplitude Modulation Analysis with Applications to Noise Suppression," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 3, pp. 184–192, 2003.

[7] S. Furui, *Digital Speech Processing, Synthesis, and Recognition*, ser. Signal processing and communications. Marcel Dekker, 2001.

[8] X. Zhao, Y. Shao, and D. Wang, "Robust Speaker Identification Using a CASA front-end," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 5468–5471.

[9] C. Kim and R. M. Stern, "Robust Signal-to-Noise Ratio Estimation Based on Waveform Amplitude Distribution Analysis," in *Proc. Interspeech*, 2008, pp. 2598–2601.

[10] M. Vondrášek and P. Pollák, "Methods for Speech SNR Estimation: Evaluation Tool and Analysis of VAD Dependency," *Radioengineering*, vol. 14, pp. 6–11, 2005.

[11] A. Narayanan and D. Wang, "A CASA-Based System for Long-Term SNR Estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2518–2527, Nov 2012.

[12] R. Hendriks, R. Heusdens, and J. Jensen, "MMSE based noise PSD tracking with low complexity," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, March 2010, pp. 4266–4269.

[13] P. C. Loizou, *Speech Enhancement: Theory and Practice* . CRC, 2007.

[14] The NIST Speech SNR Measurement. [Online]. Available: http://www.nist.gov/smartspace/nist_speech_snr_measurement.html

[15] D. Wang, "On Ideal Binary Mask As the Computational Goal of Auditory Scene Analysis," in *Speech Separation by Humans and Machines*, P. Divenyi, Ed. Springer US, 2005, pp. 181–197.

[16] G. Hu and D. Wang, "Segregation of Unvoiced Speech from Nonspeech Interference," *The Journal of the Acoustical Society of America*, vol. 124, no. 2, pp. 1306–1319, 2008.

[17] P. Papadopoulos, A. Tsiartas, J. Gibson, and S. Narayanan, "A Supervised Signal-to-Noise Ratio Estimation of Speech Signals," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 8237–8241.

[18] P. Papadopoulos, A. Tsiartas, and S. Narayanan, "Long-term SNR Estimation of Speech Signals in Known and Unknown Channel Conditions," IEEE Transactions on Audio, Speech and Language Processing, (In Press).

[19] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis for Speaker Verification," *IEEE Trans. Audio, Speech & Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[20] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 55–59.

[21] D. M. Gonzlez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, "Language Recognition in iVectors Space," in *INTERSPEECH*, 2011, pp. 861–864.

[22] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language Recognition via i-vectors and Dimensionality Reduction," in *INTERSPEECH*, 2011, pp. 857–860.

[23] A. W. Senior and I. Lopez-Moreno, "Improving DNN speaker independence with I-vector inputs," in *ICASSP*, 2014, pp. 225–229.

[24] R. Travadi and S. Narayanan, "Non-Iterative Parameter Estimation for Total Variability Model Using Randomized Singular Value Decomposition," in *INTERSPEECH*, 2016, pp. 3221–3225.

[25] P. Ghosh, A. Tsiartas, and S. Narayanan, "Robust Voice Activity Detection Using Long-Term Signal Variability," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 600–613, 2011.

[26] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

[27] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[28] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference for Learning Representations (ICLR)*, 2015.

[29] J. Chen, R. Monga, S. Bengio, and R. Jozefowicz, "Revisiting distributed synchronous sgd," in *International Conference on Learning Representations Workshop Track*, 2016.

[30] J. Thiemann, N. Ito, and E. Vincent, "The Diverse Environments Multi-channel Acoustic Noise Database (DEMAND): A database of multichannel environmental noise recordings," in *21st International Congress on Acoustics*. Acoustical Society of America, 2013.