Convolutional Bias Removal Based on Normalizing the Filterbank Spectral Magnitude

Zekeriya Tufekci, *Member, IEEE*

***Abstract*—In this letter, a novel convolutional bias removal technique is proposed. The proposed method is based on scaling the filterbank magnitude by the average of filterbank magnitude over time. The relation between the cepstral mean normalization (CMN) and proposed algorithm is derived. The experimental results show that the proposed algorithm is more robust than the CMN for both convolutional bias and additive noise. For example, the proposed method reduced the equal error rate by 5.66% and 10.16% on average for the convolutional bias and 12-dB additive noise, respectively.**

***Index Terms*—Additive noise, convolutional noise, robust speaker verification.**

I. Introduction

REAL-WORLD applications require that speech recognition or speaker verification systems be robust to interfering noise. The performance of a speech recognition or speaker verification system drops dramatically when there is a mismatch between training and testing conditions. Many different approaches have been studied to decrease the effect of noise on the performance [1]. The main focus of this letter is on reducing the effect of stationary convolutional noise on the speaker verification performance.

Convolutional noise distortions are mostly caused by variable frequency characteristic of different communication channels, the use of different microphones, and the use of different handsets for telephony speech. Cepstral mean normalization (CMN) [2] is a simple but very efficient method to improve the robustness of mel-frequency cepstral coefficients (MFCCs) [3] to stationary convolutional noise. The CMN also reduces the effect of additive background noise. A new method, namely magnitude spectrum normalization (MSN), which is based on scaling the filterbank spectral magnitude by the average filterbank spectral magnitude, is proposed for convolutional bias removal.

Research on speaker verification [4] has been an active area for decades. The goal of speaker verification system is to determine from a voice of sample if a person is whom he or she claims. The speech can be constrained to be a known phrase (text-dependent) or totally unconstrained (text-independent). This study is concerned with the text-independent speaker verification. The GMMs [5] recently have become dominant approach in text-independent speaker verification. In this letter, speakers were modeled using Gaussian mixture models (GMMs).

Manuscript received June 20, 2006; revised November 5, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Brian Kan-Wing Mak.

The author is with the Izmir Yuksek Teknoloji Enstitusu, Elektrik-Elektronik Muhendisligi Bolumu, 35430 Izmir, Turkey (e-mail: zekeriyatufekci@iyte.edu.tr).

Digital Object Identifier 10.1109/LSP.2006.891313

II. Cepstral Mean Normalization and the Proposed Algorithm

Consider the following speech signal corrupted with stationary convolutional noise:

$$x[n] = s[n] * h[n] \tag{1}$$

where n is the time index, $s[n]$ is the clean speech sequence, $h[n]$ is the impulse response of stationary convolutional noise, and $x[n]$ is the corrupted speech signal. After applying the discrete Fourier transform (DFT) to a frame of speech, we get

$$X_m[k] = S_m[k]H[k] \tag{2}$$

where m is the frame index and k is the DFT index. X_m, S_m, and H are DFT magnitudes for the noisy speech, clean speech and convolutional noise, respectively. The filterbank spectral magnitude is given by

$$Y_m[i] = \sum_k W_i[k]X_m[k] = \sum_k W_i[k]S_m[k]H[k] \tag{3}$$

where $Y_m[i]$ is the i^{th} filterbank spectral magnitude for the m^{th} frame and W_i is the filterbank weight vector for the i^{th} filter. If we assume that $H[k]$ is constant within the frequency band of filter i, we can express the filterbank magnitude as

$$Y_m[i] = \sum_k W_i[k]S_m[k]H[k] \approx H_i \sum_k W_i[k]S_m[k] \tag{4}$$

where H_i is the convolutional bias for the i^{th} filterbank. For the rest of the letter, superscripts will be used to denote the domain of the observation, thus $\boldsymbol{Y_m^c}$ is the noisy speech vector in the cepstral domain for the m^{th} frame and $\boldsymbol{Y_m^l}$ is the noisy speech vector in the log-filterbank magnitude domain for the m^{th} frame. Absence of a superscript indicates the filterbank magnitude domain, e.g, $\boldsymbol{Y_m}$ represents the noisy speech vector in the filterbank magnitude domain for the m^{th} frame. All variables in bold are matrices or vectors, and variables in square brackets indicate elements of the vectors or matrices.

A. Cepstral Mean Normalization

Consider the sequence of MFCCs vectors $\{\boldsymbol{Y_0^c}, \boldsymbol{Y_1^c}, \ldots \boldsymbol{Y_m^c}, \ldots \boldsymbol{Y_{N-1}^c}\}$, where $\boldsymbol{Y_m^c}$ represents the MFCCs vector for the m^{th} frame. Its sample mean is

$$\bar{\boldsymbol{Y}}^c = \frac{1}{N}\sum_{k=0}^{N-1} \boldsymbol{Y_k^c}. \tag{5}$$

The CMN consists of subtracting $\bar{\boldsymbol{Y}}^{\boldsymbol{c}}$ from each cepstral vector $\boldsymbol{Y_m^c}$ to obtain normalized cepstral vector $\hat{\boldsymbol{Y}}_{\boldsymbol{m}}^{\boldsymbol{c}}$

$$\hat{\boldsymbol{Y}}_{\boldsymbol{m}}^{\boldsymbol{c}} = \boldsymbol{Y_m^c} - \bar{\boldsymbol{Y}}^{\boldsymbol{c}}. \tag{6}$$

In the log-filterbank magnitude, domain (6) can be written as (if all cepstral coefficients are used)

$$\hat{\boldsymbol{Y}}_{\boldsymbol{m}}^{\boldsymbol{l}} = \boldsymbol{C}^{-\mathbf{1}}(\boldsymbol{Y}_{\boldsymbol{m}}^{\boldsymbol{c}} - \bar{\boldsymbol{Y}}^{\boldsymbol{C}}) = \boldsymbol{Y}_{\boldsymbol{m}}^{\boldsymbol{l}} - \bar{\boldsymbol{Y}}^{\boldsymbol{l}} = \log\left(\frac{\boldsymbol{Y}_{\boldsymbol{m}}}{\left(\prod_{k=0}^{N-1}\boldsymbol{Y}_{\boldsymbol{k}}\right)^{1/N}}\right) \tag{7}$$

where $\boldsymbol{C}^{-\mathbf{1}}$ is the inverse cosine matrix. Therefore, the CMN is equal to scaling the filterbank magnitude by the geometric mean of filterbank magnitude over N frames. If we do not use all cepstral coefficients, then

$$\hat{\boldsymbol{Y}}_{\boldsymbol{m}}^{\boldsymbol{l}} \approx \log\left(\frac{\boldsymbol{Y}_{\boldsymbol{m}}}{\left(\prod_{k=0}^{N-1}\boldsymbol{Y}_{\boldsymbol{k}}\right)^{1/N}}\right). \tag{8}$$

B. The Proposed Algorithm

Instead of subtracting the cepstral mean from the cepstrum, the logarithm of the average filterbank magnitude over a length of N frames is subtracted from the log-filterbank magnitude as defined below. The sample mean of the filterbank magnitude is given by

$$\bar{\boldsymbol{Y}} = \frac{1}{N}\sum_{k=0}^{N-1}\boldsymbol{Y}_{\boldsymbol{k}}. \tag{9}$$

The MSN consists of subtracting logarithm of $\bar{\boldsymbol{Y}}$ from each log-filterbank vector $\boldsymbol{Y}_{\boldsymbol{m}}^{\boldsymbol{l}}$ to obtain normalized log-filterbank vector $\hat{\boldsymbol{Y}}_{\boldsymbol{m}}^{\boldsymbol{l}}$

$$\hat{\boldsymbol{Y}}_{\boldsymbol{m}}^{\boldsymbol{l}} = \boldsymbol{Y}_{\boldsymbol{m}}^{\boldsymbol{l}} - \log\bar{\boldsymbol{Y}} = \log\left(\frac{\boldsymbol{Y}_{\boldsymbol{m}}}{\frac{1}{N}\sum_{k=0}^{N-1}\boldsymbol{Y}_{\boldsymbol{k}}}\right). \tag{10}$$

The only difference between the CMN and the proposed algorithm is on scaling the filterbank spectral magnitude. The filterbank spectral magnitude is normalized by arithmetic mean for the proposed algorithm and by geometric mean for the CMN as seen from (8) and (10). It can be easily shown that both method are able to remove the convolutional bias. It was shown that

$$Y_m[i] \approx H_i \sum_k W_i[k]S_m[k] = H_i Z_m[i] \tag{11}$$

where $Z_m[i]$ is the i^{th} filterbank spectral magnitude for the clean speech. Then for the CMN

$$\hat{Y}_m^l[i] \approx \log\left(\frac{H_i Z_m[i]}{\left(\prod_{k=0}^{N-1} Z_k[i]H_i\right)^{1/N}}\right) = \log\left(\frac{Z_m[i]}{\left(\prod_{k=0}^{N-1} Z_k[i]\right)^{1/N}}\right). \tag{12}$$

Therefore, the CMN removes the stationary convolutional noise but the filterbank magnitude will be scaled by the geometric mean of the filterbank magnitude over N frames. For the proposed algorithm

$$\hat{Y}_m[i]^l = \log\left(\frac{Z_m[i]H_i}{\frac{1}{N}\sum_{k=0}^{N-1} Z_k[i]H_i}\right) = \log\left(\frac{Z_m[i]}{\frac{1}{N}\sum_{k=0}^{N-1} Z_k[i]}\right). \tag{13}$$

So the MSN will remove the convolutional bias but the resulting filterbank magnitude will be scaled by the arithmetic mean of the filterbank magnitude over N frames. As a result, both methods remove the convolutional bias but the resulting filterbank spectral magnitude will be scaled by the geometric mean for the CMN and arithmetic mean for the proposed algorithm. In this letter, the past N frames of speech signal were used to compute the normalized i^{th} cepstral coefficients for both CMN and MSN to make the systems causal.

C. The Effect of Additive and Convolutional Noises on CMN and MSN-Based MFCCs

It is well known [6] that the arithmetic mean is always greater than or equal to the geometric mean

$$\bar{\boldsymbol{Y}} = \frac{1}{N}\sum_{k=0}^{N-1}\boldsymbol{Y}_{\boldsymbol{k}} \geq \left(\prod_{k=0}^{N-1}\boldsymbol{Y}_{\boldsymbol{k}}\right)^{1/N}. \tag{14}$$

The relation between the expected values of the arithmetic and geometric means can be computed if the distributions of Y_k are given. If we assume that the MFCCs are Gaussian distributed random vectors, then the distributions of the MFCCs in the log magnitude domain will be lognormal. The mean and covariance matrix of the MFCCs in the log filterbank magnitude domain can be computed as

$$\boldsymbol{\mu}^{\mathbf{l}} = \mathbf{C}^{-\mathbf{1}}\mu^{\mathbf{c}} \tag{15}$$

$$\boldsymbol{\Sigma}^{\mathbf{l}} = \mathbf{C}^{-\mathbf{1}}\boldsymbol{\Sigma}^{\mathbf{c}}(\mathbf{C}^{-\mathbf{1}})^{\mathbf{T}} \tag{16}$$

where μ is the mean, Σ is the covariance matrix, C^{-1} is the inverse cosine matrix, and $(.)^T$ represents the transpose of a matrix.

If we assume that Y_k's are independent and identically distributed (i.i.d.) random vectors, then we can easily compute the expected values of the arithmetic and geometric means. The expected value of the arithmetic mean of Y_m can be computed as

$$\boldsymbol{E}\left\{\frac{1}{\boldsymbol{N}}\sum_{\boldsymbol{k=0}}^{\boldsymbol{N-1}}\boldsymbol{Y}_{\boldsymbol{k}}\right\} = \boldsymbol{E}\left\{\boldsymbol{Y}_{\mathbf{0}}\right\} = \boldsymbol{E}\left\{\boldsymbol{e}^{\boldsymbol{Y}_{\mathbf{0}}^{\boldsymbol{l}}}\right\} = \boldsymbol{\mu} \tag{17}$$

where $\boldsymbol{\mu}$ is the mean of $\boldsymbol{Y}_{\boldsymbol{k}}$. The expected value of the arithmetic mean for the i^{th} filterbank can be computed as

$$\boldsymbol{E}\left\{\frac{1}{N}\sum_{k=0}^{N-1}\boldsymbol{Y}_{\boldsymbol{k}}[\boldsymbol{i}]\right\} = \boldsymbol{E}\left\{e^{Y_0^l[i]}\right\} = \mu[i] = e^{\mu^l[i]+\Sigma^l[i,i]/2} \tag{18}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of $\boldsymbol{Y}_{\boldsymbol{k}}$. The expected value of the geometric mean of Y_m can be computed as

$$\boldsymbol{E}\left\{\left(\prod_{k=0}^{N-1}\boldsymbol{Y}_{\boldsymbol{k}}\right)^{1/N}\right\} = \boldsymbol{E}\left\{e^{1/N\sum_{k=0}^{N-1}\boldsymbol{Y}_{\boldsymbol{k}}^{\boldsymbol{l}}}\right\} = \left(\boldsymbol{E}\left\{\boldsymbol{e}^{(\mathbf{1}/\boldsymbol{N})\boldsymbol{Y}_{\mathbf{0}}^{\boldsymbol{l}}}\right\}\right)^{N}. \tag{19}$$

The expected value of the geometric mean for the i^{th} filterbank can be computed as

$$\left(E\left\{e^{1/NY_0^l[i]}\right\}\right)^N = \left(e^{\mu^l[i]/N+\Sigma^l[i,i]/(2N^2)}\right)^N = e^{\mu^l[i]+\Sigma^l[i,i]/(2N)} \approx e^{\mu^l[i]}. \quad (20)$$

The approximation in (20) is reasonable since N is chosen to be greater than or equal to 200 in the experiments as shown in Section III. Using (18) and (20), it can be shown that

$$\frac{\boldsymbol{E}\left\{\frac{1}{N}\sum_{k=0}^{N-1} Y_k[i]\right\}}{\boldsymbol{E}\left\{\left(\prod_{k=0}^{N-1} Y_k[i]\right)^{1/N}\right\} \approx e^{\Sigma^l[i,i]/2}}. \quad (21)$$

If $Y_k^l s$ are i.i.d Gaussian random vectors, the expected value of the arithmetic mean is always greater than the expected value of the geometric mean by a factor of $e^{\Sigma^l[i,i]/2}$, as seen from (21).

We must compare the effect of noise on the MSN- and CMN-based features when the speech signal is corrupted by additive and/or convolutional noises. One way of doing this is to compare the expected value of the difference between the clean speech filterbank magnitude spectrum and noisy speech filterbank magnitude spectrum for the MSN- and CMN-based features. Let Z_m, Y_m be the clean speech and noisy speech (the noise could be additive, convolutional or both) filterbank magnitude spectrum for the m^{th} frame, respectively. Let $\boldsymbol{\mu_z}$, $\boldsymbol{\mu_y}$, $\boldsymbol{\Sigma_z}$, $\boldsymbol{\Sigma_y}$ be the mean of $\boldsymbol{Z_m}$, mean of $\boldsymbol{Y_m}$, covariance matrix of $\boldsymbol{Z_m}$, and covariance matrix of $\boldsymbol{Y_m}$, respectively. If we ignore the variances of $(1/N)\sum_{k=0}^{N-1} Z_k[i]$, and $(1/N)\sum_{k=0}^{N-1} Y_k[i]$ which are very small compared to the variances of $Z_m[i]$ and $Y_m[i]$, respectively, the expected value of the difference between the clean speech filterbank magnitude spectrum and noisy speech filterbank magnitude spectrum for the MSN-based feature can be computed as

$$E\left\{\left(\frac{Z_m[i]}{\frac{1}{N}\sum_{k=0}^{N-1} Z_k[i]}\right) - \left(\frac{Y_m[i]}{\frac{1}{N}\sum_{k=0}^{N-1} Y_k[i]}\right)\right\} \approx E\left\{\frac{Z_m[i]}{\mu_z[i]} - \frac{Y_m[i]}{\mu_y[i]}\right\} = \left\{\frac{\mu_z[i]}{\mu_z[i]} - \frac{\mu_y[i]}{\mu_y[i]}\right\} = 0. \quad (22)$$

If we ignore the variances of $\left(\prod_{k=0}^{N-1} Z_k[i]\right)^{1/N}$ and $\left(\prod_{k=0}^{N-1} Y_k[i]\right)^{1/N}$, which are very small compared to the variances of $Z_m[i]$ and $Y_m[i]$, respectively, the expected value of the difference between the clean speech filterbank magnitude spectrum and noisy speech filterbank magnitude spectrum for the CMN-based feature can be computed as

$$E\left\{\left(\frac{Z_m[i]}{\left(\prod_{k=0}^{N-1} Y_k[i]\right)^{1/N}}\right) - \left(\frac{Y_m[i]}{\left(\prod_{k=0}^{N-1} Y_k[i]\right)^{1/N}}\right)\right\} \approx E\left\{Z_m[i]e^{-\mu_z^l[i]} - Y_m[i]e^{-\mu_y^l[i]}\right\} = e^{\Sigma_z^l[i,i]/2} - e^{\Sigma_y^l[i,i]/2}. \quad (23)$$

It is known [1] that additive noise decreases the variance, and convolutional noise (with a variance) increases the variance in the log-filterbank domain. As seen from (22) and (23), the expected value of the difference between the clean speech filterbank magnitude spectrum and noisy speech filterbank magnitude spectrum is always zero for the MSN-based feature for all noise conditions but is always nonzero for the CMN-based feature for all noise conditions (it is assumed that noise has a variance). This could be the reason that the MSN-based speaker verification system gives better results than the CMN-based speaker verification system.

TABLE I
EERs FOR THE BASELINE SYSTEM

6 dB	12 dB	18 dB	Clean	Conv.
41.15	33.75	24.53	6.12	22.80

III. Experimental Setup and Results

The NIST 1998 speaker recognition [7] and NOISEX-92 [8] databases were used to evaluate and compare the performance of the CMN and MSN for convolutional and additive noise conditions on a speaker verification task.

The NIST 1998 speaker recognition database contains conversational telephone speech signals of 250 male and 250 female speakers sampled at 8 kHz. Only the training and test data of male speakers were used in the experiments. There are three training conditions: one session, two-session, and two-session-full. Two session full training data were used in the experiments. For each speaker, there are five training files with 1 min of speech in each taken from two different conversations collected from the same phone number for the two-session-full training condition. There are three different test conditions: test segment duration, same/different phone number, and same/different handset type. Only the test data with 30-s durations were used in the experiments. There are 1308 speech files collected from the same phone number using the same handset type and 1192 speech files collected from different phone numbers using different handset types for testing in the database. The total number of files with 30-s duration is 2500 for testing in the database. For each test file, there are one trial for the target speaker and nine trials for non-target speakers. Thus, the total number of trials is 13 080 for the same phone number using the same handset type and 25 000 for all files. Noise signals from NOISEX-92 database were downsampled from 16 to 8 kHz to have the same sampling rate with the NIST 1998 speaker recognition database. Then Factory, Operations room, Stitel, Speech, and Lynx noises were artificially added to the test speech signals (the NIST 1998 speaker recognition database) at SNR levels of 6, 12, and 18 dB to obtain noisy speech data. All the speech files were normalized to have the same average power. The speech signal was analyzed with a 32-ms hamming window every 10 ms. The FFT of each frame was used to calculate the magnitude spectrum of the signal. For the computation of mel-scaled log filterbank magnitude spectrums, 26 triangular mel-scaled bandpass filters were designed. MFCCs were computed by taking the DCT of mel-scaled log filterbank magnitude spectrums. The first twelve of the MFCCs as well as the zeroth coefficient were used. All feature vectors also include delta coefficients. Each speaker was modeled with a 64-component GMM. The background model was also modeled with a 64-component GMM and trained with all speaker's training data. The HTK toolkit [9] was used for training and testing.

We conducted a series of experiments under different noise conditions, different noise levels, using the CMN, MSN, and

TABLE II
EERS FOR THE CMN AND MSN

Duration	CMN					**MSN**				
	6 dB	12 dB	18 dB	Clean	Conv.	**6 dB**	**12 dB**	**18 dB**	**Clean**	**Conv.**
1 sec.	36.44	26.03	17.78	4.43	15.64	**33.78**	**23.47**	**16.39**	**4.59**	**15.28**
2 sec.	34.83	25.31	18.00	4.36	15.60	**31.82**	**22.77**	**16.64**	**4.36**	**14.80**
4 sec.	34.62	25.44	18.67	4.36	15.56	**30.96**	**22.69**	**16.61**	**4.43**	**14.80**
6 sec.	34.21	25.11	18.26	4.43	15.72	**30.81**	**22.40**	**16.90**	**4.28**	**14.72**
8 sec.	34.50	25.17	18.46	4.43	15.96	**30.95**	**22.72**	**17.02**	**4.51**	**14.92**
10 sec.	33.85	24.97	18.50	4.28	15.80	**30.87**	**22.6**	**16.93**	**4.59**	**14.96**
Average	34.40	25.20	18.38	4.37	15.73	**31.08**	**22.64**	**16.82**	**4.43**	**14.84**
% Red.						**9.65**	**10.16**	**8.49**	**-1.34**	**5.66**

baseline systems. No normalization was applied for the baseline system. We also conducted a series of experiments using different window lengths to see the effect of window length on the performance of CMN and MSN. Since it is difficult to compare all the results for the CMN, MSN, and baseline systems, equal error rates (EER) are averaged over all noise types (Factory, Operations room, Stitel, Speech, and Lynx noises).

Table I shows the average EERs for the baseline system. The first row shows noise levels and the second row shows the EERs. "Clean" represents the test data that were collected from the same phone number using the same handset type. "Conv." represents the all test data that were collected from the same or different phone numbers using the same or different handset types. Hence, "Conv" represents convolutional noise.

Table II shows the average EERs for six different window lengths, three additive noise levels using CMN and MSN. The first column shows the window length in seconds. As seen from Table II, increasing the window length more than 2 s do not change the performance significantly for both CMN and MSN. The proposed algorithm gave better result than CMN for all noise levels and noise types as seen from Table II. The CMN performed slightly better than the proposed method only for the clean speech condition. It is difficult to compare the performance for all conditions. Since the window length more than two second does not change the performance significantly, the EERs were averaged over window length excluding the one second window length. The ninth row shows the average EERs. The last row shows average percentage reduction in EERs over the CMN. The proposed algorithm reduces the average EERs by 9.65%, 10.16%, and 8.49% over the CMN for 6-, 12-, and 18-dB additive noises, respectively. CMNs and MSNs yielded approximately the same EERs for clean speech while the MSNs improved the performance 5.66% for the convolutional noise over the CMNs. The difference between the expected values of clean speech filterbank magnitude spectrum and noisy speech filterbank magnitude spectrum is zero for the MSN-based feature but nonzero for the CMN-based feature as shown in Section II-C. This could be the reason that the MSN-based feature yielded better results than the CMN-based feature for noisy speech.

As seen from Tables I and II, both CMN and MSN improve the performance significantly over the baseline system for the clean and noisy speech (additive and convolutional noises).

IV. CONCLUSIONS

In this letter, the use of average filterbank spectral magnitude was investigated for text-independent noise robust speaker verification. The relation between CMN and MSN has been derived. It was theoretically shown that both methods remove the convolutional bias. However, the expected values of the clean speech filterbank magnitude spectrum and noisy speech filterbank magnitude spectrum are the same for the MSN-based features, but different for the CMN-based features, which may cause performance degradation. The performance of CMN and MSN were compared. It was experimentally shown that the proposed convolutional bias removal algorithm outperforms the CMN for both convolutional and additive noises. It was also shown that it is not necessary to use all avaliable data for CMN and MSN. Only about 2 s of moving average of the cepstrums or magnitude spectrums are sufficient for convolutional bias removal.

REFERENCES

[1] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech Commun.*, vol. 16, no. 3, pp. 261–291, Apr. 1995.
[2] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, no. 2, pp. 254–272, Apr. 1981.
[3] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.
[4] J. Campbell, "Speaker recognition: A tutorial," *Proc. IEEE*, vol. 85, no. 9, pp. 1437–1462, Sep. 1997.
[5] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 1, pp. 72–83, Jan. 1995.
[6] E. J. Dudewicz and S. N. Mishra, *Modern Mathematical Statistics*. New York: Wiley, 1988.
[7] The 1998 Speaker Recognition Evaluation Plan NIST, 1998 [Online]. Available: www.nist.gov/speech/tests/spk/1998/current_plan.htm
[8] A. P. Varga, H. J. M. Steenekan, M. Tomlinson, and D. Jones, The Noisex-92 Study on the Effect of Additive Noise on Automatic Speech Recognition," Tech. Rep. DRA Speech Research Unit, 1992.
[9] S. Young, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book*, 2.1 ed. Cambridge, U.K.: Entropic Cambridge Research Laboratory Ltd., 1997.