IJoFCS (2008) 1, 60-69

e-forensic PRESS

The International Journal of FORENSIC COMPUTER SCIENCE

www.IJoFCS.org

Noise-Robust Speaker Recognition using Reduced Multiconditional Gaussian Mixture Models

Frederico Q. D'Almeida, Francisco Assis Nascimento, Pedro de A. Berger, and Lúcio M. da Silva

Abstract – Multiconditional Modeling is widely used to create noise-robust speaker recognition systems. However, the approach is computationally intensive. An alternative is to optimize the training condition set in order to achieve maximum noise robustness while using the smallest possible number of noise conditions during training. This paper establishes the optimal conditions for a noise-robust training model by considering audio material at different sampling rates and with different coding methods. Our results demonstrate that using approximately four training noise conditions is sufficient to guarantee robust models in the 60 dB to 10 dB Signal-to-Noise Ratio (SNR) range.

Keywords – Automatic speaker recognition, Gaussian mixture models, Multiconditional models, noise robustness, computing power.

1. Introduction

Modern Automatic Speaker Recognition (ASR) systems, based on Gaussian Mixture Models (GMMs) and using Mel Frequency Cepstral Coefficient (MFCC) parameters, have proven to be quite successful at identifying the author of a voice extract. Nevertheless, a limiting factor for the performance of such systems is the quality of the source material available for comparison. In order to make ASR systems robust to noise, a common strategy is to use Multiconditional Model Training (Lippmann et al., 1987; Deng et al., 2000; Ming et al., 2007a). This approach models the speaker's voice using a number of low complexity models of the same speaker, each one trained at a particular Signal-to-Noise Ratio (SNR). Recently, Ming et al. (2007b) employed a variation of this technique where a single complex model with a large number of Gaussian components was trained simultaneously with different SNR samples. This approach was a replacement for training with a group of simple models. Despite the consistent improvement in the model's noise tolerance, both approaches produced a remarkable increase in model complexity and, consequently, in the computational cost of the overall task.

Until now, the effects resulting from adding a specific level of noise to test samples submitted to a system trained on different noise conditions was unknown. Therefore, the optimal SNR to be used on noise-robust multiconditional model training was likewise unknown. Somewhat arbitrary choices are generally made. For instance, increasing the model complexity and the number of noise levels used for training will surely improve

the robustness of the system. However, for most speaker recognition applications, response time and processing power are limiting factors that can preclude excessive model complexity.

The goal of this paper is to present a comprehensive study on the performance of ASR systems under different noise conditions, and to explore the optimal audio database for testing signals at different sampling rates and with different coding methods. In particular, we explore the precise system performance degradation that results from testing models under mismatched noise conditions. We also show that it is possible to effectively determinate the relevant SNR for the multiconditional model training that simultaneously guarantees noise robustness and keeps the model's complexity low. In this sense, our work proposes a low-complexity model that has little effect on performance degradation (in comparison to the complete model) while also maintaining a low overall computational cost.

2. Multicondition Model

Speaker recognition systems based on GMM using MCFF parameters are a widely employed, successful technique for recognizing voice excerpts in large speaker databases when the audio has not been corrupted by noise (Campbell, 1997). Noisy speaker recognition systems, however, are still under development. The most common technique when dealing with noisy speaker recognition systems is multiconditional model training. This technique builds noise-robust speaker models by incorporating noisy audio to the model training.

In order to set up a multiconditional model, several training audio sets at different noise levels must be available. Here we represent these training data sets as Φ_n, where the index *n* refers to the noise level. Often, these training sets are built up from a single noiseless training database,Φ_0 by adding progressive amounts of simulated white noise.

In traditional multiconditional model training, each training data set Φ_n is used to train one specific speaker model. The set of models for each speaker is then combined to form the multiconditional model. This is mathematically represented by

$$p(X \mid S) = \sum_{n=0}^{N} p(X \mid S, \Phi_n) P(\Phi_n \mid S), \quad (1)$$

where *X* is a feature vector extracted from the test data, *S* represents the speaker, $p(X \mid S, \Phi_n)$ is the likelihood function of vector *X* for the speaker *S* trained on set Φ_n, and $P(\Phi_n \mid S)$ is the prior probability for the occurrence of noise condition Φ_n for speaker *S*. Obviously, the prior probabilities are restricted to

$$\sum_{n=0}^{N} P(\Phi_n \mid S) = 1. \quad (2)$$

If using GMM speaker models, each likelihood function $p(X \mid S, \Phi_n)$ has the form

$$p(X \mid S, \Phi_n) = \sum_{i=1}^{M} p_{n,i} b_{n,i}(X), \quad (3)$$

where the $p_{n,i}$ are the mixture coefficients and $b_{n,i}$ are the GMM components. The mixture coefficients are restricted to

$$\sum_{i=1}^{M} p_{n,i} = 1, \quad (4)$$

since they are the prior probabilities of each Gaussian component in the GMM model. The GMM components $b_{n,i}$ are Gaussian distributions of the form

$$b_{i,n}(\vec{x}) = \frac{1}{(2\pi)^{D/2} \left|\Sigma_{i,n}\right|^{1/2}} \exp\left\{ -\frac{(\vec{x} - \vec{\mu}_{i,n})^T \Sigma_{i,n}^{-1} (\vec{x} - \vec{\mu}_{i,n})}{2} \right\}, \quad (5)$$

where *D* is the dimension of the model.

Recently, Ming et al. (2007b) proposed an alternative multiconditional model training. In this novel approach, the training data sets Φ_n where gathered into a unified multiconditional data set $\Phi = \Phi_0 + \Phi_1 + \ldots + \Phi_N$. The model training was performed in a single turn, considering the multiconditional model as a large GMM model. Although this new approach makes it difficult to visualize the underlying multiconditional nature of the model, it is nonetheless a more flexible

method of treating the multiconditional training than the traditional scheme. As one can easily verify by inserting eq. into eq. , the multiconditional model can be re-written as

$$p(X \mid S) = \sum_{n=0}^{N} \left[\sum_{i=1}^{M} p_{n,i} b_{n,i}(X) \right] P(\Phi_n \mid S) \cdot \quad (6)$$

After making a simple re-indexing from the dual reference *n* and *i* to the single reference *k*, we have

$$p(X \mid S) = \sum_{n=0}^{N} \sum_{i=1}^{M} p'_{n,i} b'_{n,i}(X) = \sum_{k=0}^{(N+1)M} p'_k b'_k(X), \quad (7)$$

where

$$\begin{aligned} p'_{n.i} &= p_{n,i} P(\Phi_n \mid S) \\ b'_{n.i} &= b_{n,i} P(\Phi_n \mid S) \end{aligned} \quad . \quad (8)$$

Treating the multiconditional model as a unique large GMM model has the advantage of having only one constraint, namely

$$\sum_{k=0}^{(N+1)M} p'_k = 1 \,, \quad (9)$$

while the traditional multiconditional model has *N+1* constrains: one as indicated by eq. , and *N* constraints from the multiple instances of eq. .

3. Model Reduction

Using any of the multiconditional model formulations inevitably causes a considerable increase in model complexity and the required computational power. The complexity of models grows linearly with *N*, the number of conditions used in the multiconditional training. Although this linear growth may not appear to be a severe problem for speaker recognition tasks, one must consider that the whole speaker set must be tested in order to find the author of a voice excerpt. Using multiconditional modeling means that the complexity of the entire speaker universe will grow by a factor of *N*, transforming a very large problem into an even larger one.

The most straightforward way of reducing the demanded computer effort in multiconditional speaker recognition tasks is by minimizing the number of conditions used during model training. This reduction, however, must be strictly controlled in order to maintain the noise-robustness of the method. Building a multiconditional model that requires a minimum number of training conditions but is still robust to varying noise conditions requires an analysis of the exact effect that each training condition has on the overall system performance.

With this aim in mind, we have carried out a set of noise mismatch tests. Our mismatched testing procedure consists of submitting a system of uniconditional models, trained with unique noise conditions Φ_m, to speaker recognition tasks with questioned voice excerpts of different noise characteristics Ψ_n, where *n = 1,...,N*. Here, Ψ represents the testing audio data set, which is different from the training data set Φ, and Ψ_n represents the audio data set Ψ at noise level *n*. Although Φ and Ψ are different audio sets, the noise level in Φ_m is the same used in Ψ_n for every *m = n*. By doing this, the noise range that each individual uniconditional model represents can be determined. Thus, one can establish the optimal noise conditions set to build up the multiconditional model.

The actual mismatched testing was carried out by training uniconditional systems at 60 dB, 50 dB, 40 dB, 30 dB, 26 dB, 20 dB, 16 dB and 10 dB SNR audio sets ($\Phi_0,\ldots,\Phi_7$), and testing each one at 60 dB, 50 dB, 40 dB, 30 dB, 26 dB, 20 dB, 16 dB and 10 dB SNR audio sets ($\Psi_0,\ldots,\Psi_7$). The 60 dB SNR corresponds to the "noiseless" audio, as this is the average estimated SNR level of the original audio database. Additional mismatched testing was performed using different audio quantization schemes for the training and test.

4. Audio Database Description

We used a voice database containing 30 distinct speakers, half male and half female. Each speaker was recorded reading a pre-defined Brazilian-Portuguese text; the same text was used for all

speakers. Each recording was then fragmented in 21 files for a total of 630 audio extracts. The first and last cut points in these files were the same for all speakers in relation to the text content of speech. Thus, we generated 21 files for each speaker, with one used for model training (Φ_0) and the other 20 used for testing (Ψ_0). The duration of each voice extract was approximately 30 s.

All of the recordings were performed in acoustically prepared environments, using professional audio caption microphones and plates. The files were captured at a sample rate of 16 kHz, 16-bit quantization in monaural mode. From this initial audio database, three subsampled 8 kHz database versions were built for use in the simulations: one with 16-bits per sample, one with 8-bit μ-law coding, and one with 8-bit linear coding. A second database version resampled at 11 kHz and 16-bits was also used.

4.1 Pre-processing

All audio files were pre-processed before testing. First, each file was normalized such that the peak amplitude corresponded to 100% of the maximum quantization value. Each file's silent extracts were then excluded. This was performed using an automatic silence detector based on the measure of the signal energy in 20 ms windows, with an overlap of 15 ms (5 ms advances) and a manually defined silence threshold. The silence threshold definition was performed by successively adjusting and retesting for verification.

4.2 Noise Addition

The noisy audio samples were generated from the "noiseless" audio database (Φ_0, Ψ_0) by adding a precise amount of Additive White Gaussian Noise (AWGN). The procedure was as follows. First, we calculated the average signal energy (E_s) in the pre-processed audio extracts, given by

$$E_s = \sum_m y_s^2[m], \tag{10}$$

where y_s is the noiseless audio waveform and m is the time index of the sample audio extracts. A noise vector (y_n) was then generated with the same dimension as the audio vector signal (y_s) and containing zero mean Gaussian distributed samples. The energy (E_n) of this noise vector was also calculated, given by

$$E_n = \sum_m y_n^2[m]. \tag{11}$$

The noise vector amplitudes were then adjusted in order to get the desirable SNR, or

$$y'_n = y_n \cdot \sqrt{10^{\frac{-SNR}{10}} \cdot \frac{E_s}{E_n}}. \tag{12}$$

Finally, the noise vector with adjusted amplitudes was added to the signal vector, resulting in the noisy audio vector (y), or

$$y = y_s + y'_n. \tag{13}$$

The SNRs used in the simulations were 60 dB, 50 dB, 40 dB, 30 dB, 26 dB, 20 dB, 16 dB and 10 dB, both for the training data sets (Φ_n) and the testing data sets (Ψ_n). As mentioned earlier, the 60 dB SNR audio is equivalent to "noiseless" audio since this is the average estimated SNR level of the original audio database. This SNR level was calculated based on the average signal energy of the speech and silent extracts of the entire audio database before preprocessing. In other words, the 60 dB SNR level has no noise added.

The noise addition procedure was performed on the 16 kHz / 16-bit original database using 32-bit arithmetic. The noisy databases were obtained by resampling and recoding the high coding audio set. This was done in order to avoid quantization error accumulation which would occur if the noise addition had been performed over the 8-bit audio.

It should be remarked that the calculation of the noiseless audio energy in eq. was performed after removing the silent intervals. Thus, the calculated average power (total energy divided by total time) of that signal is higher than it would be if we considered the original signal (with the silent intervals intact), since the segments of low energy were excluded from the calculation. Consequently, to obtain the established SNR, the average power

of the noise signal (y'_n) added to the signal is higher than it would be if we considered the original unedited audio file. Based on our tests, we concluded that the SNR values which were presented in this paper are roughly 10% higher than those obtained if the noise addition process was performed on the original audio extracts.

Note that we chose this particular methodology for measuring SNRs for a couple of reasons. First, since the locution rhythms and time pauses between words and phrases vary for distinct individuals, the measured SNR is sensitive to these particular individual characteristics. Therefore, it is possible to achieve distinct SNR values even if the signal and noise energy are kept fixed. Second, the identification of the extracts with voice and silent intervals is much simpler in noiseless audio (where we can use a simple energy detector) than in noisy audio.

Despite the fact that the noise addition was performed in a simulated computer environment, real ASR system tests assure that this procedure closely reproduces the real-world acoustic addition of noise (Ming et al., 2007b).

5. Recognition System

The recognition system used for performance evaluation is based on Gaussian Mixture Models (GMMs). This kind of speaker modeling is currently the widely employed and has yielded satisfactory results in ASR systems, especially for noisy situations (Graciarena et al., 2007; Reynolds et al., 1992; Rose et al., 1991).

As shown in eq. , GMMs are a combination of individual Gaussian models which are intended to represent the different vocal productions of a single person (Reynolds et al., 1992b). Each Gaussian of the composite GMM first models one specific sound class. In this way, the complete group of Gaussians is capable of modeling a large number of sound classes to an acceptable level of precision, and could therefore recognize the speaker independently of the spoken content.

For each simulation performed here, we used a 16 component GMM following the approach of Reynolds et al. (1995). Increasing the model complexity further does not significantly improve system performance. Also, for an additional simplification, the model covariance matrix was restricted to be diagonal. Such a restriction does not significantly impact the system results (Reynolds et al., 2000).

The GMMs were trained using Mel Frequency Cepstral Coefficients (MFCCs). This approach has proven to be a superior chose of parameters for speaker representation when compared to other parameterizations (D'Almeida et al., 2006; Jankowski et al., 1995). Additionally, this particular type of parameter is widely used in other ASR studies, which facilitates comparisons with the wider literature.

The MFCC parameters were calculated for each 20 ms window of audio, without window overlapping, through filter banks applied directly to the signal frequency spectrum as calculated in this same window. From each window, 12 parameters were extracted.[1] After computing the MFCC parameters, the cepstral mean subtraction procedure was applied as described in Reynolds et al. (1995), both for the training and testing phases. Spectral mean subtraction techniques are a well known tool to improve ASR systems performance (Schwartz et al., 1993; Jankowski et al., 1995; Vaseghi et al., 1997), which aim to minimize constant or slowly varying spectrum background noise.

6. Results

The system performance evaluation was computed using the correct speaker recognition ratio. For each situation (sampling frequency, training SNR and test SNR), 600 speaker recognition tasks were performed, one for each audio test sample (30 speakers multiplied by 20 test audio samples per speaker). The results of

1 The same number of 12 MFCC parameters was used for 16 kHz, 11 kHz and 8 kHz audio databases, although the number of filters in the filterbank was properly adjusted to each sampling frequency.

the complete mismatched testing procedure are organized into two-dimensional matrixes. Rows indicate how a system trained with a particular SNR performs for different noise test conditions, and columns indicate how a specific test SNR is handled by different training conditions. These results are displayed in tables 1 through 7, containing the correct recognition rates for each tested configuration. Each table refers to a specific audio coding and sampling frequency.

The mismatched testing was performed using different audio quantization schemes for model training and testing. The results of training the models with 16-bit audio and testing with 8-bit μ-law audio are summarized in table 6. The results of training the models with 16-bit audio and testing with 8-bit linear audio are summarized in table 7.

Our results show that it is possible to build noise-robust multiconditional models using a reduced set of conditions. For example, in the 16 kHz / 16-bit test audio, training conditions of 60 dB, 30 dB, 20 dB and 10 dB are enough to build a decent model that performs close to the full model. Fig. 1 shows a comparison of the best results for each noise condition considering all training conditions used (full set) and the reduced set. With the exception of the 30 dB noise condition, no significant performance differences were found between the reduced training set case and the full training set.

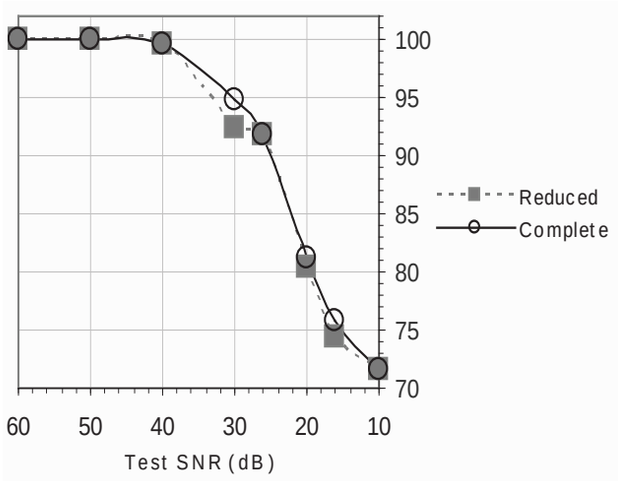

Fig. 1. Comparison of the best correct recognition rate results for each noise condition considering complete and reduced training sets. Testing and training model audio were sampled at 16 kHz and coded with 16-bits.

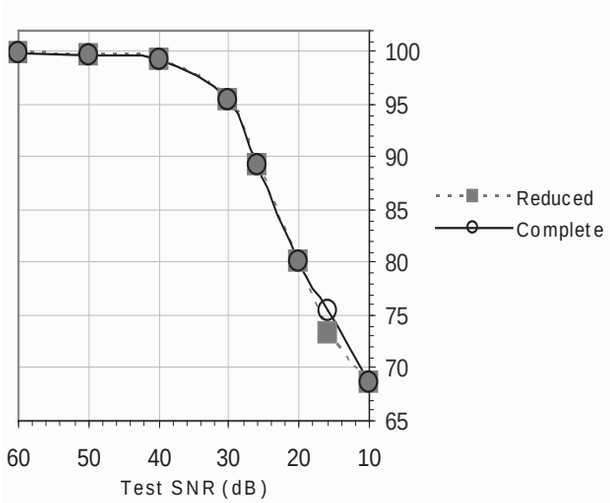

Fig. 2. Comparison of best correct recognition rate results for each noise condition considering complete and reduced training sets. Testing and training model audio were sampled at 11 kHz and coded with 16-bits.

For the 11 kHz / 16-bit and 8 kHz / 16-bit test audio, the differences between the full training condition set and the reduced set is even less noteworthy, as seen in fig. 2 and fig. 3. In these simulations, the reduced set used was the same as in the 16 kHz case (60 dB, 30 dB, 20 dB and 10 dB). For the 8 kHz / 8-bit μ-law test audio, two sets of simulations were performed. We trained the models with the same audio codification scheme, and with using 8 kHz / 16-bit audio. Both situations are represented in fig. 4, where we find that although the test audios were coded at 8-bit μ-law, the models trained with 16-bit audio had superior performance, especially for the low noise cases. For the reduced training condition models and for models trained with 8-bit μ-law audio, the optimal reduced condition training set is 50 dB, 26 dB, 20 dB and 10 dB. For models trained with 16-bit audio, the best reduced condition training set is 50 dB, 30 dB, 20 dB and 10 dB. With these reduced training conditions set, one can verify that the reduced model trained with 16-bit audio maintained good performance at the 16 dB noise level, while the reduced model trained with 8-bit μ-law audio had a noticeable performance loss.

For the 8 kHz / 8-bit test audio shown in fig. 5, the performance difference between systems trained with the same audio coding method and

with 16-bit audio were less expressive. Again, models trained with 16-bit audio showed better performance. The optimal reduced conditions training sets for this case are 60 dB, 30 dB, 20 dB and 10 dB for models trained with 8-bit audio, and 50 dB, 30 dB, 20 dB and 10 dB for models trained with 16-bit audio.

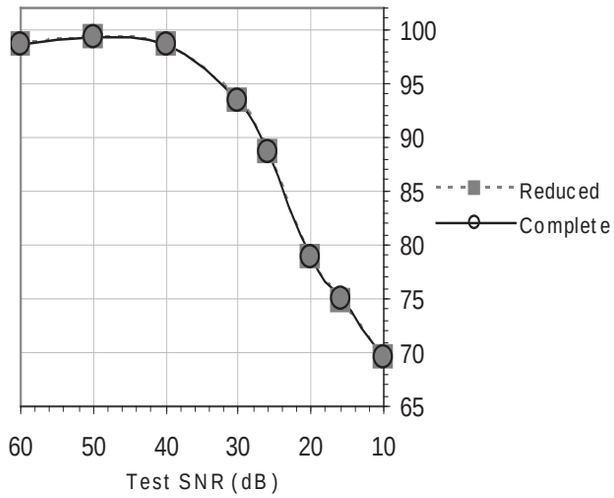

Fig. 3. Comparison of best correct recognition rate results for each noise condition considering complete and reduced training sets. Testing and training model audio were sampled at 8 kHz and coded with 16-bits.

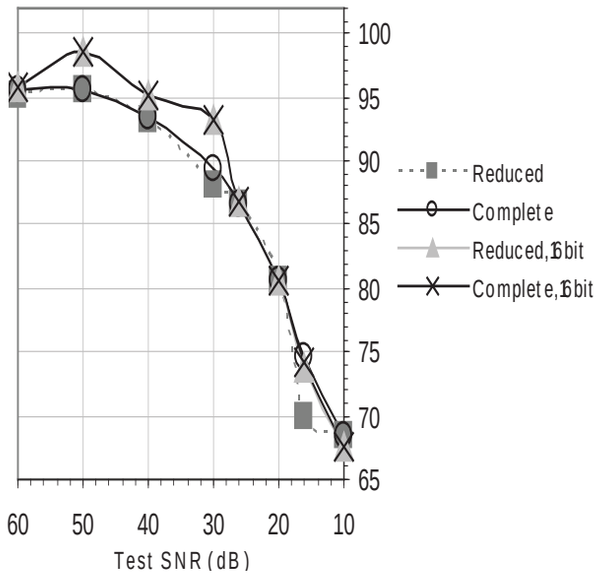

Fig. 4. Comparison of best correct recognition rate results for each noise condition considering complete and reduced training sets. The testing audio was sampled at 8 kHz and coded with 8-bit μ-law. This figure shows the results for two sets of simulations: training the models with the same audio coding scheme as for the testing audio, and training with audio sampled at 8 kHz and coded with 16-bits.

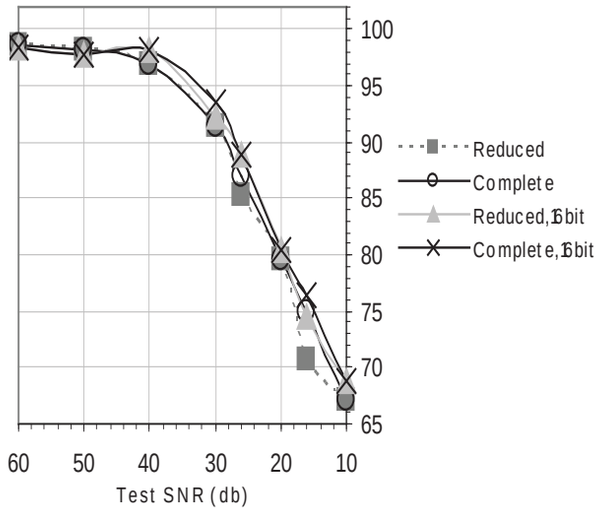

Fig. 5. Comparison of best correct recognition rate results for each noise condition, considering complete and reduced training sets. The testing audio was sampled at 8 kHz and coded with 8-bits. This figure shows the results for two sets of simulations: training the models with the same audio coding scheme as for the testing audio, and training the audio sampled at 8 kHz and coded with 16-bits.

7. Discussion

Our results indicate that to build robust models in the low noise range from 60 dB to 30 dB SNR, only one training condition is sufficient. For the 16-bit audio simulations (at 16 kHz, 11 kHz and 8 kHz), the noiseless condition training provided acceptable noise robustness in this range. For the 8-bit μ-law simulations with models trained using 16-bit audio, which performed better than the 8-bit μ-law trained models, the 50 dB training condition yielded a robust model in the 60 dB to 30 dB SNR range. For the 8-bit linear simulations, models trained with 16-bit noiseless audio showed acceptable results. For models trained with 8-bit linear audio, no unique audio condition performed satisfactorily in the 60 dB to 30 dB SNR range. In this case, acceptable results can be achieved using two training conditions, 60 dB (noiseless) and 30 dB noise level audio. On the other hand, for the high noise range between 26 dB and 10 dB, a robust model demands at least three training conditions (30 dB, 20 dB and 10 dB), and, in some cases, a moderate performance increase in the 16 dB SNR case can be achieved by using the 16 dB training condition.

Reduced Multiconditional Models have proven to be a good technique to build noise-robust speaker recognition systems with minimal model complexity. By correctly choosing the training conditions set, it is possible to build speaker recognition systems that are robust up to 10 dB SNR using only four training conditions.

8. Conclusions

In this work, the performance degradation of ASR systems was analyzed as a function of test audio SNR mismatches with respect to the training audio SNR. The aim of this paper was to identify which audio SNR were relevant for model training in order to guarantee noise robustness and, at the same time, keep model complexity to a minimum. Out results indicate that, for the tested sampling frequencies and audio coding methods, a training model with only four selected noise conditions is sufficient to provide a level of noise robustness very close to that achieved by the full eight condition model. This leads to a 50% model complexity reduction with little effect on overall system performance.

IX Acknowledgments

This work was accomplished with the support of the Sagem Orga of Brazil.

References

[1] R. P. Lippmann,E. A. Martin, D. B. Paul, Multi-style Training for Robust Isolated Word Speech Recognition, Proceedings of ICASSP, pp. 705-708, Dallas, TX, USA, 1987.

[2] L. Deng, A. Acero, M. Plumpe, and X.-D. Huang, Large Vocabulary speech recognition under adverse acoustic environments, Proc. ICSLP'00, pp. 806-809, Beijing, China, 2000.

[3] J. Ming, B. Hou, Speech Recognition in Unknown Noise Conditions, Speech Recognition and Understanding, Michael Grimm and Kristian Kroschel editors, Austria, 2007.

[4] J. Ming, T. Hazen, J.R. Glass, D.A. Reynolds, Robust Speaker Recognition in Noisy Conditions, IEEE Trans. on Audio, Speech and Lang. Proc., vol. 15, pp. 1711-1723, Julho/2007.

[5] J.P. Campbell Jr., Speaker Recognition: A Tutorial, Proceedings of the IEEE, Vol. 85, No. 9, pp. 1437-1462, 1997.

[6] R. Schwarz et al, Comparative experiments on large vocabulary speech recognition, Proc. ARPA Workshop on Human Lang. Tech., 1993.

[7] D.A. Reynolds, A Gaussian Mixture Modeling Approach to Text-Independent Speaker Identification, Ph. D. Thesis, Georgia Inst. of Tech, 1992.

[8] D.A. Reynolds e R.C. Rose, Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models, IEEE Trans. Speech and Audio Proc., Vol. 3, no. 1, pp 72-83, 1995.

[9] D.A. Reynolds, T.F. Quatieri e R.D.Dunn, Speaker Verification Using Adapted Gaussian Mixture Models, Digital Signal Processing, Vol. 10, nos 1-3, pp. 19-41, 2000.

[10] F.Q. D'Almeida e F.A.O Nascimento, Comparação de Desempenho de Parâmetros da Fala em Sistemas de Reconhecimento Automático de Locutor, Congresso Brasileiro de Automática – CBA, 2006.

[11] M. Graciarena et at, Noise Robust Speaker Identification for Spontaneous Arabic Speech, Int. Conf. on Accoustic, Speech and Signal Proc. – ICASSP, 2007.

[12] C. Jankowski, J. Hoang-Doan e R. Lippmann, A Comparison of signal processing front ends for automatic word recognition, IEEE Trans. Speech and Audio Proc., vol. 3, pp. 286-293, 1995

[13] S. Vaseghi e B. Milner, Noise compensation methods for hidden markov model speech recognition in adverse environments, IEEE Trans. on Speech and Audio Proc., vol. 5, pp. 11-21, Jan. 1997.

[14] D.A. Reynolds e R.C. Rose, An Integrated Speech-Background Model for Robust Speaker Identification, Proc. Intl. Conf. Accoustic, Speech and Signal Proc., 1992.

[15] R.C. Rose, J. Fitzmaurice, E.M.Hofstetter e D.A. Reynolds, Robust Speaker Identification in Noisy Environments Using Noise Adaptive Speaker Models, Proc. Intl Conf. Accoustic, Speech and Signal Proc., 1991.

Table 1: Correct recognition rates for tests performed using 16 kHz / 16-bit audio and model trained with 16 kHz / 16 bit audio

SNR (dB)		TEST							
		30	25	20	15	13	10	8	5
TR.	30	100.0	100.0	99.7	92.3	80.5	53.3	32.0	7.2
	25	98.8	99.0	98.8	93.7	88.2	53.8	33.0	13.5
	20	96.2	96.8	97.5	94.8	87.5	56.8	30.8	7.8
	15	75.0	79.7	86.0	92.5	91.8	77.5	47.2	9.2
	13	60.5	59.8	70.5	85.2	86.3	81.2	56.8	14.7
	10	25.7	32.7	40.8	63.7	70.8	80.5	74.5	30.2
	8	12.5	13.5	19.2	37.7	52.8	71.2	75.8	53.8
	5	6.7	6.7	7.2	16.2	17.2	35.7	54.0	71.7

Table 2: Correct recognition rates for tests performed using 11 kHz / 16-bit audio and model trained with 11 kHz / 16-bit audio.

SNR (dB)		TEST							
		30	25	20	15	13	10	8	5
TR. INI	30	99.8	99.7	99.2	95.5	84.0	51.5	30.3	8.0
	25	99.5	99.0	98.2	93.3	86.3	58.8	33.8	18.0
	20	96.0	96.5	97.3	93.3	85.0	54.0	31.8	10.5
	15	72.7	75.5	84.2	91.7	89.2	71.8	45.8	8.7
	13	59.0	58.8	70.3	84.2	87.7	78.5	54.3	13.3
	10	24.0	27.3	40.3	62.7	73.5	80.2	73.3	27.0
	8	15.3	16.3	19.7	40.3	56.7	70.7	75.3	51.7
	5	7.2	5.0	8.0	20.0	21.8	44.3	56.3	68.7

Table 3: Correct recognition rates for tests performed using 8 kHz / 16-bit audio and model trained with 8 kHz / 16-bit audio.

SNR (dB)		TEST							
		30	25	20	15	13	10	8	5
TRAIN	30	98.5	99.2	98.5	93.3	86.0	53.3	30.7	16.2
	25	98.3	98.0	98.5	92.5	86.3	53.0	30.8	9.2
	20	94.7	95.8	96.0	92.2	85.5	57.2	38.5	10.5
	15	71.2	73.5	77.8	90.3	88.5	69.3	44.7	14.0
	13	52.3	61.5	66.8	82.8	86.8	77.8	52.3	18.2
	10	28.8	26.8	35.0	62.0	70.8	78.8	74.7	28.3
	8	23.2	22.8	21.5	38.5	51.5	71.5	75.0	52.5
	5	17.2	16.3	20.8	20.5	27.0	44.3	57.7	69.5

Table 4: Correct recognition rates for tests performed using 8 kHz / 8-bit μ-law audio and model trained with 8 kHz / 8-bit μ-law audio.

SNR (dB)		TEST							
		30	25	20	15	13	10	8	5
TRAIN	30	95.5	95.3	92.7	76.0	65.0	34.5	23.0	18.2
	25	95.3	95.3	93.5	76.7	66.0	33.7	25.2	22.8
	20	95.2	95.7	93.3	81.0	68.2	41.3	29.3	23.3
	15	88.0	89.2	90.8	89.3	82.7	62.7	40.0	26.7
	13	79.3	78.0	80.8	88.2	86.5	69.8	52.3	26.8
	10	49.3	50.8	53.7	69.5	77.8	80.7	70.0	42.2
	8	29.8	25.7	32.5	45.5	53.8	73.5	74.7	56.0
	5	8.5	9.2	10.8	13.0	15.7	29.2	48.8	68.5

Table 5: Correct recognition rates for tests performed using 8 kHz / 8-bit linear audio and model trained with 8 kHz / 8-bit linear audio.

SNR (dB)		TEST							
		30	25	20	15	13	10	8	5
TRAIN	30	95.5	97.7	92.5	90.7	77.0	50.8	28.5	8.2
	25	95.8	98.5	95.2	93.2	75.3	54.3	26.7	10.8
	20	92.7	95.8	94.3	92.0	82.2	56.8	33.2	9.8
	15	76.5	77.7	80.8	90.5	86.8	71.0	45.2	13.0
	13	62.3	65.8	68.7	83.5	86.5	78.0	58.3	16.8
	10	36.5	36.5	43.3	64.7	70.2	80.7	73.7	28.5
	8	20.5	24.0	25.3	40.8	52.8	70.0	74.2	49.7
	5	19.7	20.0	19.3	20.7	29.2	40.2	58.0	67.7

Table 6: Correct recognition rates for tests performed using 8 kHz / 8-bit μ-law audio and model trained with 8 kHz / 16-bit audio.

SNR (dB)		TEST							
		30	25	20	15	13	10	8	5
TRAIN	30	97.7	97.5	97.3	92.0	81.3	54.0	32.5	14.2
	25	98.5	97.8	98.2	92.3	82.7	52.8	32.0	12.0
	20	97.3	96.7	97.0	93.5	84.5	59.0	33.3	11.5
	15	84.2	85.3	86.8	92.3	88.8	70.3	42.8	12.2
	13	71.3	71.7	73.3	84.0	86.8	78.0	55.8	18.5
	10	42.5	43.5	48.7	64.7	70.5	80.3	74.5	31.8
	8	22.5	27.2	30.7	41.3	53.8	70.7	76.3	49.2
	5	18.7	19.0	19.0	25.7	32.2	42.5	56.5	68.8

Table 7: Correct recognition rates for tests performed using 8 kHz / 8-bit linear audio and model trained with 8 kHz / 16-bit audio.

SNR (dB)		TEST							
		30	25	20	15	13	10	8	5
TRAIN	30	97.7	97.5	97.3	92.0	81.3	54.0	32.5	14.2
	25	98.5	97.8	98.2	92.3	82.7	52.8	32.0	12.0
	20	97.3	96.7	97.0	93.5	84.5	59.0	33.3	11.5
	15	84.2	85.3	86.8	92.3	88.8	70.3	42.8	12.2
	13	71.3	71.7	73.3	84.0	86.8	78.0	55.8	18.5
	10	42.5	43.5	48.7	64.7	70.5	80.3	74.5	31.8
	8	22.5	27.2	30.7	41.3	53.8	70.7	76.3	49.2
	5	18.7	19.0	19.0	25.7	32.2	42.5	56.5	68.8

F.Q.D'Almeida. He was born in Salvador-BA, Brazil, on January 24, 1978. He graduated in Electrical Engineering, Federal University of Bahia - UFBA, Salvador-BA, Brazil, 2000, got his Master Degree in Electrical Engineering, UFBA, 2003, and graduated in Physics, University of Brasilia - UnB, 2006. He is pursuing his Doctorate Degree in Electrical Engineering at UnB. His field of study is Automatic Speaker Recognition. He also works as a forensic expert at Brazilian Federal Police.

F. Assis Nascimento received his B.Sc. in Electrical Engineering from the University of Brasilia in 1982, his M.Sc. in Electrical Engineering from the Federal University of Rio de Janeiro (UFRJ), in 1985, and his Ph.D. in Electrical Engineering from UFRJ in 1988. Currently, he is an Associate Professor at the University of Brasilia and a coordinator of the GPDS (Grupo de Processamento Digital de Sinais).

Pedro de A. Berger graduated in Electrical Engineering at Federal University of Ceara - UFC, Fortaleza-CE, Brazil in 1999, earned his M.Sc. in Electrical Engineering at the University of Brasília (UnB) in 2002 and his Ph.D. at UnB in 2006. He has been a professor in the Department of Computer Science at UnB since 2006. His field of study includes digital signal processing, artificial neural networks and biomedical engineering.

Lúcio M. da Silva was born in Delfinópolis, MG, Brazil, on April 27, 1958. He received the B.S. in electrical engineering from Pontifical Catholic University of Minas Gerais, in 1981, the M.S. degree from University of Brasilia, in 1989, and the Ph. D. degree from Pontifical Catholic University of Rio de Janeiro, in 1996. He is with the Electrical Engineering Department of University of Brasilia. He is involved in teaching and research activities in speech signal processing and digital transmission systems.