International Journal of Speech Technology

# A Nonlinear Autoregressive Model for Speaker Verification

--Manuscript Draft--

| | |
|---|---|
| **Manuscript Number:** | |
| **Full Title:** | A Nonlinear Autoregressive Model for Speaker Verification |
| **Article Type:** | Manuscript |
| **Keywords:** | Gaussian mixture models;  mixture autoregressive model;  nonlinear statistical models; speaker verification |
| **Corresponding Author:** | Joseph Picone, Ph.D.<br>Temple University<br>Elkins Park, Pennsylvania UNITED STATES |
| **Corresponding Author Secondary Information:** | |
| **Corresponding Author's Institution:** | Temple University |
| **Corresponding Author's Secondary Institution:** | |
| **First Author:** | Sundararajan Srinivasan, PhD |
| **First Author Secondary Information:** | |
| **Order of Authors:** | Sundararajan Srinivasan, PhD |
| | Tao Ma, PhD |
| | Georgios Lazarou, PhD |
| | Joseph Picone, Ph.D. |
| **Order of Authors Secondary Information:** | |
| **Abstract:** | Gaussian Mixture Models (GMM) have been the most popular approach in speaker recognition and verification for over two decades. The inefficiencies of this model for signals such as speech are well documented and include an inability to model temporal dependencies that result from nonlinearities in the speech signal. The resulting models are often complex and overdetermined, which leads to a lack of generalization. In this paper, we present a nonlinear mixture autoregressive model (MixAR) that attempts to directly model nonlinearities in the trajectories of the speech features. We apply this model to the problem of speaker verification. Experiments with synthetic data demonstrate the viability of the model. Evaluations on standard speech databases, including TIMIT, NTIMIT, and NIST-2001, demonstrate that MixAR, using only half the number of parameters and only static features, can achieve a lower equal error rate when compared to GMMs, particularly in the presence of previously unseen noise. Performance as a function of the duration of both the training and evaluation utterances is also analyzed. |

# A NONLINEAR AUTOREGRESSIVE MODEL FOR SPEAKER VERFICATION[1]

Sundararajan Srinivasan[2], Tao Ma[3], Georgios Lazarou[4] and Joseph Picone[5]

*Abstract*— Gaussian Mixture Models (GMM) have been the most popular approach in speaker recognition and verification for over two decades. The inefficiencies of this model for signals such as speech are well documented and include an inability to model temporal dependencies that result from nonlinearities in the speech signal. The resulting models are often complex and overdetermined, which leads to a lack of generalization. In this paper, we present a nonlinear mixture autoregressive model (MixAR) that attempts to directly model nonlinearities in the trajectories of the speech features. We apply this model to the problem of speaker verification. Experiments with synthetic data demonstrate the viability of the model. Evaluations on standard speech databases, including TIMIT, NTIMIT, and NIST-2001, demonstrate that MixAR, using only half the number of parameters and only static features, can achieve a lower equal error rate when compared to GMMs, particularly in the presence of previously unseen noise. Performance as a function of the duration of both the training and evaluation utterances is also analyzed.

***Keywords***— Gaussian mixture models, mixture autoregressive model, nonlinear statistical models, speaker verification

Manuscript submitted February 15, 2013.

[1] This material is based upon work supported by the National Science Foundation under Grant No. IIS-0414450. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

[2] S. Srinivasan is with Nuance Communications Inc., 1198 East Arques Avenue Sunnyvale, CA 94085, USA (phone: 408-992-6243; email: sundararajan.srinivasan@gmail.com).

[3] T. Ma is with Siri at Apple Inc., 2 Infinite Loop, mailstop 302-4APP, Cupertino, California 95014, USA (phone: 408-643-5909; email: tma@apple.com).

[4] G. Lazarou is with The New York City Transit Authority, 30-74 38th Street, Apt 1A, Astoria, New York, New York, USA 11103 (phone: (662) 617-2064; email: glaz@ieee.org).

[5] J. Picone is with the Department of Electrical and Computer Engineering at Temple University, 1947 North 12th Street, Philadelphia, Pennsylvania 19027 USA (phone: 215-204-4841; fax: 215-204-5960; email: joseph.picone@isip.piconepress.com).

# A NONLINEAR AUTOREGRESSIVE MODEL FOR SPEAKER VERFICATION

*Abstract*— Gaussian Mixture Models (GMM) have been the most popular approach in speaker recognition and verification for over two decades. The inefficiencies of this model for signals such as speech are well documented and include an inability to model temporal dependencies that result from nonlinearities in the speech signal. The resulting models are often complex and overdetermined, which leads to a lack of generalization. In this paper, we present a nonlinear mixture autoregressive model (MixAR) that attempts to directly model nonlinearities in the trajectories of the speech features. We apply this model to the problem of speaker verification. Experiments with synthetic data demonstrate the viability of the model. Evaluations on standard speech databases, including TIMIT, NTIMIT, and NIST-2001, demonstrate that MixAR, using only half the number of parameters and only static features, can achieve a lower equal error rate when compared to GMMs, particularly in the presence of previously unseen noise. Performance as a function of the duration of both the training and evaluation utterances is also analyzed.

***Keywords*—** Gaussian mixture models, mixture autoregressive model, nonlinear statistical models, speaker verification

## I. INTRODUCTION

The goal in speaker verification is to accept or reject an identity claim made by a speaker. This biometric is widely used in a variety of applications ranging from secured access and surveillance to multimodal verification. A challenge for statistical modeling in speaker verification is to accurately and efficiently represent the probability distribution of speaker features so that even similar sounding speakers can be distinguished. The majority of speaker recognition and verification systems today utilize Gaussian Mixture Models (GMMs) either entirely or as part of a hybrid model (Beigi, 2011).

There are two well-known drawbacks of the GMM model. The first involves statistical independence – there are obviously dependencies in the temporal evolution of both the static features and the derivatives of these features. Constructing a GMM from standard features decorrelated using only a diagonal covariance matrix does not adequately model these dependencies. Use of full covariance matrices results in models with an extremely large number of parameters and creates parameter estimation problems. Performance improvements with full covariance approaches (or constrained versions of these large matrices) have been minimal, and often increase the system’s sensitivity to mismatched channel conditions. A major overarching goal in our work is to improve performance when the training and evaluation conditions are mismatched, and the evaluation data contains previously unseen noise and channel conditions.

The second more serious drawback, which is the focus of this work, is the implicit assumption of linearity in the feature vector dynamics. The derivatives of the cepstral features, commonly used in speech processing as part of the MFCC feature vector representation (Chen & Bilmes, 2007) are only a linear approximation of the actual dynamics of the static features. However, a survey of studies on the subject shows that the speech signal contains significant nonlinear information, and using only derivative features to represent speech dynamics with GMM modeling is tantamount to discarding any nonlinear information present in the signal (May, 2008; Kokkinos & Maragos, 2005).

An obvious solution to this problem is to add features that can represent the nonlinear dynamic information. However, adding nonlinear invariants as features has not improved the robustness of recognition and verification technologies in harsh or mismatched environments. The reasons for these failures include (1) it is difficult to estimate invariants reliably from speech, resulting in parameter estimation algorithms that need to be extensively tuned; (2) these estimation algorithms typically require an acoustic event to have a long duration (Petry et al., 2002), and this gravely undermines the applicability of invariant features for a short-term stationary signal like speech; and (3) invariants only quantify the degree of nonlinearity and do not characterize the nature of the dynamics completely.

The primary goal of this work is to approach the information representation problem at the acoustic modeling level using a nonlinear mixture autoregressive model (MixAR) (Zeevi et al., 2000), thereby accounting for the nonlinear dynamics of speech in the base model and minimizing the dimensionality of the feature space. This model is shown in Figure 1. Previous work on mixture autoregressive modeling for speech has been in the context of hidden Markov models for speech recognition (Juang & Rabiner, 1985). A more recent investigation of AR-HMMs (Ephraim & Roberts, 2005) used a switching autoregressive process to capture signal correlations during state transitions. Another model considered speech features as a GMM white noise process filtered through an autoregressive signal for speaker identification (Ayadi, 2008). Results on speech recognition showed that at best these models were only comparable to an MFCC-based HMM using a GMM observation model.

A more sophisticated model (Wong & Li, 2000) considers a mixture of autoregressive filters (MAR) for the observation model. Our earlier work (Srinivasan et al., 2008) considered this model for phone classification. MixAR is a generalization of MAR, where the mixture weights are allowed to be time-varying and data-dependent. In this work, we apply the MixAR model to feature vectors in a speaker verification task, and demonstrate improved performance over the more limited MAR model.

The rest of the paper is organized as follows: Section II formally defines the MixAR model, explains some of the relevant properties of this model and discusses the parameter estimation problem. Results of experiments using synthetic data are included in Section III and speaker verification experiments with real speech data are presented in Section IV. Experiments documenting variation in performance with the duration of training and evaluation utterances are also discussed in Section IV. Finally, in Section VVI we present our conclusions and discuss future directions for this research.

## II. THE MIXAR MODEL

A mixture autoregressive process (MixAR) of order *p* with *m* components, *X*={*x*[*n*]}, is defined as (Zeevi et al., 2000):

$$x[n] = \begin{cases} a_{1,0} + \sum_{i=1}^{p} a_{1,i} x[n-i] + e_{1}[n] & w.p. \quad W_{1}(x[n-1]) \\ a_{2,0} + \sum_{i=1}^{p} a_{2,i} x[n-i] + e_{2}[n] & w.p. \quad W_{2}(x[n-1]) \\ \vdots & \\ a_{m,0} + \sum_{i=1}^{p} a_{m,i} x[n-i] + e_{m}[n] & w.p. \quad W_{m}(x[n-1]) \end{cases} \qquad (1)$$

where $\varepsilon_{\mathrm{i}}$ is a zero-mean Gaussian random process with a variance of $s_{j}^{2}$, “w.p.” denotes “with probability” and the gating weights, $W_{\mathrm{i}}$, sum to 1. The linear prediction coefficients, $\{a_{\mathrm{i}}\}$, represent the dynamic model, where $a_{\mathrm{i},0}$ are the component means, while $w_i$ and $g_i$ are called gating coefficients. It is apparent that an *m*-mixture MixAR process is the weighted sum of *m* Gaussian autoregressive processes, with the time-dependent weights depending on previous data and the gating coefficients.

One insightful way of viewing this model is as a process in which each data sample at any one point in time is generated from one of the component AR mixture processes chosen randomly according to its weight $W_{\mathrm{i}}$, as depicted in Figure 1. One property of MixAR that is of particular relevance here is the ability of MixAR to model nonlinear time series (Zeevi et al., 2000; Wong & Li, 2000). Though the individual component AR processes are linear, the probabilistic mixing of these AR processes constitutes a nonlinear model. Even when the mixture weights are fixed, the model reduces to MAR, which is still nonlinear. The addition of a gating system layer for weight generation increases the flexibility of the model even further, allowing us to model distributions as a function of past data.

Several other properties of MixAR, including a mathematically rigorous proof of the asymptotic performance of a MixAR model for stochastic processes, are derived in (Zeevi et al., 2000). Note that in the original formulation, both the gate and prediction orders were constrained to be equal. There are some practical implementation issues regarding parameter estimation for this model, and these are discussed next. In this paper, we restrict ourselves to MixAR models of order one to avoid some difficulties with parameter estimation.

### *A. Estimation of the Prediction and Variance Parameters*

Similar to the well-known training procedure for GMM, maximum likelihood estimates for MixAR prediction and variance parameters can be calculated using the Expectation-Maximization (EM) algorithm (Dempster et al., 1977). Given an order, *p*, the parameter set for each of the *m* components of a MAR model consists of *p*+1 predictor coefficients (including the mean), the error variance, and mixing weight, and is denoted as:

$$\theta_l = \{a_{l,0}, a_{l,1}, \cdots, a_{l,p}, \sigma_l, w_l, g_l\}, \qquad l = 1,\ 2,\ \ldots,\ m. \tag{2}$$

To estimate these parameters, we first need an initial guess for these parameters and then we iterate with EM to successively refine the estimates.

An initialization strategy that we found to work reasonably well was to first train a GMM with the same number of mixtures and then set each component of the MixAR model to have the same mean, variance and weight as the GMM model. We initialize the predictor coefficients and the data-dependency gating coefficients, $\{A_i\}$, to zero. These initial parameters can be then refined recursively using an E-step (Zeevi et al., 2000; Wong & Li, 2000):

$$\gamma_l[n] = \frac{W_l p_l(x[n]\,|\,\theta)}{\sum_{k=1}^{m} W_k\, p_k(x[n]\,|\,\theta)}\ , \tag{3}$$

where

$$p_l(x[n]\,|\,\theta) \propto \frac{1}{\sigma_l} e^{\frac{-1}{2\sigma_l}\left(x[n] - a_{l,0} - \sum_{i=1}^{m} a_{l,i}\, x[n-i]\right)^2} \tag{4}$$

represents the probability that a sample was generated from component *l* at time instant *n*.

The corresponding M-step is given by:

$$\hat{A}_l = R_l^{-1} r_l \tag{5}$$

$$\hat{\sigma}_l^2 = \frac{\sum_{n=p+1}^{N} \gamma_l[n]\left(x[n] - \hat{a}_{l,0} - \sum_{i=1}^{m} \hat{a}_{l,i}\, x[n-i]\right)^2}{\sum_{n=p+1}^{N} \gamma_l[n]} \tag{6}$$

$$R_l = \sum_{n=p+1}^{N} \gamma_l[n] X_{n-1} X_{n-1}^T \tag{7}$$

$$r_l = \sum_{n=p+1}^{N} \gamma_l[n] X_{n-1}\, x[n] \tag{8}$$

$$X_{n-1} = \begin{bmatrix} 1 \\ x[n-1] \\ x[n-2] \\ \vdots \\ x[n-p] \end{bmatrix}. \tag{9}$$

*B. Estimation of the Gating Coefficients*

A complication arises with respect to the estimation of gating coefficients for MixAR. There is no closed-form solution for these, and hence a Newton gradient-ascent approach must be used:

$$\hat{w}_l = w_l + \beta \frac{\Delta Q}{\Delta w_l} \tag{10}$$

$$\hat{g}_l = g_l + \beta \frac{\Delta Q}{\Delta g_l}\ , \tag{11}$$

where $Q$ denotes the log-likelihood of the MixAR model for the training data, and β and Δ are design parameters to be chosen empirically. The expression for computing Q is:

$$Q(\theta) = \sum_{n=1}^{N} \sum_{l=1}^{m} \gamma_l[n] \log(W_l[n]) + \sum_{n=1}^{N} \sum_{l=1}^{m} \gamma_l[n] \log(p_l(x[n] \mid \theta)\ . \tag{12}$$

Due to this complication in the updates for the gate coefficients, the training procedure outlined above is not in the realm of strict EM algorithm but falls under a class of algorithms known as generalized EM algorithms (GEM) (McLachlan et al., 2008). For both EM and GEM algorithms, the E-step is similar. However, while an EM algorithm actually maximizes the expectation during each M-step, a GEM algorithm only guarantees that parameters that the model likelihood for the data is increased but does not guarantee that it is maximized at each M-step. This has the potential to increase the number of iterations required for training compared to a pure EM algorithm.

In a process analogous to the choice of an adaptation constant in adaptive filter theory, we can postulate that quick and smooth convergence of the GEM algorithm can be achieved by starting with a relatively high value for β and then reducing this value with successive iterations. In our experiments, we found that fixing Δ = 0.01 and running 10 iterations each with β = 0.9, β = 0.5, and β = 0.2 in succession provided a reasonably smooth and quick convergence. However, such convergence is not guaranteed in general and this poses a generalization problem for wider application of this model.

Fortunately, we can do better than guessing an appropriate value for β. We can use the secant method for root-finding and maximization (Dennis & Schnabel, 1996). The iterative formula for finding the maximum using Newton's method is:

$$\hat{x} = x + f'(x) \big/ f''(x) \ . \tag{13}$$

In the secant method, the double derivative in the denominator is estimated numerically using the secant at the point. We estimate the scaling factor β as the inverse of double derivative of the log-likelihood with respect to the gate parameters:

$$b = 1/\left( D^2 Q \big/ D^2 w_l \right). \tag{14}$$

During implementation, this scheme amounts to finding the value of Q at three different points (Q($w_l$), Q($w_l$ +Δ), Q($w_l$ - Δ)) for each gate coefficient $w_l$ and then using the following update equation:

$$\hat{w}_l = w_l + \frac{Q(w_l + D) - Q(w_l + D)}{Q(w_l + D) + Q(w_l - D) - 2Q(w_l)} \ . \tag{15}$$

Similarly, the update equation for gate coefficients $g_l$ is:

$$\hat{g}_l = g_l + \frac{Q(g_l + D) - Q(g_l + D)}{Q(g_l + D) + Q(g_l - D) - 2Q(g_l)} \ . \tag{16}$$

Using this method, we obtain reasonable convergence as shown in Figure 2. We have used three GEM iterations in the experiments described in the following sections.

## III. SYNTHETIC DATA EXPERIMENTS

In the experiments described in the next two sections, we used the ISIP public domain speech recognition software (Huang & Picone, 2002) to implement the MixAR model as well as integrate it into an existing speaker verification system.

*A. Two-Way Classification of Synthetic Speech-Like Data*

In this section we describe a set of pilot experiments designed to validate the basic properties of the MixAR model. We first selected two speakers from the 2001 NIST SRE Corpus (Greenberg & Martin, 2009) and generated the static features corresponding to a standard MFCC feature vector (12 mel-ceptral coefficients). A 3-state HMM and a MixAR model, both with 4 Gaussian mixture components per state, were trained for each speaker. For each class (e.g., a speaker), two speech-like signals of 40,000 vectors were generated from these models – a linear speech-like signal (***X**<sub></sub>*$\boldsymbol{X_1}$) was synthesized from the HMM model, and a nonlinear speech-like signal ($\boldsymbol{X_2}$) was generated from the MixAR model. To simulate a range of signals with varying degrees of nonlinearity, the two signals were mixed with a mixing coefficient α:

$$X_{\alpha} = (1 - \alpha)X_1 + \alpha X_2 \ . \tag{17}$$

The first 20,000 vectors from each $\boldsymbol{X_\alpha}$ were used as a training set while the remaining vectors were split into 200 segments of 100 vectors each for evaluation.

The results of the classification experiments with this data are shown in Table 1. Since the data contains nonlinearities, we compared MixAR using static features to a GMM using static features plus first derivatives (static+Δ). Performance for GMMs with only static features was significantly worse except for the case of a purely linear signal (*α* = 0). We also allocated 8 mixture components for the GMM system since its feature vectors had additional components. The MixAR system had only 4 mixture components. Overall, the MixAR system had approximately 50% fewer parameters than the GMM system. We can see that when the amount of nonlinearity is insignificant, GMM performs as well as MixAR. However, as the amount of nonlinearity in the signal increases, MixAR performs significantly better. These results validated the basic model and provided motivation to do further testing on more realistic data.

### *B. Speaker Verification With Synthetic Data*

Since our goal is to study speaker verification, we next used the development database from the 1-speaker detection task of the 2001 NIST SRE Corpus (Greenberg & Martin, 2009). Though the use of development database is not standard practice in most published baselines, it is small enough to allow us to quickly generate results using only modest computing power and yet large enough to provide a reliable estimate of the performance. All 60 speakers in the training set were used. Each training utterance was about 2 minutes long. Static (12 MFCCs + energy), delta (26 MFCCs) and delta-delta (39 MFCCs) features were extracted.

Two types of clean data were synthesized. For the first type, a 10-state HMM with 4-Gaussians per state was trained for each utterance. For the second type, a 32-mixture MixAR model of prediction order 1 was trained for each utterance. For each of the models trained, new training data of about 30,000 frames per speaker and evaluation data of 20 utterances with about 200 frames for each utterance per speaker were generated according to that model.

Similarly, two types of noisy data were generated. First, the clean training utterances from the development data were corrupted with car noise to achieve an SNR of 5 dB. This approach followed a methodology previously used to generate the Aurora database (Parihar et al., 2004). The remaining steps to yield the two types of noisy data were the same as those for the clean case. The goal of creating data in this way was to simulate 4 different test conditions: clean+linear, clean+nonlinear, noisy+linear and noisy+nonlinear.

Using the synthetic training data, both GMMs and prediction order-1 MixAR models were trained for each speaker under each condition. Next, the corresponding synthesized evaluation data were used for evaluating speaker verification performance. For the clean case, there was little difference in performance between GMM and MixAR. For noisy evaluation data at 5 dB SNR, there was not much variation in performance between GMM and MixAR for HMM-generated data.

However, for the data generated from the nonlinear MixAR model and with the addition of noise, the MixAR model showed a significant improvement in performance using far fewer parameters. This is evident from the DET plot shown in Figure 3. These results provide support to the hypothesis that when there are significant nonlinearities in the signal, using this information makes the nonlinear model much more robust in the presence of noise.

## IV. SPEAKER VERIFICATION EXPERIMENTS

### *A. Evaluation Using the NIST-2001 Development Database*

We applied the MixAR model to the 1-speaker detection task in the 2001 NIST SRE Corpus. All 60 speakers in the development set were used for training and all 78 utterances were used for evaluation. Each training utterance was about 2 minutes long, while the durations of the test utterances varied but did not exceed 60 seconds. A standard 39-dimensional MFCC feature vector was used.

Performance was evaluated with and without delta features and energy for a fixed number of mixtures. The results are tabulated in Table 2. For GMM, a substantial improvement is obtained using the delta features and marginal improvements were obtained using delta-delta features. For MixAR, the use of any delta features provided no measurable improvements. This clearly indicates that MixAR can extract all necessary information from only the static features.

MixAR and GMM performance was then evaluated as a function of the number mixtures. The detection error trade-off (DET) curves are shown in Figure 4. The EER results are shown in Table 3. Also indicated in parenthesis is the number of parameters for each case. From this table it is clear that MixAR can achieve about the same performance using 2x fewer parameters than GMM. This reduction in the number of parameters points to the efficiency of MixAR in capturing the dynamic information.

Moreover, even when considering the best-case scenario for GMM with a large number of parameters (8 mixtures with static as well as delta and delta-delta coefficients), there is a 10.6% relative reduction in EER with MixAR. These results appear to strongly indicate that there is nonlinear evolution information in speech features that the GMM model cannot capture using linear derivatives alone and that MixAR can effectively employ this information for achieving better speaker verification.

### *B. Evaluation on Unseen Noise Conditions*

To evaluate the robustness of MixAR compared to GMM on unseen noise conditions, several noise conditions were simulated with the TIMIT database (Garofolo et al., 1993) by adding synthesized noise from three different noise sources: white, car, and babble. Three SNR levels were used: 10, 5 and 0 dB (in addition to the clean set). The core test partition of the database containing 168 speakers was used. The three types of noise sources were chosen to represent commonly occurring types of noise.

The matrix of experimental results is shown in Table 4. From this table, it is clear that while unseen noise conditions degrades performance for both models, MixAR performs relatively better than GMM and also uses 2.5x fewer parameters. The DET curves shown for the different noise conditions in Figure 5 also support the conclusion that MixAR performs better than GMM.

A related problem to unseen noise conditions is variation in acoustic channel. NTIMIT is a database that was created by transmitting TIMIT utterances over different analog telephone channels (Jankowski et al., 1990). We studied speaker verification performance on NTIMIT by splitting the data for each speaker into 8 utterances for training and the remaining 2 utterances for evaluation. The DET performance curves for the 8-mixture MixAR using only static MFCCs (with 480 parameters) and for the 16-mixture GMM (with 1168 parameters) using both static and delta features are shown in Figure 6. The corresponding EERs are shown in Table 5. From this analysis it is clear that MixAR using 2.5x fewer parameters achieves the same or a higher level of performance as a GMM.

### *C. Effect of Utterance Duration on Speaker Verification Performance*

Even if MixAR could do better under the conditions we have tested so far, it is possible that MixAR requires more training data than GMM for reliable parameter estimation. This could be a particular concern considering that MixAR attempts to learn nonlinear dynamic information, and nonlinear dynamics are notoriously difficult to characterize from short lengths of data. For example, it is known that estimates of Lyapunov exponents can be unreliable when the length of data is short (Banbrook et al., 1997). One particular concern with insufficient training data is the problem of overfitting. It is therefore necessary to study performance as a function of the amount of training data.

Towards this end, we conducted experiments with varying training utterance durations keeping the evaluation utterance duration a constant. Utterances corresponding to five durations – 120, 90, 60, 30 and

15s – were extracted from training data for each of the 60 speakers from the training part of the NIST 2001 development database. All evaluation data for the 78 speakers with durations ranging mostly between 20 and 40s were used. The NIST-2001 database is particularly suited here because the training data is clean and the evaluation data is corrupted by different kinds of noise. This means that models that are overtrained will perform poorly on the evaluation utterances.

The number of mixtures for MixAR was fixed at 8. The number of mixtures for GMM was also fixed at 8 to alleviate the problem of overfitting. The results of the experiment are reported in Table 6. There is a 43.9% increase in EER for GMM when the training utterance duration is reduced from 120s to 15s. On the other hand, the corresponding increase in EER for MixAR is only 26.56%. Thus, MixAR does not necessarily require longer training utterance durations than GMM. Utterance duration is an important practical constraint on voice biometrics.

It is also well known that increasing the evaluation utterance duration improves speaker verification performance. We tested this hypothesis by examining evaluation utterances of five different durations – 30, 15, 10, 5, and 3s. These were extracted from each of the 78 test utterances in the NIST-2001 development database. All training data from all 60 speakers were used. The results of this experiment are reported in Table 7. For GMM, there is an increase in EER of 31.2% as the evaluation utterance duration reduces from 30s to 3s. The corresponding reduction for MixAR is 33.3%. Hence, both systems are equally sensitive to the duration of the evaluation data.

## V. SUMMARY

In this work, we have applied a nonlinear mixture autoregressive model (MixAR) to several speaker verification tasks. Our experiments with synthetic as well as real speech data show that the MixAR model outperforms GMM under several noise conditions, particularly the case where the type of noise in the evaluation data was not observed in the training database. Equally important, MixAR did not require delta features and used 2.5x fewer parameters to achieve comparable or better performance as that of GMM. The dynamic modeling capability of MixAR is effective at capturing and exploiting speech dynamics better than GMM.

This work is part of our ongoing interest in exploiting nonlinear statistical models in speech recognition. It has only begun to explore the potential of the nonlinear MixAR model in speaker verification. Most modern systems incorporate more advanced adapted GMMs (Reynolds & Campbell, 2008) or discriminative model approaches (Li & Kinnunen, 2010). Before an unbiased comparison can be made to these systems, a framework for integrating these approaches into MixAR model must be developed. We also plan to apply the MixAR model to large vocabulary speech recognition tasks.

## VI. REFERENCES

Ayadi, M. (2008). *Autoregressive models for text independent speaker identification in noisy environments*. University of Waterloo.

Banbrook, M., Ushaw, G., & McLaughlin, S. (1997). How to extract Lyapunov exponents from short and noisy time series. *IEEE Transactions on Signal Processing*, *45*(5), 1378–1382.

Beigi, H. (2011). *Fundamentals of Speaker Recognition* (p. 942). Upper Saddle River, New Jersey, USA: Springer.

Chen, C.-P., & Bilmes, J. A. (2007). MVA Processing of Speech Features. *IEEE Transactions on Audio, Speech and Language Processing*, 15(1), 257–270.

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, *39*(1), 1–38.

Dennis, J., & Schnabel, R. (1996). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* (p. 394). Englewood Cliffs, New Jersey, USA: Prentice Hall.

Ephraim, Y., & Roberts, W. (2005). Revisiting autoregressive hidden Markov modeling of speech signals. *IEEE Signal Processing Letters*, *12*(2), 166–169.

Garofolo, J., Lamel, L., Fisher, W., Fiscus, J., Pallet, D., Dahlgren, N., & Zue, V. (1993). TIMIT Acoustic-Phonetic Continuous Speech Corpus. *The Linguistic Data Consortium Catalog*. Philadelphia, Pennsylvania, USA: The Linguistic Data Consortium.

Greenberg, C. S., & Martin, A. F. (2009). NIST speaker recognition evaluations 1996-2008. *Proceedings of SPIE (Stereoscopic Displays and Applications XX)* (p. 732411). Orlando, Florida, USA.

Huang, K., & Picone, J. (2002). Internet-Accessible Speech Recognition Technology. *Proceedings of the IEEE Midwest Symposium on Circuits and Systems* (pp. III–73 – III–76). Tulsa, Oklahoma, USA.

Jankowski, C., Kalyanswamy, A., Basson, S., & Spitz, J. (1990). NTIMIT: a phonetically balanced, continuous speech, telephone bandwidth speech database. *IEEE International Conference on Acoustics Speech and Signal Processing* (pp. 109–112 vol.1). Albuquerque, New Mexico, USA.

Juang, B.-H., & Rabiner, L. (1985). Mixture autoregressive hidden Markov models for speech signals. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *33*(6), 1404–1413.

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

Kokkinos, I., & Maragos, P. (2005). Nonlinear speech analysis using models for chaotic systems. *IEEE Transactions on Speech and Audio Processing*, *13*(6), 1098–1109.

McLachlan, Geoffrey, & Thriyambakam, K. (2008). *The EM Algorithm and Extensions* (p. 400). Hoboken, New Jersey, USA: Wiley-Interscience.

May, D. (2008). *Nonlinear dynamic invariants for continuous speech recognition*. Mississippi State University.

Parihar, N., Picone, J., Pearce, D., & Hirsch, H.-G. (2004). Performance Analysis of the Aurora Large Vocabulary Baseline System. *Proceedings of the European Signal Processing Conference* (pp. 553–556). Vienna, Austria.

Petry, A., Augusto, D., & Barone, C. (2002). Speaker Identification using Nonlinear Dynamical Features. *Chaos, Solitons and Fractals*, *13*(2), 221–231.

Reynolds, D., & Campbell, W. (2008). Text-Independent Speaker Recognition. *Springer Handbook of Speech Processing* (1st ed., p. 1176). Berlin, Germany: Springer.

Srinivasan, S., Ma, T., May, D., Lazarou, G., & Picone, J. (2008). Nonlinear Mixture Autoregressive Hidden Markov Models For Speech Recognition. *Proceedings of the International Conference on Spoken Language Processing* (pp. 960–963). Brisbane, Australia.

Zeevi, A., Meir, R., & Adler, R. (2000). *Nonlinear Models for Time Series using Mixtures of Autoregressive Models* (p. 25). Haifa, Israel. Retrieved from http://ie.technion.ac.il/~radler/mixar.pdf.

Wong, C. S., & Li, W. K. (2000). On a Mixture Autoregressive Model. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, *62*(1), 95–115.

## VII. LIST OF FIGURES

# VIII. FIGURES

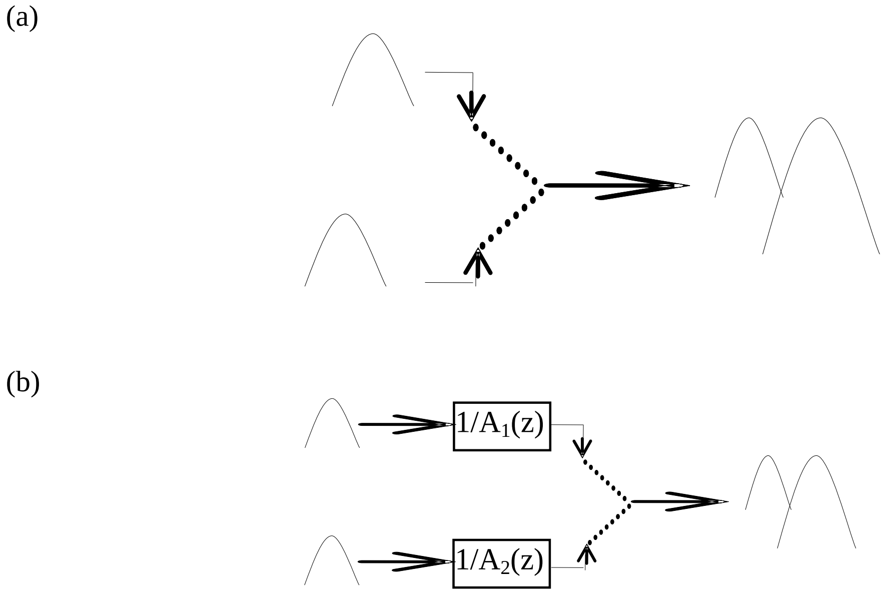

Figure 1. An overview of the (a) GMM and (b) MixAR approaches. The MixAR model is a weighted sum of Gaussian autoregressive models with time-dependent weights.

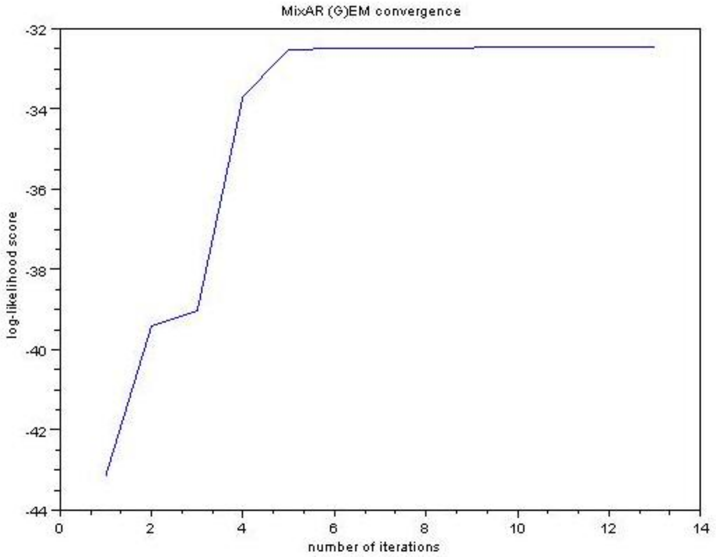

Figure 2. Performance of (Generalized) EM using the secant method as a function of the number of iterations for an 8-mixture MixAR model is shown (speaker *4516* from the NIST-2001 database).

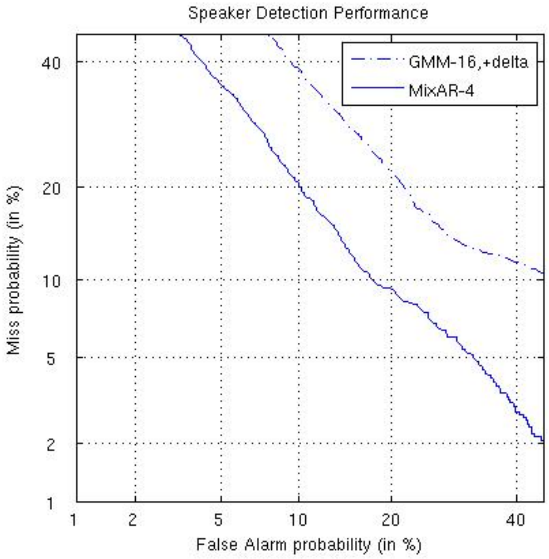

Figure 3. DET curves are shown for a simulated speaker verification task. MixAR performance in the presence of noise exceeds GMM performance.

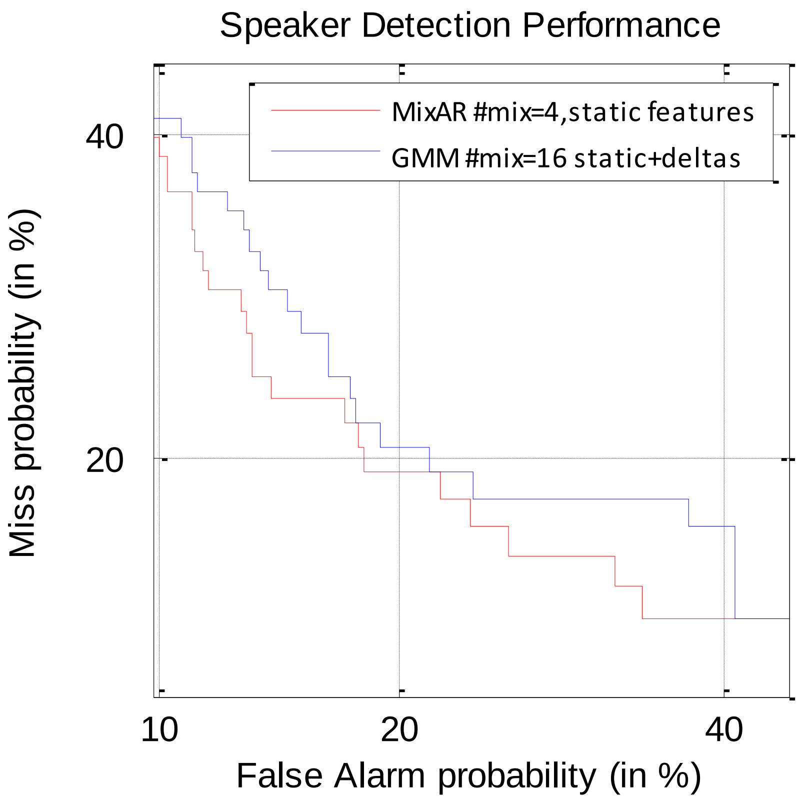

Figure 4. A DET curve is shown for a 1-speaker detection task based on the NIST-2001 development database. MixAR with 4 mixture components and only static features performs better than a GMM with 16 mixture components and static+delta features.

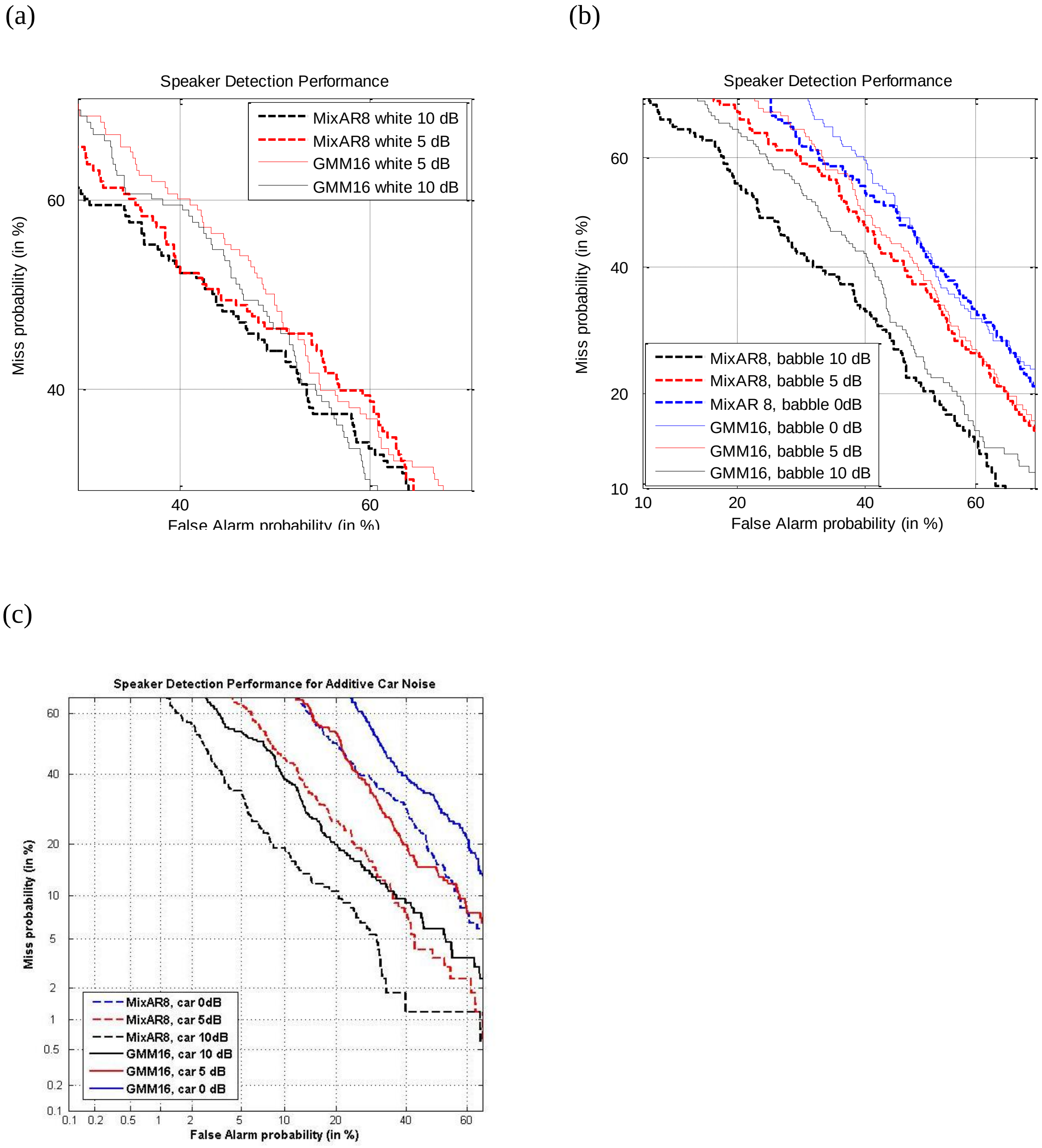

Figure 5. DET curves for GMM and MixAR models are shown for noisy TIMIT data with three types of additive noise: a) white, b) babble and c) car noise. A variety of SNRs are used.

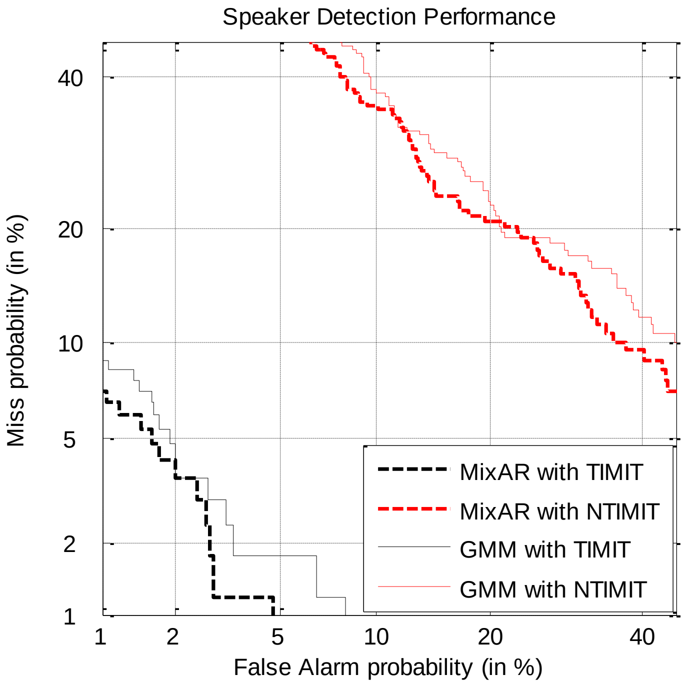

Figure 6. DET curves for GMM and MixAR models on TIMIT and NTIMIT are shown. MixAR performance exceeds GMM performance while using a fewer number of parameters.

## IX. LIST OF TABLES

Table 1. A comparison of classification error rates is shown for a GMM system using static+delta features and a MixAR system operating only on the static features is shown on synthetic data. The number of parameters for each system is shown in parentheses. The GMM system, which uses static and delta features, performs significantly worse than the MixAR system as the nonlinearity in the data increases.

Table 2. Speaker verification EERs are shown for MixAR and GMM for a variety of feature vector combinations. MixAR does not need delta features since the model itself encodes temporal dynamics.

Table 3. EERs are shown as a function of the number of mixtures. MixAR performs slightly better with almost half the number of parameters.

Table 4. EERs are shown for a variety of noise conditions.

Table 5. EERs are shown for TIMIT (clean data) and NTIMIT (noisy data).

Table 6. Performance is analyzed as a function of the duration of the training data utterances. The evaluation utterance durations were held constant and varied between 20 and 40 seconds.

Table 7. Performance is analyzed as a function of the duration of the evaluation data is shown. The training utterance duration was fixed and averaged around 120s.

## X. TABLES

| *α* | GMM (8 mix.)<br>Static+Δ | MixAR (4-mix.)<br>Static |
|---|---|---|
| 0.00 | 1.50 (288) | 1.50 (240) |
| 0.25 | 3.25 (576) | 3.50 (240) |
| 0.50 | 10.25 (576) | 6.25 (240) |
| 0.75 | 24.75 (576) | 9.75 (240) |
| 1.00 | 26.75 (576) | 13.75 (240) |

Table 1. A comparison of classification error rates is shown for a GMM system using static+delta features and a MixAR system operating only on the static features is shown on synthetic data. The number of parameters for each system is shown in parentheses. The GMM system, which uses static and delta features, performs significantly worse than the MixAR system as the nonlinearity in the data increases.

| Features | GMM<br>16-mix. | MixAR<br>8-mix. |
|---|---|---|
| Static(12) | 22.1 | 19.1 |
| Static+E(13) | 33.1 | 41.1 |
| Static+Δ(24) | 20.6 | 20.4 |
| Static+Δ+ΔΔ(36) | 20.5 | 20.5 |

Table 3. Speaker verification EERs are shown for MixAR and GMM for a variety of feature vector combinations. MixAR does not need delta features since the model itself encodes temporal dynamics.

| No.<br>Mixtures | GMM<br>Static+Δ+ΔΔ | MixAR<br>Static Only |
|---|---|---|
| 2 | 23.1 (216) | 24.1(120) |
| 4 | 21.7 (432) | 19.2(240) |
| 8 | 20.5 (864) | 19.1(480) |
| 16 | 20.5 (1728) | 19.2(960) |

Table 2. EERs are shown as a function of the number of mixtures. MixAR performs slightly better with almost half the number of parameters.

| | SNR<br>(dB) | Clean | Car<br>Noise | White<br>Noise | Babble<br>Noise |
|---|---|---|---|---|---|
| **GMM<br>(1168)** | | **2.4** | | | |
| | 10 dB | | 19.7 | 48.7 | 40.6 |
| | 5 dB | | 31.2 | 50.0 | 44.7 |
| | 0 dB | | 39.3 | 49.8 | 48.2 |
| **MixAR<br>(480)** | | **1.8** | | | |
| | 10 dB | | 13.7 | 47.0 | 36.9 |
| | 5 dB | | 23.2 | 47.6 | 42.8 |
| | 0dB | | 33.9 | 48.5 | 47.6 |

Table 4. EERs are shown for a variety of noise conditions.

| Database | GMM<br>Static+Δ+ΔΔ<br>(1728) | MixAR<br>Static Only<br>(480) |
|---|---|---|
| TIMIT | 2.4 | 1.8 |
| NTIMIT | 21.0 | 20.9 |

Table 6. EERs are shown for TIMIT (clean data) and NTIMIT (noisy data).

|  | Training Utterance Duration | EER |
|---|---|---|
| GMM (864) | 120 | 20.5 |
|  | 90 | 20.4 |
|  | 60 | 20.4 |
|  | 30 | 24.4 |
|  | 15 | 29.5 |
| MixAR (480) | 120 | 19.2 |
|  | 90 | 21.5 |
|  | 60 | 21.8 |
|  | 30 | 21.8 |
|  | 15 | 24.3 |

Table 5. Performance is analyzed as a function of the duration of the training data utterances. The evaluation utterance durations were held constant and varied between 20 and 40 seconds.

|  | Evaluation Utterance Duration | EER |
|---|---|---|
| GMM (864) | 30 | 20.5 |
|  | 15 | 21.8 |
|  | 10 | 21.5 |
|  | 5 | 24.4 |
|  | 3 | 26.9 |
| MixAR (480) | 30 | 19.2 |
|  | 15 | 23.4 |
|  | 10 | 23.1 |
|  | 5 | 25.6 |
|  | 3 | 25.6 |

Table 7. Performance is analyzed as a function of the duration of the evaluation data is shown. The training utterance duration was fixed and averaged around 120s.

## LIST OF FIGURES

# I. FIGURES

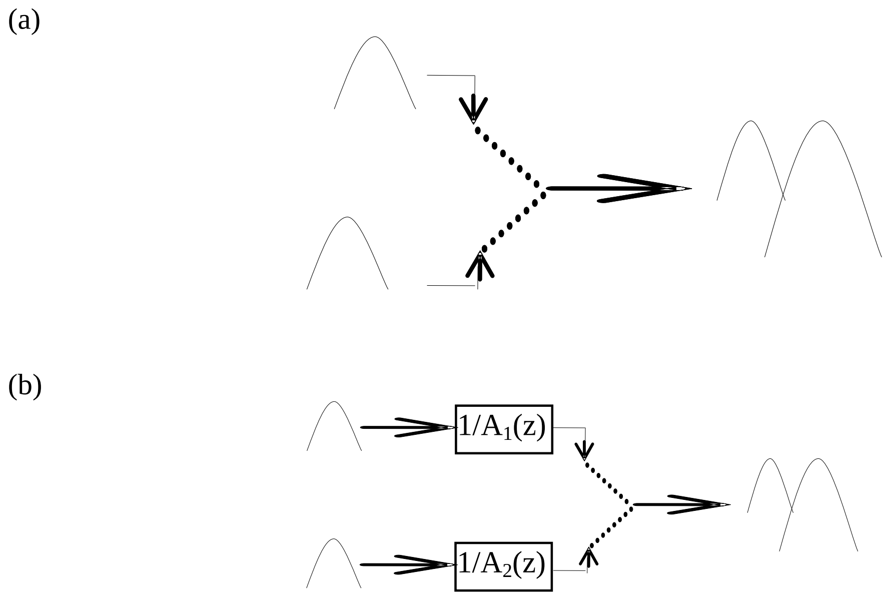

Figure 1. An overview of the (a) GMM and (b) MixAR approaches. The MixAR model is a weighted sum of Gaussian autoregressive models with time-dependent weights.

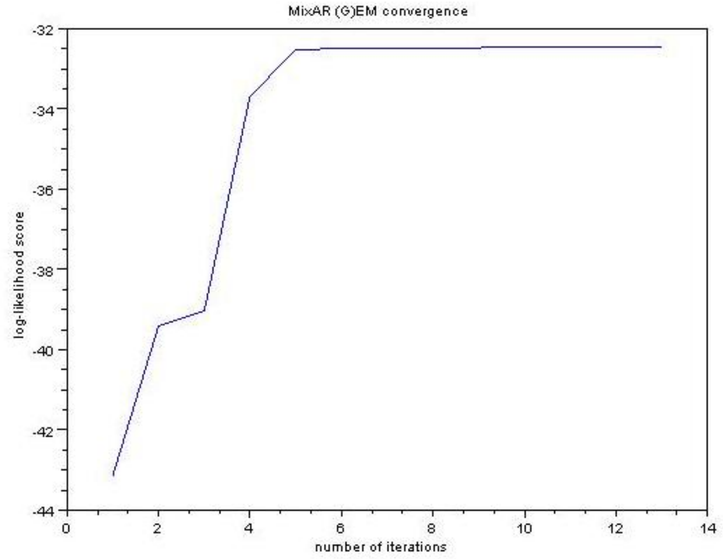

Figure 2. Performance of (Generalized) EM using the secant method as a function of the number of iterations for an 8-mixture MixAR model is shown (speaker *4516* from the NIST-2001 database).

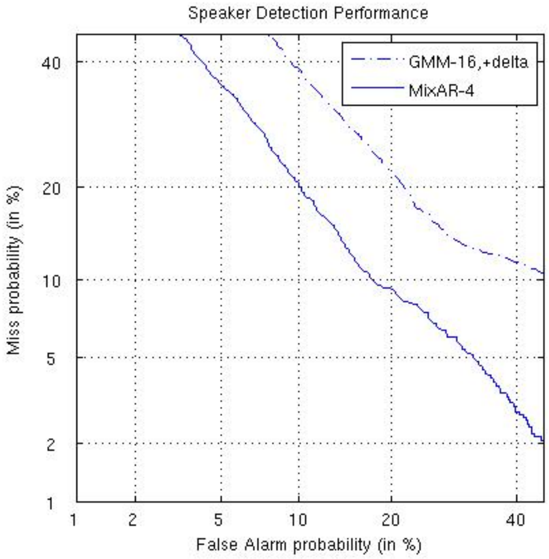

Figure 3. DET curves are shown for a simulated speaker verification task. MixAR performance in the presence of noise exceeds GMM performance.

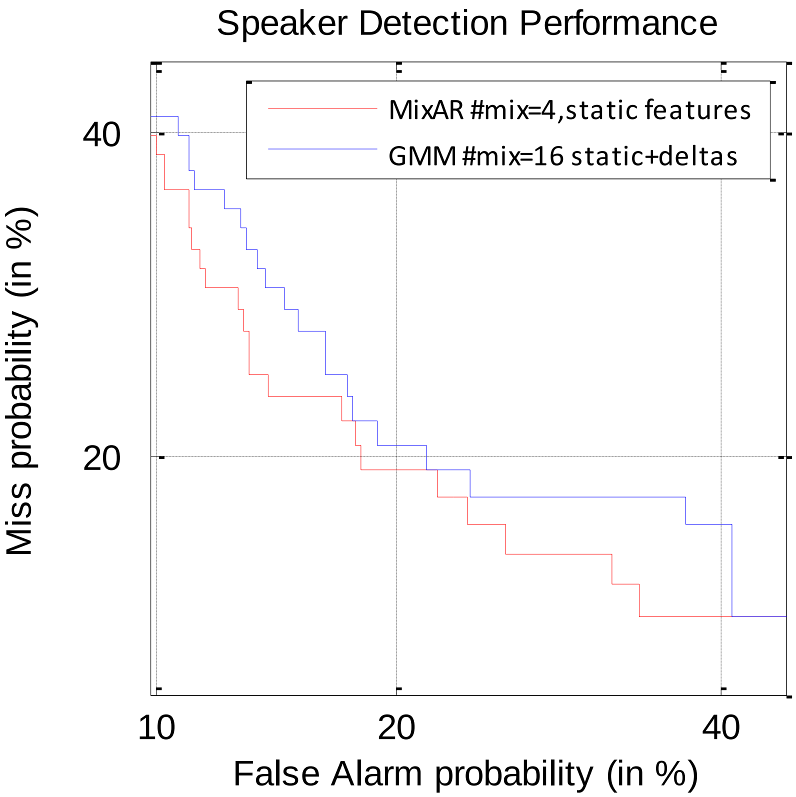

Figure 4. A DET curve is shown for a 1-speaker detection task based on the NIST-2001 development database. MixAR with 4 mixture components and only static features performs better than a GMM with 16 mixture components and static+delta features.

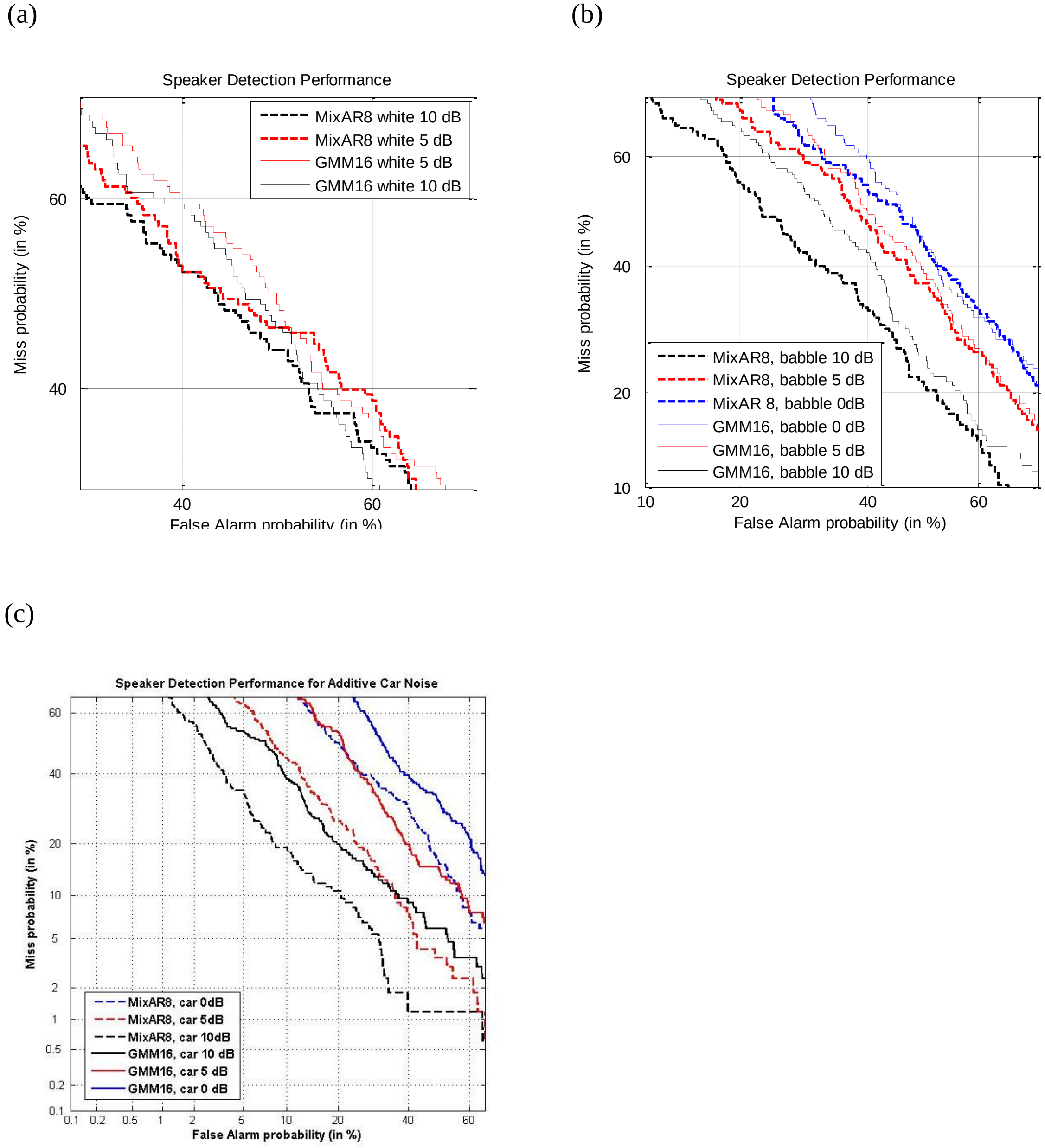

Figure 5. DET curves for GMM and MixAR models are shown for noisy TIMIT data with three types of additive noise: a) white, b) babble and c) car noise. A variety of SNRs are used.

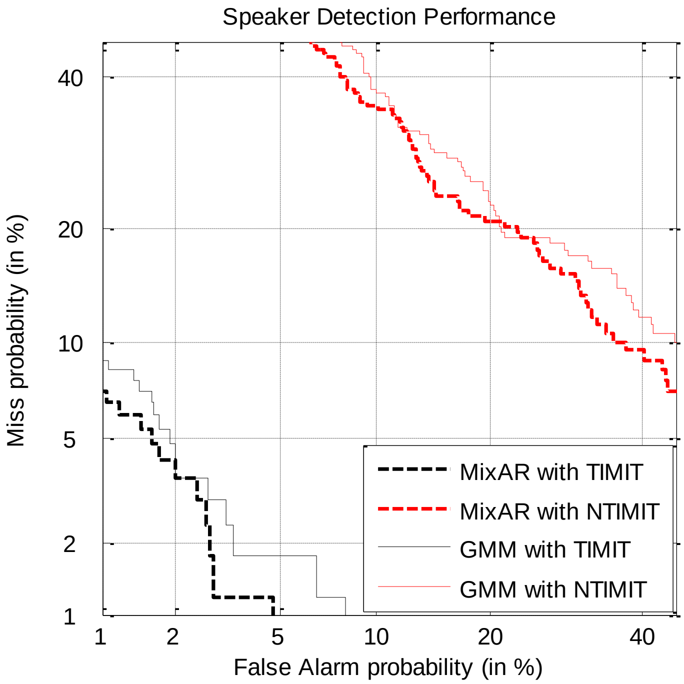

Figure 6. DET curves for GMM and MixAR models on TIMIT and NTIMIT are shown. MixAR performance exceeds GMM performance while using a fewer number of parameters.

# LIST OF TABLES

# I. TABLES

| *α* | **GMM (8 mix.)<br>Static+Δ** | **MixAR (4-mix.)<br>Static** |
|---|---|---|
| 0.00 | 1.50 (288) | 1.50 (240) |
| 0.25 | 3.25 (576) | 3.50 (240) |
| 0.50 | 10.25 (576) | 6.25 (240) |
| 0.75 | 24.75 (576) | 9.75 (240) |
| 1.00 | 26.75 (576) | 13.75 (240) |

Table 1. A comparison of classification error rates is shown for a GMM system using static+delta features and a MixAR system operating only on the static features is shown on synthetic data. The number of parameters for each system is shown in parentheses. The GMM system, which uses static and delta features, performs significantly worse than the MixAR system as the nonlinearity in the data increases.

| **Features** | **GMM<br>16-mix.** | **MixAR<br>8-mix.** |
|---|---:|---:|
| Static(12) | 22.1 | 19.1 |
| Static+E(13) | 33.1 | 41.1 |
| Static+Δ(24) | 20.6 | 20.4 |
| Static+Δ+ΔΔ(36) | 20.5 | 20.5 |

Table 3. Speaker verification EERs are shown for MixAR and GMM for a variety of feature vector combinations. MixAR does not need delta features since the model itself encodes temporal dynamics.

| **No.<br>Mixtures** | **GMM<br>Static+Δ+ΔΔ** | **MixAR<br>Static Only** |
|---:|---|---|
| 2 | 23.1 (216) | 24.1(120) |
| 4 | 21.7 (432) | 19.2(240) |
| 8 | 20.5 (864) | 19.1(480) |
| 16 | 20.5 (1728) | 19.2(960) |

Table 2. EERs are shown as a function of the number of mixtures. MixAR performs slightly better with almost half the number of parameters.

| | **SNR<br>(dB)** | **Clean** | **Car<br>Noise** | **White<br>Noise** | **Babble<br>Noise** |
|---|---:|---:|---:|---:|---:|
| **GMM<br>(1168)** | | **2.4** | | | |
| | 10 dB | | 19.7 | 48.7 | 40.6 |
| | 5 dB | | 31.2 | 50.0 | 44.7 |
| | 0 dB | | 39.3 | 49.8 | 48.2 |
| **MixAR<br>(480)** | | **1.8** | | | |
| | 10 dB | | 13.7 | 47.0 | 36.9 |
| | 5 dB | | 23.2 | 47.6 | 42.8 |
| | 0dB | | 33.9 | 48.5 | 47.6 |

Table 4. EERs are shown for a variety of noise conditions.

| Database | GMM<br>Static+Δ+ΔΔ<br>(1728) | MixAR<br>Static Only<br>(480) |
|---|---|---|
| TIMIT | 2.4 | 1.8 |
| NTIMIT | 21.0 | 20.9 |

Table 6. EERs are shown for TIMIT (clean data) and NTIMIT (noisy data).

| | Training Utterance Duration | EER |
|---|---|---|
| GMM (864) | 120 | 20.5 |
| | 90 | 20.4 |
| | 60 | 20.4 |
| | 30 | 24.4 |
| | 15 | 29.5 |
| MixAR (480) | 120 | 19.2 |
| | 90 | 21.5 |
| | 60 | 21.8 |
| | 30 | 21.8 |
| | 15 | 24.3 |

Table 5. Performance is analyzed as a function of the duration of the training data utterances. The evaluation utterance durations were held constant and varied between 20 and 40 seconds.

| | Evaluation Utterance Duration | EER |
|---|---|---|
| GMM (864) | 30 | 20.5 |
| | 15 | 21.8 |
| | 10 | 21.5 |
| | 5 | 24.4 |
| | 3 | 26.9 |
| MixAR (480) | 30 | 19.2 |
| | 15 | 23.4 |
| | 10 | 23.1 |
| | 5 | 25.6 |
| | 3 | 25.6 |

Table 7. Performance is analyzed as a function of the duration of the evaluation data is shown. The training utterance duration was fixed and averaged around 120s.