# MixAR Modeling of Speech Signals for Speaker and Speech Recognition

Gaussian mixture models are a very successful method for modeling the output distribution of a state in a hidden Markov model (HMM). However, this approach is limited by the assumption that the dynamics of speech features are linear and can be modeled with static features and their derivatives. In this work, a class of nonlinear mixture autoregressive models are used to model state output distributions (MixAR-HMM). These model can handle both static and dynamic features. We apply this model to speech as well as speaker recognition.

**Background and Motivation for Using Nonlinear Mixture Autoregressive Models for Speech MFCCs**

Over the past decade there has been a great deal of interest in breaking the barrier imposed by the assumption of linearity of speech signals. Early in the history of speech processing, linear modeling of speech became the de facto standard due to several reasons.

First, linearity assumption is the simplest possible one (of course, with the exception of assuming a degenerate constant model) for any model. A system is said to be linear if output is proportional to the input and superposition principle holds true. This makes interpretations of the model simple and also it is easy obtain insights into the model easily. Furthermore, it is computationally simple enough to be processed with the computers available then. Moreover, in spite of it’s simplicity, linear modeling provided remarkably good performance in speech processing. Notwithstanding these advantages, we have now reached a threshold in speech research where this linearity assumption is an impediment to further advancements in recognition performance.

Nonlinear modeling goes beyond the simple linear systems concepts and are found to be increasingly useful in describing natural systems. In the case of nonlinear systems the superposition principle no longer holds true and this has a variety of implications. First, the output of a nonlinear model is no longer constrained to be on the same hyperplane as the inputs, i.e., output need not be just a weighted sum of inputs. Secondly, the nonlinearity lends itself to modeling cycles, periods, nonlinear attractors, and invariant properties of the associated with the attractors that are common in natural systems. Thridly, the use of nonlinear techniques can lead to entirely new insights into the structure of the data to be modeled. This is seen especially with topological methods like fractals, nonlinear manifold learning, and, more recent topological invariants like Betti numbers from point clouds.

The foremost argument in favor of nonlinear modeling approaches is that natural systems are never linear. In speech preception too, there is without doubt, some degree of nonlinearity – for e.g., doubling of sound volume is not perceived as twice louder, rather loudness is more of a logarithmic function of volume. Any hope for modeling natural phenomenon, including speech, lies ultimately in our ability to understand and apply nonlinear models. At best, linear systems, can only be simple but crude approximations to naturally occuring signals, and with all their limitations, cannot explain all the salient properties of such signals.

To understand where the linearity assumption arises in speech or speaker recognition, we need to take a closer look at the speech features employed at the front-end and also at the statistical models used for recognition.

The predominantly used features in recognition are Mel-Frequency Cepstral Coefficients [1]. These are derived from speech samples mainly through linear transforms (DFT, DCT, etc.), with the only nonlinear component coming from a log-frequency domain averaging and warping. This nonlinearity is designed to mimic the auditory response of the human ear, but it is not a direct representation of the nonlinearity in the speech signal itself. However, we would expect that if the speech time-series is nonlinearly evolving, then so would the MFCCs.

Conventional statistical modeling of MFCCs have involved Gaussian Mixture Modeling (GMM) of state output probabilities in a Hidden Markov Model (HMM) [1]. In this paradigm, each state of the HMM represents a phoneme (or some other unit of a stable segment of speech), and the corresponding MFCCs are modeled by a mixture of Gaussians random variables - GMM. Transitions between states in the HMM correspond to movement from one phoneme to another. This is illustrated in Fig. 1.

**

*Figure 1: An overview of the GMM-based HMM Modeling.*

One main drawback of applying this model for MFCCs is that the use of GMM enforces the assumption of time-independence [1] - that the output at each time frame is independent of the previous one. This is clearly known to be false - natural speech is much more gradual and smooth, and the same is also true of its MFCC representation. To make up for this incorrect assumption, a convenient fix that is typically employed is to use derivative features in addition to the static ones.

At least in speech recognition tasks, this linear derivative modeling of MFCC dynamics, improves the performance significantly, thus alleviating the effect of the afore-mentioned drawback to some extent [1]. For speaker recognition tasks [2] however, this approach typically has no effect, or leads to worse performance. This latter scenario is contrary to what we would expect if the derivative features were a true and sufficient representation of the speech dynamics, prompting the questions: what’s wrong with linear derivative features, and, how else can we model the MFCC dynamics?

To address the first question about the insufficiency of linear derivative features for modeling speech MFCC dynamics, it is important to assess the presence and importance of nonlinear effects in speech signals. We reason that if significant nonlinearities are found to be present in speech time-series, then these would also manifest in the dynamics of the MFCC representation of speech, making the use of linear derivatives for representing speech dynamics ill-founded. This begs the question, how much nonlinearity is present in speech signals and does it have any bearing on the recognition problem?

On this subject of nonlinearity in speech signals, there have been several attempts at measuring it’s amount and it’s application in recognition [3][4][5][6][7]. In attempting to employ information in nonlinear dynamics for recognition, most of these approaches built on computing nonlinear invariants (e.g. Lyapunov Exponents, Fractal Dimension, Correlation Entropy). These nonlinear invariants quantify the amount or degree of nonlinearity in signals, and under certain assumptions, their values remain unchanged under smooth transformations to the original signal - hence the name invariants. This property of invariance has been one main motivation why these features were applied in the hope that they would remain robust to influences of noise and mismatch in channel conditions. When applying nonliner features, it is hoped that irrespective of signal conditions, different classes (phones) of speech signals will produce different values for the invariants but signals from the same class will have very close values, thus enabling robust recognition. All of the prior work (including ours as a first part of this project [6][7]) have shown that there are significant and varying amounts of nonlinearities in different classes of speech that a broad-phone class classification can be effected based solely on nonlinear invariants as features. However, one respect in which they have all failed is to show robust improvements in recognition with natural continuous speech [7]. We conjecture that this is due to two problems with this approach:

1. In practice, estimation algorithms for nonlinear invariants are sensitive to algorithmic parameters and/or mismatch conditions. While theory leads us to expect that these features should be “invariant” to signal conditions, practical realities like finiteness of data overthrow any such possibilities.
2. Even theoretically speaking invariant features have a disadvantage – they measure only the degree of nonlinearity. Two signals having very different dynamics but the same amount of nonlinearity will have identical invariant features. For example, most periodic-like signals, including all voiced vowels, would have Lyapunov exponent values about zero, and cannot be distinguished based on this alone.

From the above discussion we see that what we need is a way to directly model the nonlinearity rather than just the degree of nonlinearity.

This leads us to the second question: how to model the nonlinear dynamics of MFCC features? One possibility is to try using nonlinear autoregression of each MFCC feature, for example a polynomial model, or a Taylor series. However, these models typically have so many more parameters to estimate that it may be difficult to get reliable estimates. Furthermore, it would be desirable to have a new model that has obvious parallels to GMM so that we may build on past experience and also integrate the new model into the HMM framework, but polynomial and Taylor series models are far removed from this ideal.

In retrospect, what we desire is a statistical model for speech MFCCs that:

1. takes past dependence into account explicitly (rather than using extra dynamic features)
2. models the actual nonlinear evolution (instead of quantifying only the amount of nonlinearity)
3. is a weighted mixture of simpler models (just as GMM is a weighted mixture of Gaussian models)

Mixture of autoregressive models fits particularly well with these objectives [8][9]. It is a mixture model with each component made of a simple linear autoregressive filter and a mean. Each autoregressive component uses a weighted sum of past samples to predict the present sample. The components are weighted probabilistically, and this probabilisitc mixing of linear AR processes lends itself to nonlinear evolution modeling.

**Mixture of Autoregressive Models**

There are several kinds of models that fall under the same name of mixture of autoregressive models [8][9][10][11][12], found in both statistics and speech processing literature. Of these, the most general is the MixAR [8]. Other mixture of autoregressive models can be derived from this general model under special conditions. In this work, we identify four types that can be derived from the general MixAR model.

The general MixAR process is defined by:



where,

εi : zero-mean Gaussian random process with a variance of σj2

w.p.: with probability

*p1*: prediction order

*p2*: gate order.

{*ai,j*} *j>0*: linear predictor coefficients for component *i*

*ai,0*: mean for component *i*.

*gi*: gate functionfor component *i*, assigns probability to each mixture component based on the previous *p2* samples. The only requirement for the gate functions is that their values sum to *1* at each sample instant.

A convenient and popular functional form for the gate function is :



where,

*{Aj,i}*: gate parameters for mixture component *j*.

It is apparent that an *m*-mixture MixAR process is the weighted sum of *m* Gaussian autoregressive processes, with time-dependent weights dependent on previous samples. Here we have generalized the model in [8] by decoupling the gate order and the prediction order. In its previous formulation both these orders were constrained to be equal. Since there is no reason for forcing this constraint, we consider the values for these two orders to be distinct and independent. This allows us to test for the contributions from time-dependency of gate components and AR components, individually and also in conjunction. For this, we make four kinds of assumptions to derive four types of models.

Letting MixAR to denote the general class of mixture of autoregressive models, we identify the four special types of models in Table 1:

*Table 1: The four types of models derived from general MixAR*

|  |  |  |
| --- | --- | --- |
| **MixAR Type** | **Assumptions** | **Equivalent Model in Literature** |
| 1 | *p1=0, p2=0* | GMM [1] |
| 2 | *p1=0, p2>0* | Mixture of Experts [13] |
| 3 | *p1>0, p2=0* | MAR [9] |
| 4 | *p1>0, p2>0* | MixAR (general model) [8] |

The theoretical utility of identifying these 4 types of models from the general model is that it povides a unified view of these models. Taking this approach for mixture autoregressive models, we can see the effects of the various assumptions independently. In addition, the practical advantage in distinguishing these models from the general MixAR model is that the training procedures for models vary depending on the assumptions of gate and prediction orders. When gate order is 0 (Types 1 and 3), the reestimation equations have a closed form expression but when it is non-zero (Types 2 and 4), a gradient descent approach is required. When prediction order is 0 (Types 1 and 2), we can use the same EM reestimation equations for mean and variance as for a GMM (Type 1), but when this is non-zero (Types 3 and 4), we need to resort to a weighted-covariance type approach for estimation of prediction coefficients.

One property of MixAR that is of particular relevance here is the ability to model nonlinearity in time series. Though the individual component AR processes are linear, the probabilistic mixing of these AR processes constitutes a nonlinear model. In a GMM, the distribution remains invariant to the past samples due to the static nature of the model. For MixAR, the conditional distribution given past data varies with time. This model is capable of modeling both the conditional means and variances. Thus, MixAR can model time series that evolve nonlinearly. This property becomes important in speech processing in the light of recent work on nonlinear processing of speech [4][5]. Some other properties of MAR including conditions required for the process to be stationary are derived in [9].

We integrated the MAR model into the HMM framework by replacing the GMM output probabilities with that of MAR. This is illustrated in Fig. 2.



*Figure 2: An overview of the MAR-HMM approach.*

**Relationship to Other Mixture Auoregressive Models**

As mentioned earlier, the MixAR model is related to a family of models found both in statistical and speech literature. The general MixAR model has been applied to two benchmark time series - sunspots, and Canadian lynx trapping data – for prediction application [9], and shown to be superior to linear models. MAR model was shown to be superior to linear models on two real data prediction – IBM stock prices, and, Canadian lynx data. Mixture of Experts model has remained an integral part of several neural networks applications related to soft-threshold partitioning of feature space [13].

There have also been applications of some variants of autoregressive models with HMMs in speech processing. All of these can be derived as special cases of MAR-HMM. One of the first applications of autoregressive HMMs in speech processing assumed an autoregressive (AR) model for each state, so that the short term correlations in the speech signal and known linguistic properties of sound combinations could be modeled by the state transitions [12]. This model was effectively a single-mixture component MAR-HMM.

The next major advance was the introduction of mixture autoregressive HMMs in [10]. This work applied a weighted mixture of AR filters to model observations at each state. While this appears to be very similar to the MAR-HMM developed in this paper, this approach had two major shortcomings. The model in [10] assumed that all AR components had the same variance, and that each was zero mean. This is equivalent to constraining the MAR model to have zero means and equal variance. In this respect the MAR-HMM considered in our work is more general than the autoregressive models previously applied to speech.

A variant of the original AR-HMM, using switching autoregressive process was considered in [11]. In this approach, the signal correlations during HMM state transitions were also modeled by the switching process. However, this model again was restricted to a single component AR, and thus it too is equivalent to a single‑component MAR. Moreover, these variants of AR‑HMMs considered only scalar speech time series as observations. Our extensions to vector time series are crucial to application of these models to speech recognition.

**Parameter Estimation using (G)EM** **- Derivation**

Similar to GMM training, maximum likelihood estimates for MAR parameters can be calculated using the Expectation Maximization (EM) algorithm [14]. During E-step, the probability (expectation) that each sample was generated from each of the mixture component from the current model is computed. During M-step, weight, mean, and, covariance parameters are updated to maximize the overall data likelihood. These two steps are then run iteratively until convergence of likelihood is achieved.

Given the orders, *p1* and *p2*, the parameter set for each of the *m* components of a MAR model consists of predictor coefficients (including the mean), the error variance, and the gate parameters:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Unlike the earlier publications on MixAR model, we use an alternative approach using Q-function to derive the EM update equations [15]. Since direct maximization of likelihood (or log likilihood) is difficult for this problem, we resort to the use of the auxillary Q-function. It is known from Jensen’s inequality that updating parameters to maximize Q-function is equivalent to maximizing likelihood. Hence this approach is justified.

The auxillary function is defined as [15]:



where,  is the updated parameter set,  is the current parameter values,  is the set of training data samples, and is the set hidden states. At any instant, the sample could have arisen from any of the components of the mixture and the hidden state refers to the specific outcome that a sample arose from a particular mixture. Realizing the expectation operator in terms of the actual sum over the marginal distribution, we get:



where, the summation is over the set of all possible hidden state sequence *Y* and *L* is the overall likelihood of data given the model. With the gate function defined as:



and the Gaussian AR probabilities are defined as:



the *Q* function can be expanded using independence assumption, we obtain:



Here *p(l,.)* is the probability that the hidden state at sample instant n is *l*. Denoting this with the more common notation of , we have:



Using the identity log(AB)=log(A)+log(B), we can expand the partition the Q-function into the constituent contributions from the gates and the Gaussian AR components as follows:



From our expression for the Gaussian AR probability, we have:



This can also be written using vectorial notation as:



where, .

Maximizing *Q* w.r.t. variance, we differentiate w.r.t. to σ and setting it to zero, obtaining:



Solving:



Differentiating w.r.t. the prediction coefficients, a set of *m* linear equations are obtained:



From these, the solution for the prediction coefficients (including mean) are obtained by the following matrix solution:



where,





Unfortunately, there is no closed-form solution for the gate parameter updates. Instead we need to resort to a steepest ascent approach as follows:

Note that there are two additional design parameters involved in this update: α, and a δ parameter for estimating the derivative of *Q*. The equations for gate parameter update have no closed-form solution, but we only move the parameter values in the direction of increase in likelihood (with a scale factor *α*) at each iteration. In this framework, we are under the framework of Generalized EM algorithm (GEM), where instead of maximizing likelihood at each iteration we only gaurantee parameter updates towards increasing likelihood.

To estimate these parameters, we first need an initial guess for these parameters and then we iterate with EM steps 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 MAR to have the same mean, variance, and weight as the GMM model. These initial parameters can be then refined recursively using Expectation and Maximization steps.



*MixAR Types as Special Cases:*

If *p2=0* (Types 1 and 3) we reduce to the familiar weight equations for GMM. In this case we have the closed-form solution for gate parameters:



If *p1=0* (Types 1 and 2), the AR parameter update equation simplifies to the familiar GMM mean and variance update equation:





**Pilot Experiments with Mixture Autoregressive Model as Classifier**

To better understand the efficacy of the MAR-HMM model, we evaluated its performance on two simple pattern recognition tasks. The first task represents data with known nonlinearities. The second task is a simple phone classification task.

*Two-Class Problem*

The MAR-HMM approach, like GMM-HMMs, can perform classification using a maximum likelihood approach. The log likelihood of data given a set of MAR-HMM model parameters is used to score each model and the class with the maximum score is chosen. A two-class classification problem was designed where data are randomly generated randomly according to the following MAR model parameters:





where Θ1 and Θ2 correspond to classes 1 and 2, respectively.

For this example we chose the parameters for class 2 such that the marginal distribution is about the same as that of the first class, but it lacked the dependence on past samples unlike class 1. Hence the data for class 2 follows only a GMM distribution. This was done to demonstrate a case where GMM would be unable to achieve good classification due to its ability to capture the dynamics in the model. The results of these experiments, along with the number of parameters for each model, are shown in Table 2.

In addition to listing accuracy, the numbers of parameters for each model are shown. Since in this case we knew that the distribution can have a maximum of 4 modes, we use only 2‑ and 4‑mixture models. It can be observed that MAR, with just 2 components and 8 parameters can achieve 100% classification accuracy using only static features. The GMM approach using only static features is unable to do much better than a random guess strategy since the two classes have similar static marginal distribution. This demonstrates MAR-HMM’s ability to learn dynamic information.

With the inclusion of delta coefficients, the GMM performance increases significantly, but even in this case it achieves only 85% accuracy with 28 parameters. Though delta features capture some amount of dynamic information in the features, it is still only a linear approximation, and we cannot capture their nonlinear evolution with just GMMs. From the above, it is clear that at least some dynamic information is better modeled using MAR-HMM.

**Speaker Recognition Experiments with Mixture Autoregressive Models**

The goal in speaker recognition is to validate the identity of a speaker given speech data from that speaker. The two types of speaker recongition tasks are speaker identification and speaker verification. In speaker identification, the identity of a speaker is found from a set of apriori known database of speakers. Speaker verification, on the other hand, involves a particular claim for a speaker and the claim is accepted or rejected. In the suite of experiments presented here, we are concerned with speaker identification. An overview of this task is presented in Fig. 3.

**

*Figure 3: An Overview of the Speaker Recognition Problem.*

In spite of advancements like HMMs and use of dynamic features (deltas) in speech recognition, straight-forward GMM modeling of speakers has proved to be the most effective in speaker recognition. While these advancements over GMMs have improved performance significantly for speech recognition, it is clear that they are inept at capturing dynamical information for speakers. Our work attempts to remedy this situation by using MixAR models for quantitatively capturing the nonlinear dynamics of speakers. In the following we describe the data we used for speaker recognition and the corresponding results.

We use TIMIT database for speaker recongition. All of the 168 speakers in the test part of the database were utilized. For each speaker, the 4 SX and 4 SI sentences were used for training. The remaining 2 SA sentences were combined to form the test data for that speaker. We first perform tuning experiments with only 26 speakers from the DR2 dialect region, while the final experiments comprise of tests with all 168 test speakers.

Table 2: *Classification (% accuracy) results for synthetic data*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **# mixtures** | **GMM**  **Static only** | **MAR**  **Static  only** | **GMM**  **static+**  ∆**+**∆∆ | **MAR**  **static+**  ∆**+**∆∆ |
| 2 | 47.5 (6) | 100.0 (8) | 82.5 (14) | 100.0 (20) |
| 4 | 52.5 (12) | 100.0 (16) | 85.0 (28) | 100.0 (40) |

*Table 3: Tuning Experiment for Number of Mixtures - Speaker Recognition Performance for Different Number of GMM Mixtures (26 speakers from DR2 dialect region of TIMIT test data)*

|  |  |
| --- | --- |
| **GMM # mixtures** | **% Speaker Recog. Error** |
| 1 | 46.15 |
| 2 | 19.23 |
| 4 | 15.38 |
| 8 | 15.38 |
| 16 | 19.23 |

We extracted only 12 static MFCC features since, as mentioned earlier, dynamic features are ineffective in speaker recognition. In addition, we did not use the energy coefficient, since this lowered the recognition performance slightly. Even though all TIMIT data were recorded with the same recording conditions, in future we want to investigate mismatch condition effects using NTIMIT, which is essentially TIMIT data passed over telephone channels. Cepstral mean subtraction (CMS) method was used on MFCCs to compensate for mismatch in channel conditions.

Since our model formulation was for scalar case only, we apply all the four types of model for each MFCC coefficient individually. Thus, we have forced an assumption of independence between the features, but this is not too restrictive for MFCCs since they are known to be uncorrelated. Note that this is a common assumption even with GMM-based approaches where the covariance matrices are assumed to be diagonal. However, there is a weak tying between the scalar features within each Gaussian in the form of a commonly tied weight. In our present experiments, we do not do this, and instead model each MFCC scalar completely independent of the other, even with distinct weights for each model component for each scalar. It was done this way because Type 2 and Type 4 MixARs have frame-dependent weights and these weights are dependent on the previous frame MFCC values. Hence, it is not possible to tie the weights across scalar components for these two models. For uniformity, we chose to keep all the four types to have such untied weights.

As a first experiment to test for the efficacy of each of the four types of models and to fix the number of mixtures, we used a smaller database of 26 speakers from dialect region DR2 and compared the speaker recognition error rate. First we run an experiment to tune the number of mixtures using GMM model. The results are shown in Table 4. From this, 4 mixtures per feature was found to be about optimal for each speaker and we use this in the following experiments. In this case, the decrease in performance for 16 mixtures is contrary to expectations, and could be because we have only very limited training data to estimate a large numbe rof parameters, and hence we do not get reliable estimates.

*Table 4: Speaker Recognition Performance with 26 speakers from DR2 dialect region of TIMIT test data*

|  |  |
| --- | --- |
| **MixAR Type** | **% Speaker Recog. Error** |
| 1 | 15.38 |
| 2 | 30.77 |
| 3 | 38.46 |
| 4 | 11.54 |

Next, the speaker recognition performance for the four types of MixAR models were found and results are in Table 4. From this table we see that while Types 2 and 3 do not perform that well compared to GMM, Type 4 (full MixAR) does better than GMM. However, this study suffers from the fact that there are only 26 speakers and so this may not be statistically significant.

*Table 5: Speaker Recognition Performance with all 168 speakers from TIMIT test data*

|  |  |
| --- | --- |
| **MixAR Type** | **% Speaker Recog. Error** |
| 1 (GMM) | 37.50 |
| 4 (full MixAR) | 36.90 |

In the next experiment we test with all 168 speakers for Type 1 (GMM) and Type 4 (full MixAR). The results are in Table 5. From this table, it is clear that while there is an improvement in performance using MixAR instead of GMM for speaker recognition and that this improvement could be significant statistically, the improvement is only marginal. This is for a clean test case, and we would like to study the further the performance difference under noisy and mismatch conditions. We expect that MixAR modeling would prove to be more robust than GMM modeling.

**Speech Recognition Experiments with Mixture Autoregressive Models**

For speech recognition, we applied the MAR model (MixAR Type 3 model) in the framework of HMMs to phone classification and recognition tasks. The MAR-HMM model we have developed is a generalized version of [5] that has been extended to handle vector observations, so that we can operate on the speech feature vector stream rather than speech samples. One property of MAR that is of particular relevance is the ability of MAR to model nonlinearity in time series. Though the individual component AR processes are linear, the probabilistic mixing of these AR processes constitutes a nonlinear model. In a GMM, the distribution remains invariant to the past samples due to the static nature of the model. For MAR, the conditional distribution given past data varies with time. This model is capable of modeling both the conditional means and variances. Thus, MAR can model time series that evolve nonlinearly.

We first wanted to test the efficacy of MAR-HMM in a simple speech recognition setting, so we performed a sustained phone classification experiment. We made 16 kHz recordings of three distinct phones – “aa” (vowel), “m” (nasal), and “sh” (sibilant). For each phone and for each speaker, 35 recordings were made to serve as training database, while another 15 were reserved for testing. Silence was removed so that we could focus on the ability of the approach to model speech. We evaluated the performances of 2-, 4-, 8-, and 16 mixture GMM-HMM and MAR-HMM with the 13 dimensional static MFCC features. The results are in Table 6.

For an equal number of parameters, MAR outperformed GMM significantly. For instance, MAR-HMM achieved a phone classification accuracy of 94.4% with only 320 parameters while a GMM system using 432 parameters could only achieve a 93.3%.

To determine whether MAR is more effective at exploiting dynamics than what GMM can achieve using dynamic features, we also performed another experiment with 39-dimensional features containing both static as well as velocity and acceleration coefficients. The results are in Table 7.

*Table 6: Sustained phone classification (% accuracy) results with MAR and GMM using static MFCC features (the numbers of parameters are shown in parentheses).*

|  |  |  |
| --- | --- | --- |
| **# mixtures** | **GMM** | **MAR** |
| 2 | 77.8 (54) | 83.3 (80) |
| 4 | 86.7 (108) | 90.0 (160) |
| 8 | 91.1 (216) | 94.4 (320) |
| 16 | 93.3 (432) | 95.6 (640) |

In this case, the results were not conclusive. While MAR-HMM showed an accuracy rate of 97.8% with 472 parameters, GMM-HMM attained only 96.7% accuracy with 632 parameters.

Unfortunately, the performance of MAR-HMM saturated with an increase in the number of parameters. For example, MAR-HMM at 1888 parameters achieved only 98.9% accuracy while GMM-HMM achieved 100% with 1264 parameters. We suspect that this could be due to the fact that our parameter estimation and likelihood computation procedures assume that the features are independent. It is well-known that the static MFCC features are uncorrelated (at least, theoretically), but obviously the delta features are correlated with the static ones. While this should also cause problems for GMM, the problem is more acute for MAR because in this case, unlike GMMs, we employ the past history explicitly.

This work is described in a paper that was presented at INTERSPEECH’2008 .

*Table 7: Sustained phone classification (% accuracy) results with MAR and GMM using static+∆+∆∆ MFCC features (the numbers of parameters are shown in parentheses).*

|  |  |  |
| --- | --- | --- |
| **# mixtures** | **GMM** | **MAR** |
| 2 | 92.2 (158) | 94.4 (236) |
| 4 | 94.4 (316) | 97.8 (472) |
| 8 | 96.7 (632) | 97.8 (944) |
| 16 | 100 (1264) | 98.9 (1888) |

Since the positive results with static features on the simple phone classification experiment above was encouraging, we next applied MAR-HMM to a larger scale phone recognition experiment with TIMIT database. We used the full training part of TIMIT for training and the core test part for testing. Since for all speakers the SA sentences had the same transcriptions, these were avoided both during training and testing to avoid biasing the results. A bigram language model was used with 16 mixture MAR-HMM and a 16-mixture GMM-HMM served as the baseline. The results are in the Tables 8 and 9.

*Table 8: Performance for 39 (static+dynamic) MFCCs and with 16 mixtures*

|  |  |
| --- | --- |
| **Acoustic Model for HMM** | **Phone Recognition Accuracy (%)** |
| GMM | 69.5 |
| MAR | 59.2 |

From table 8, it is seen that for recognition with static as well as dynamic features, GMM performs significantly better. We expected this to be the case from our experience with sustained phone experiments, where MAR proved superior to GMM with static only features, but inferior when derivative features are included. Unfortunately, contrary to our expectations, we found that MAR performance was worse than for GMM even for static features only case, as shown in Table 8. We are investiagating the possible causes for this; perhaps it is a problem of reliable parameter estimation. However, it is alos possible that MAR is not suited for speech representation as was the case for speaker recognition. We hope, like our success in speaker recognition, the full MixAR Type 4 model will work well with speech recognition too. This work is in progress.

**Future Work with MixaR**

*Table 9: Performance for 13 static MFCCs and with 16 mixtures*

|  |  |
| --- | --- |
| **Acoustic Model for HMM** | **Phone Recognition Accuracy (%)** |
| GMM | 51.5 |
| MAR | 42.3 |

There are several areas which are still left unexplored in this work. Some of the most promising ones are mentioned below.

For speech recognition, first we would like to make a thorough study of the 4 types of MixARs. Till now we have only experimented with GMM (Type 0) and MAR (Type 4) in an HMM setting and we found the performance of the latter to be worse than that of the former. Comapring this with the speaker recognition case, we see that same was also true in that case. On the other hand, for speaker recognition, MixAR (Type 4) was the most successful, and it is highly likely that this trend would also seen for speech recognition. Moreover, from experience with linear dynamic features, we know that their impact on speaker recognition performance is not so significant, but that over speech recognition is. Thus we expect the MixAR (Type 4) model to perform much better than GMMs over static features – and at least as good as GMMs over combined static and derivative features.

In addition, we want to perform an analysis of the recognition performances for the different phonemes. This would give us an indication of what MixAR learns that GMM does not and for which sounds nonlinear dynamic information is more important. This should add to our understanding of the complexity of the speech sounds that linear models fail to pick up.

For speaker recognition, we saw that the performance improved slightly with the use of MixAR model compared to the conventional GMM, for clean TIMIT database. Next, we will investigate these two models for robustness under different noise and channel mismatch conditions. Since MixAR captures additional nonlinear information in speech dynamics compared to GMM, we expect MixAR to be more robust than GMM. We propose to test this hypothesis using NIST Speaker Evaluation database. In this database the training database is clean while the test database contains speaker data corrupted by noise and differing channel conditions. This would be an ideal test for studying robustness of nonlinear models.

Furthermore, we would like to analyze how the individual mixture components are utilized by different speakers. In particular, we want to investigate how the components are partitioned among genders, dialects, etc. and how different this is for MixAR and GMM.

Also, we would next like to investigate the performance of MixAR model for speaker verification. For this, we will use the true speaker scores from the correct claimants, and impostor scores from the impostor speakers, and then plot Detection Error Tradeoff (CET) Curves between miss and false alarm probabilities by varying threshold for speaker score acceptance. If the DET curve for MixAR is closer to the origin than that for GMM then this would indicate that MixAR performs better. In particular we can compare Equal Error Rates (EER) and Minimum Discrete Cost Function (MinDCF) points to get a more quantitative and direct comparison between the two models.

Finally, it could also be worthwhile to investigate discriminative training approaches for MixAR modeling. Over the past few years, there has been rapidly growing interest in discriminative training for GMM and HMM, and it is highly likely the advantages of these methods for GMM also carry over to MixAR modeling.

**References :**

1. X. Huang, A. Acero and H. W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, Prentice Hall, NJ, 2001.
2. A. Morris, D. Wu and J. Koreman, “GMM based clustering and speaker separability in the TIMIT speech database,” IEICE Trans. Fundamentals/Communn./Electron./Inf. & Syst., Vol. E85-A/B/C/D, No. 1, March 2005.
3. M. Banbrook, “Nonlinear analysis of speech from a synthesis perspective,” *PhD Dissertation,* The University of Edinburgh, Edinburgh, UK, 1996.
4. A. Kumar and S. K. Mullick, “Nonlinear Dynamical Analysis of Speech,” *Journal of the Acoustical Society of America*, vol. 100, no. 1, pp. 615-629, July 1996.
5. I. Kokkinos and P. Maragos, “Nonlinear Speech Analysis using Models for Chaotic Systems,” *IEEE Transactions on Speech and Audio Processing*, pp. 1098- 1109, Nov. 2005.
6. S. Prasad, S. Srinivasan, M. Pannuri, G. Lazarou and J. Picone, “Nonlinear Dynamical Invariants for Speech Recognition,” *Proceedings of the International Conference on Spoken Language Processing*, pp. 2518-2521, Pittsburgh, Pennsylvania, USA, Sept. 2006.
7. D. May, “Nonlinear Dynamic Invariants for Continuous Speech Recognition,” *PhD Dissertation*, Mississippi State University, Mississippi, USA, 2008.
8. A. Zeevi, R. Meir and R. Adler, "Non-linear models for time series using mixtures of autoregressive models," *Technical report*, Technion and Stanford University, 2000.
9. C.S. Wong and W.K. Li, "On a Mixture Autoregressive Model," *Journal of Royal Statistical Society*, vol. B62, pp. 91-115, 2000.
10. Juang, B. H., and Rabiner, L. R., “Mixture Autoregressive Hidden Markov Models for Speech Signals,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 6, pp. 1404-1413, December 1985.
11. Ephraim, Y. and Roberts, W. J. J., “Revisiting Autoregressive Hidden Markov Modeling of Speech Signals,” *IEEE Signal Processing Letters*, vol. 12, no. 2, pp. 166-169, February 2005.
12. Poritz, A. B., “Linear Predictive Hidden Markov Models,” *Proceedings of the Symposium on the Application of Hidden Markov Models to Text and Speech*, Princeton, New Jersey, USA, pp. 88-142, October 1980.
13. M. Jordan and R. Jacobs, “Hierarchical Mixture of Experts and the EM algorithm,” *Neural Computation*, vol. 6, pp 181-214, 1994.
14. Dempster, A., Laird, N., and Rubin, D., “Maximum Likelihood From Incomplete Data Via the EM Algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1-38, February 1977.
15. J. A. Bilmes, “A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models,” Technical Report ICSI-TR-97-021, University of Berkeley, 1998.