# AANN-HMM Models for Speaker Verification and Speech Recognition

Sachin Joshi, Kishore Prahallad, B. Yegnanarayana

***Abstract*— Pattern classification is an important task in speech recognition and speaker verification. Given the feature vectors of an input the goal is to capture the characteristics of these features unique to each class. This paper deals with exploring Auto Associative Neural Network (AANN) models for the task of speaker verification and speech recognition. We show that AANN models produce comparable performance with that of GMM based speaker verification and speech recognition.**

## I. Introduction

In tasks like speaker verification, speech recognition pattern classification is an important task. Given the feature vectors of an input the goal is to capture the characteristics of these features unique to each class. Traditionally Gaussian Mixture Models (GMMs) are used to capture the distribution of the data. GMMs use first and second order statistics and a set of mixture weights to capture the distribution of the data. In practical speech and speaker recognition problems the data encountered has a complex distribution characterized by high order statistics. Hence it is worth exploring alternate models for the tasks of speech recognition and speaker verification. This paper deals with exploring AutoAssociative Neural Network (AANN) models which capture a probability surface characterizing the manifolds spanning the data. The probability surface could be used as signature of a particular class and in classification problems such as speech recognition and speaker recognition.

A feed-forward neural network performing an identity mapping of the input space is known as autoassociative neural network model [1]. There exists a relationship between principal component analysis and weights learned by a 3-layer AANN model [2]. Principal Component Analysis (PCA) is a method of representing the distribution of a given data in terms of orthogonal components [3], [4], [1]. These orthogonal components account for the variance of the data. Projection of the input data onto the linear subspace spanned by the significant orthogonal components has been used as a technique for dimensionality reduction [5].

Attempts have been made to relax the linear constraints of PCA by using nonlinear activation function in AANN models. Bourlard *et. al.,* [2] have shown that the use of nonlinear units in a three layer AANN model did not provide a solution that is significantly better than PCA. However, Bianchini *et. al.,* [6] have shown that nonlinear activation functions performs $\epsilon$-association,which could be interpreted as a form of clustering in linear subspace. Another interesting perspective is from the addition of more number of hidden layers. Kramer [7] has shown that addition of hidden layers before and after the compression layer projects the input data onto a nonlinear subspace.

While there have been interesting research and different interpretation of AANN Models, due to lack of comprehensive summary of these interpretations, the nonlinear subspace captured by the five layer AANN model has been mostly used in applications involving dimensionality reduction [5].

In this paper, we provide visualization of the theoretical results of [2], [6] and [7] using 2-D data and present a different perspective of using AANN models to capture the distribution of data in the input feature space. We also demonstrate that AANN models could be applied for the task of speaker verification and AANN-HMM models for the task of speech recognition.

## II. Distribution Capturing Ability of AANN Models

Consider a three layer AANN model with M units in the input and output layers, and $p < M$ units in the hidden layer. Let $X = [x_1, x_2, \cdots, x_N]$ be the $M \times N$ matrix formed by the N input vectors, and let $Y = [y_1, y_2, \cdots, y_N]$ be the $M \times N$matrix formed by the vectors realized at the units in the output layer of the network. The match between X and Y is measured in terms of mean square error $J = ||X - Y||^2$ , where $||.||^2$ indicate squared norm. Let $W^T = [u_{ij}] \; \epsilon \; R^{p \times M}$ represent the weight matrix connecting the input layer and the hidden layer, and $W = [v_{ij}] \; \epsilon \; R^{M \times p}$ represent the weight matrix connecting the output layer and the hidden layer. For linear activation function at the units in the input and output layers, $J = ||X - WF(W^T X)||^2$, where F is activation function at the hidden units.

For linear activation function at the hidden units, $F(W^T X) = W^T X$ . Therefore, $J = ||X - WW^T X||^2 = ||X - \phi X||^2$, where $\phi = WW^T$. Since the rank of the matrix $\phi X$ is $p < M$ , the product $\phi X$ minimizing J is the best rank $(p)$ approximation of X in the Euclidean space. This rank can be obtained using Singular Value Decomposition (SVD) of X [2], [8]. It is shown that optimal weights of the network minimizing J corresponds to the principal (singular) vectors of the co-variance matrix $XX^T$ [2]. In other words, the units in the hidden layer capture the linear subspace spanned by the first $p$ principal components of the given data.

For illustration, consider a three layer AANN model with one linear unit in the hidden layer. The model is trained

Sachin Joshi, Kishore Prahallad and B. Yegnanarayana are with International Institute of Information Technology, Hyderabad, India. Kishore Prahallad is also with Language Technologies Institute, Carnegie Mellon University, USA. (email: {sachin_sj, kishore, yegna}@iiit.ac.in).

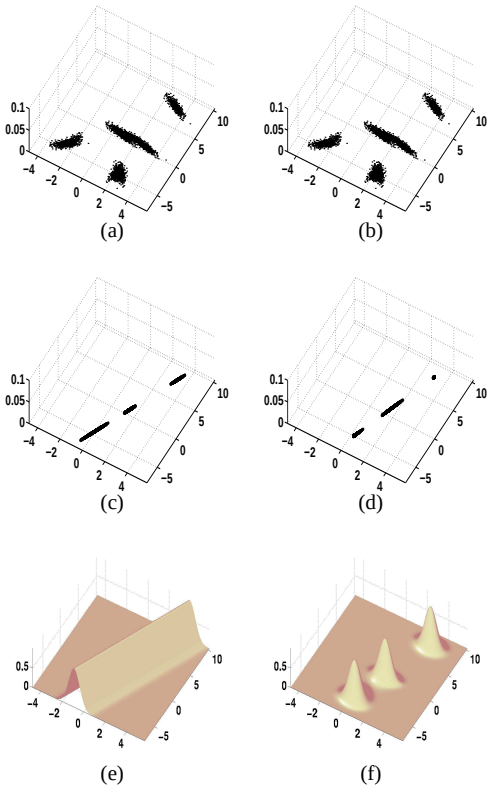



Fig. 1. (a) 2-D data (A 3-D view is shown). (b) 2-D data shown in (a) is repeated. (c) Output of the 3 layer network 2L 1L 2L. (d) Output of the 3 layer network 2L 1N 2L. (e) Probability surface captured by the network 2L 1L 2L. (f) Probability surface captured by the network 2L 1N 2L. Here L refers to a linear unit and N refers to a nonlinear unit.

with the artificial 2-D data shown in Fig.1(a) using back-propagation learning algorithm in pattern mode [4],[1]. The distribution (shown by solid lines in Fig.1(c)) of the input vectors is captured by the AANN model. From Fig.1(c), we observe that the linear subspace captured by the network is along the principal direction of the input data. In order to visualize the distribution better, one can plot the training error for each input data point in the form of some probability surface as shown in Fig.1(e). The training error $E_i$ for the data point $(i)$ in the input space is plotted as $f_i = e^{-E_i/\alpha}$ , where we have used $\alpha = 2$. We call the resulting surface of $f_i$ as $probabilitysurface$, even though it is not strictly a probability density function. The plot of the probability surface shows larger amplitude for smaller error $E_i$ , indicating better match of the network for that data point. We use the probability surface to study the characteristics of the distribution of the input data captured by the network [9].

If the activation function at the hidden units is nonlinear of the type $tanh(.)$, then the nonlinear activation function is approximated by a linear function, and hence the weights are obtained using SVD of X [2]. It follows that the sub space formed at the hidden layer is linear. The linear subspace captured by the three layer AANN model with the nonlinear hidden unit is shown in Fig.1(d). The effect of nonlinear activation function can be observed better from the probability surface shown in Fig.1(f). The network is able to cluster the input data because of the nonlinear activation function at the hidden unit.

The clustering ability of the AANN model can be explained with the concept of $\epsilon$-autoassociation described in [6]. The data set X is said to be $\epsilon$-autoassociated with $\phi = WW^T, if \frac{||Y-X||^2}{||Y||^2} < \epsilon$ , where $\epsilon \in R+$ . To find the range of $\lambda(\lambda \epsilon R+)$ for which $\lambda X$ is $\epsilon$-autoassociated with $\phi$ , consider the following:

$$\frac{||Y - \lambda X||^2}{||Y||^2} < \epsilon \Rightarrow \frac{||WF(W^T \lambda X) - \lambda X||^2}{||WF(W^T \lambda X)||^2} < \epsilon$$

We observe that if a linear activation is used at the hidden units, $\lambda$ gets canceled in the numerator and denominator, and the inequality holds for all values of $\lambda$. For a nonlinear activation function, it was shown in [6] that the above inequality holds for $\lambda < \lambda_\epsilon$ , where $\lambda_\epsilon = (1+\epsilon)||W||^2/||X||^2$. Thus, only limited points in the input space are $\epsilon$-autoassociated with the weights of network. The linear subspace captured by the three layer AANN model may not produce a low $\epsilon$ for all the training data, and hence the probability surface shown in Fig.1(f) does not reflect the distribution of the training data. It is necessary to capture the nonlinear subspace to obtain a low $\epsilon$ for all the training data.

In summary, a three layer AANN model with linear hidden units captures the linear subspace along the direction of maximum variance of the given data as shown in Fig.1(e). But if nonlinear activation function is provided at the hidden units, then the three layer architecture clusters the input data in the linear subspace. Limitation of the three layer AANN model is its inability to capture the nonlinear subspace needed to describe the distribution of data in the input space. We show that a five layer AANN model captures the desired nonlinear subspaces.

## III. Distribution Capturing Ability of 5 Layer AANN Models

The five layer AANN model shown in Fig.2 performs Nonlinear Principal Component Analysis (NLPCA) [7]. The second and fourth layers of the network have more units than in the input layer. The third layer has fewer units than the first or fifth. The activation function at the units in layer 3 may be linear or nonlinear, but the activation function at the nodes in layers 2 and 4 are nonlinear. The function of the five layer AANN model can be understood better by splitting the five layers into mapping (layers 1, 2 and 3) and demapping (layers 3, 4 and 5) networks. The mapping network projects the input space $R^M$ onto an arbitrary subspace $R^p$ , where $p < M$ .

The mapping function G is nonlinear, and a nonlinear subspace is formed at the third layer. The projection of the nonlinear subspace $R^p$ back into the input space $R^M$ is performed by the demapping network, and the demapping function H is also nonlinear. The mapping and demapping functions may not be unique for the given data. This can be observed from Figs.3(a) and 3(b), where two different hyper

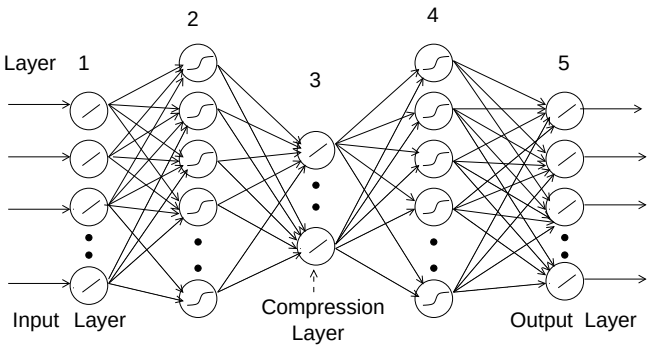
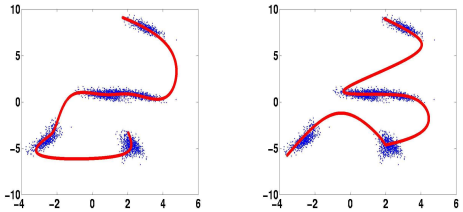
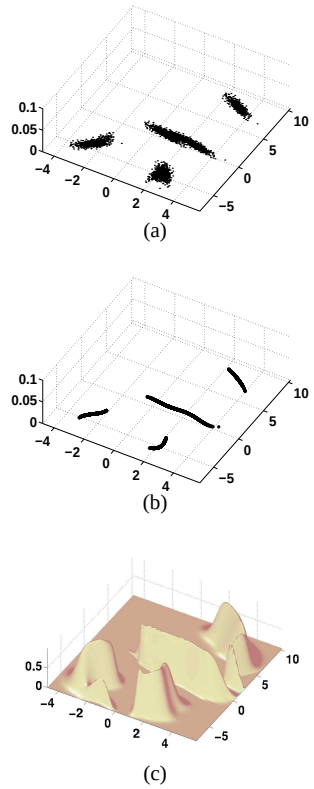



Fig. 2. Five layer AANN model



(a) Training session 1 (b) Training session 2

Fig. 3. Outputs (solid lines) of the 5 layer network 2L 12N 1N 12N 2L for uniformly spaced points in the input space for two different training sessions. The network is trained with the 2-D data shown in Fig.1. The 2-D data is also plotted in the figures.



(a) (b) (c)

Fig. 4. (a) 2-D data (A 3-D view is shown). (b) Output of the 5 layer network 2L 12N 1N 12N 2L. (c) Probability surface captured by the network.

surfaces are captured for two different trials by the same five layer network (2L 12N 1N 12N 2L) for the artificial 2-D data shown in Fig.1(a).

The hyper surfaces captured by AANN models in Figs. 3(a) and 3(b) demonstrate that a five layer AANN model is capable of capturing nonlinear subspaces. The ability of a five layer AANN model to perform clustering in the nonlinear subspace can be used to capture the complex distribution of the data in the input space. Fig.4(c) illustrates the probability surface obtained from a five layer AANN model for the artificial 2-D data shown in Fig.4(a). We notice that the five layer AANN model can be used as a nonparametric model to capture the distribution of the given data. The components spanning the nonlinear subspace captured by these models are known as nonlinear principal components or higher-order components. These models differ from other nonlinear methods such as principal curves in [10] due to the relationship between the weights of the network and the input data arising from the nonlinear activation function. In the next section we will show that the distribution capturing ability of the five layer AANN model can be exploited for the development of a text-independent speaker verification system [11], [12], [13], [14].

## IV. Speaker Verification using AANN Models

Speech corpus used in this study consists of SWITCHBOARD-2 databases of National Institute of Standards and Technology (NIST). These databases are used for the NIST-99 official speaker recognition evaluation [12]. The phase-2 SWITCHBOARD 2 database is used for background modeling, and hence referred to as *development data*. Performance of the speaker verification system is evaluated on the phase-3 database, which is referred to as *evaluation data*. The development data consists of 500 speakers (250 male and 250 female), and the evaluation data consists of 539 speakers (230 male and 309 female). The speaker sets of phase 2 and 3 databases are mutually exclusive.

Data provided for each speaker is conversational telephone speech collected from different sessions (conversations) sampled at 8000 samples/second. The training data consists of two minutes of speech data, collected from two different conversations over the same phone number. The use of the same phone number results in passing the speech data over the same handset and communication channel. Two different types of microphones (also referred as handsets) are used for collecting the speech data. They are carbon-button and electret. Performance of the speaker verification system is evaluated on the test utterances collected from different recording environments. The duration of the test utterance varies between 3 to 60 seconds. Each test utterance has 11 claimants, where the genuine speaker may or may not be one of the claimants. The gender of the claimant and the speaker of the test utterance is the same. There are no cross gender trials.

All the studies reported in this paper are performed on the male subset of 230 speakers with 1448 male test utterances of the evaluation data. Performance of the system is evaluated for the following three conditions:

(a) Matched condition: The training and testing data are collected from same phone number.

(b) Channel mismatch condition: The training and testing data are collected from different phone numbers, but it is ensured that the same handset type is used in both the cases. The use of different phone numbers results in passing the speech signal over different communication channels.

(c) Handset mismatch condition: The training and testing data are collected with different handset types.

Speaker information can be extracted using spectral features of the speech signal. The process of extraction speaker information is as follows. Speech signal is preemphasized using a difference operator. The differenced speech signal is segmented into frames of 27.5 ms using a Hamming window with a shift of 13.75 ms. The silence frames are removed using an amplitude threshold. A $16^{th}$ order linear prediction analysis is used to capture the properties of the signal spectrum [15]. The recursive relation between the predictor coefficients and the cepstral coefficients is used to convert the 16 predictor coefficients into 19 cepstral coefficients [16]. The cepstral coefficients obtained for each frame are linearly weighted to emphasize the peaks in the spectral envelope [17].

The speech signal transmitted over a telephone channel is distorted due to the filtering effect of the channel [18], [16]. Linear channel effects are compensated to some extent by removing the mean of the time trajectory of each cepstral coefficient. It has been shown that the mean subtraction improves the performance significantly when training and testing data are collected from different channels [18], [16]. But the recognition accuracy is reduced when the mean subtraction is used for a speaker verification system in which the training and testing data are collected from the same channel [18], [16].

Each speaker model is built by training an AANN model with the feature vectors extracted from the utterance of the speaker in the evaluation data. The structure of the AANN model is 19L 38N 14N 38N 19L, where L refers to a linear unit and N refers to a nonlinear unit. The integer value indicates the number of units in that particular layer. The network is trained using backpropagation learning algorithm in pattern mode. The initial weights of these models are adapted from a background model. The background model is an AANN model trained with feature vectors from a large number of speakers. This model is known as speaker-independent model or Universal Background Model (UBM). It represents the distribution of the feature vectors of several speakers [19], [20]. In our case, the UBM is trained with feature vectors of 250 male speakers of the development data. We have used 400 feature vectors per speaker to train the UBM. Each speaker model is derived by adapting the UBM.

During testing phase the feature vectors extracted from the test utterance are given to both the claimant model and the UBM to obtain the claimant score $S_c$ and the background score $S_b$ , respectively. The score of a model is defined as $\frac{1}{l}\sum_{i=1}^{l}\frac{||x_i-y_i||^2}{||x_i||^2}$ , where $x_i$ is the input vector of the model, $y_i$ is the output given by the model, and $l$ is the number of feature vectors of the test utterance. Since the claimant score is affected by the intra-speaker variability, linguistic content and recording environment of the test utterance, it is normalized with the score of the background model. The normalized score $S_n$ is obtained as $S_n = S_b - S_c$ [20].

False Acceptance (FA) and False Rejection (FR) are the two errors that are used in evaluating a speaker verification system. The trade off between FA and FR is a function of the decision threshold. Equal Error Rate (EER) is the value for which the error rates of FA and FR are equal [21]. A weighted sum of error rates of FA and FR is known as Detection Cost Function (DCF), and is given by $DCF = 0.99 * F_a + 0.01 * F_r$ [11], [22]. The percentage values of false acceptance $(F_a)$ and false rejection $(F_r)$ are chosen using a threshold such that the cost function is minimized.

For different test conditions, the performance of the AANN-based speaker verification system measured in terms of EER and DCF is shown in Table I under UBM and IBM columns. Here UBM corresponds to universal background model and IBM (which is explained later) to individual background model. The degradation in performance for mismatched conditions indicate that distortions introduced by different channels and handsets vary significantly. Fig.5 shows the distributions of genuine and impostor claimant scores. This figure clearly shows that the distributions of the genuine and impostor claimant scores overlap each other. The area of the overlapping region increases for mismatch conditions, particularly for handset mismatch condition.

TABLE I

PERFORMANCE OF SPEAKER VERIFICATION USING UBM AND IBM

| Environment between Training and Testing | EER | | DCF | |
|---|---|---|---|---|
| | UBM | IBM | UBM | IBM |
| Matched | 10.16% | 07.63% | 5.51 | 3.20 |
| Channel Mismatch | 28.64% | 27.39% | 9.21 | 8.25 |
| Handset Mismatch | 39.97% | 40.09% | 9.76 | 9.74 |

### *A. Significance of background model*

Normalization of the claimant scores using the scores of UBM may be affected by the parameters such as number of speakers, number of feature vectors per speaker and number of epochs used for training the UBM [20], [11]. The speaker-independent distribution captured by the UBM is sensitive to these parameters. Another way to address the issue of normalization is by using Individual Background Model (IBM) [13]. This approach makes use of the fact that for a given test utterance the genuine claimant may have a low score compared to the other claimant scores. To verify a claim, decision should be based upon the specified test

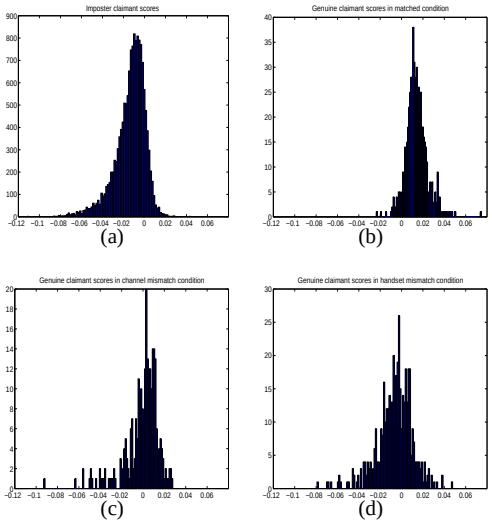



Fig. 5. (a) Distribution of impostor claimant scores. Distributions of the genuine claimant scores for the three cases: (b) matched condition (c) channel mismatch condition (d) handset mismatch condition.

utterance and the claimant model. Use of information about the other claimant models for the same test utterance may be difficult. To overcome this problem, pseudo-claimant models are used for normalization. These pseudo-claimant models are known as individual background models. A set of 92 pseudo-claimant models are generated from the development data. Each pseudo-claimant model is built by training an AANN with the feature vectors extracted from the test utterance of a particular speaker. This subset of 92 speakers belong to the male set of the development data. The utterances of 46 speakers are collected with electret handset and the utterances of remaining 46 speakers are collected with carbon-button handset. These speakers are selected arbitrarily without using any selection criteria such as cohorts [23]. The even mixture of male speaker utterances collected from both the handset types indicate the generation of gender-dependent and handset-balanced background model.

The speaker models of the evaluation data are generated independent of the pseudo-claimant models. In the testing phase, the feature vectors of the test utterance are given to the claimant model and to the pseudo-claimant models. The scores of all the pseudo-claimant models are sorted in ascending order. The rank $(R)$ of the claimant model is obtained. This rank is converted into a normalized score $(S_n)$ using, $S_n = (\rho+1)/R$, where $\rho$ is the population of the IBM. The use of this formula converts the rank of the claimant model into some form of confidence. The advantages of this simple approach are several: (1)The population of IBM is the only parameter to be chosen $apriori$. (2)No criteria is needed to select the pseudo-claimant models. (3)The claimant scores lie between 1 and $\rho$ + 1, thus the normalization of scores across the test utterances is obtained. (4)The knowledge of the best, the second best, etc., of the claimant score can be used to accept or reject the claim. Performance of the AANN-based speaker verification using IBM is shown in Table I. Comparison of the performance with UBM shows an improvement of 24.91$\%$ in EER for matched conditions. For mismatch conditions, the performance of IBM is similar to that of UBM.

It is to be recalled that the pseudo-claimants are randomly picked from the development data. The probability of a test utterance being close to one of the pseudo-claimant models is $1/\rho$, where $\rho$ is the population of IBM. Thus, the IBM population of $\rho$ implies that a FA rate of $1/\rho$ is incorporated by design. Fig.6 shows the effect of population of IBM on the performance of the speaker verification system measured in terms of EER and DCF. The decrease in FA due to increase in the population of IBM is observed from the DCF curves. The EER curves are not affected significantly, as it is a measure of possible trade-off between FA and FR. The use of IBM with large population will decrease FA, but at the cost of increase in the computation time to test the pseudo-claimant models.

The normalization procedure of IBM uses the scores of all the individual background models for deriving the rank. Instead, if we restrict our comparison of scores to the individual background models derived from the data collected over the same handset type (electret or carbon-button), then the performance of speaker verification system seems to improve marginally for the case of handset mismatch condition [13]. So we have used handset-dependent IBM for the studies reported in the following sections.

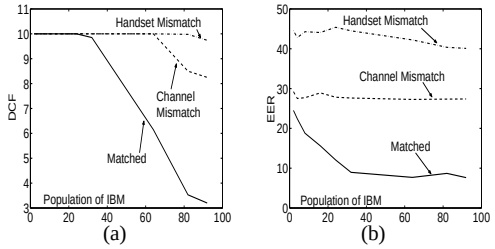



Fig. 6. Effect of population of IBM on (a) EER and (b) DCF.

### B. Structure of AANN model

As discussed in Section 2, the AANN model projects the input vectors onto the subspace spanned by the K components due to the compression layer. The structure of the AANN model used in the previous studies was 19L 38N 14N 38N 19L. Feature vectors extracted from the speech signal are projected onto the subspace spanned by K = 14 components to realize them at the output layer. The effect of changing the number (K) of these components on the performance of the speaker verification system is examined in this section. A series of experiments were conducted by systematically reducing the number (K) of units in the compression layer from K = 10 to 1. The results shown in Table II suggest that even for K = 4 case the system seems to give reasonably good performance in terms of EER.

TABLE II

PERFORMANCE OF SPEAKER VERIFICATION SYSTEM FOR DIFFERENT VALUES OF $K$ (NUMBER OF UNITS IN THE DIMENSION COMPRESSION HIDDEN LAYER).

| Environment between Training and Testing | EER | | | | | | |
|---|---|---|---|---|---|---|---|
| | $K=10$ | $K=8$ | $K=6$ | $K=4$ | $K=3$ | $K=2$ | $K=1$ |
| Matched | 6.48% | 6.45% | 6.73% | 6.69% | 8.34% | 10.45% | 14.67% |
| Channel Mismatch | 21.38% | 22.27% | 19.31% | 18.70% | 20.00% | 20.18% | 24.01% |
| Handset Mismatch | 34.43% | 30.53% | 31.71% | 30.26% | 28.65% | 29.47% | 31.36% |

### C. Channel variability

Due to the channel and handset effects, there will be a shift in the distributions of the training and testing data. Thus, in mismatch conditions, a trained model may give large error for the test data of the same speaker. The rejection of the genuine claim due to the channel or handset mismatch can be reduced either by suitable normalization of the score obtained by a speaker model, or by using a set of features not affected by channel and handset characteristics. In this section a method is proposed to normalize the score obtained by an AANN model. This method relies on the assumption that the shift in the distribution of test data of the same speaker may not be significant enough to label it as an impostor utterance. We show that this method yields significant improvement in the performance of AANN-based speaker verification system.

Let $\xi$ denote a speaker model and $I_i$ denote the score obtained by the model for an utterance $(i)$ which does not belong to the speaker. Let $\bar{I}_\xi$ denote the mean of $I_i$ .

$$\bar{I}_\xi = (1/l)\sum_{i=1}^{l} I_i$$

where $l$ is the number of other speakers. Let $S_\xi$ be the score obtained by a model for a given test utterance. We define the normalized score

$$N_\xi = S_\xi/\bar{I}_\xi = \frac{S_\xi}{(1/l)\sum_{i=1}^{l} I_i}$$

The normalized score indicates the closeness of $S_\xi$ to $\bar{I}_\xi$ . In mismatch condition, the value of $S_\xi$ may be large enough to reject the genuine speaker. But the value of $S_\xi$ in these cases may not be too close to $\bar{I}_\xi$ obtained by the same model, and hence the value of $N_\xi$ may be a better measure to accept or reject the claim. Using a set of 25 speakers utterances of NIST-97 database (the duration of each utterance is 1/2 minute), this normalization procedure is applied to the scores of the claimant and pseudo claimant models. The performance of the speaker verification system improved significantly as shown in Table III. The set of 25 speakers data used for normalization is randomly selected from the NIST-97 database. The use of NIST-97 database ensures that these utterances do not belong to any one of the claimant or pseudo-claimant models. The improved performance of the speaker verification system support our conjecture that the shift in the distribution of feature vectors of test data of the genuine speaker may not be large enough to be an impostor data for the same speaker model.

TABLE III

COMPARISON OF PERFORMANCE OF SPEAKER VERIFICATION SYSTEMS USING $S_\xi$ AND $N_\xi$ FOR AANN MODEL ($19L38N4N38N19L$) AND GMM.

| Environment between Training and Testing | EER | | |
|---|---|---|---|
| | AANN | | GMM |
| | $S_\xi$ | $N_\xi$ | |
| Matched | 6.69% | 5.81% | 5.01% |
| Channel Mismatch | 18.70% | 14.90% | 10.00% |
| Handset Mismatch | 30.26% | 23.05% | 21.00% |

Finally, the performance of a standard GMM-based speaker verification system is compared with our AANN-based speaker verification system for the same data, as shown in Table III. The results of the GMM-based OGI speaker verification system are taken from [12]. In the GMM-based approach, speaker models are built with 256 mixture components using 38 dimensional vectors, consisting of 19 melcepstral coefficients and 19 delta melcepstral coefficients. The time trajectories of the logarithmic filter-bank energies are smoothed over long (1 sec) segments using data-driven filters. The objective of this comparison is mainly to show that the AANN-based systems (with 19 weighted linear prediction cepstral coefficients) also provide a reasonable performance for speaker verification.

## V. AANN-HMM MODEL FOR SPEECH RECOGNITION

As opposed to earlier approaches to using neural network as classifier in hybrid models for speech recognition [24], it is useful to design the AANNs which can capture the distribution of given training data and associate a likelihood of a feature vector which belong to that class. Probabilistic Neural Networks and radial basis function neural networks can capture characteristics of data distribution. But both of these types have limitations. It's proven that Auto-associative neural networks can form a training error surface matching to distribution of the data [4].

We propose a hybrid model in which AANN is embedded into an Hidden Markov Model (HMM). Two different aspects of speech signal are taken care by two components of this

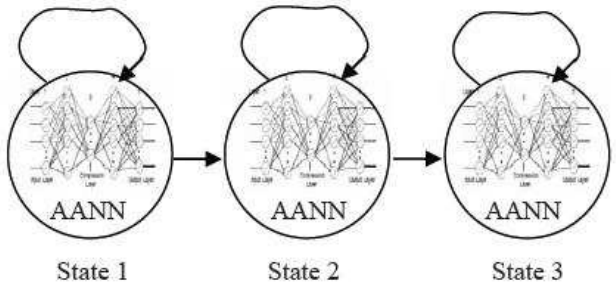



Fig. 7. Three state HMM using AANNs for modeling emissions

model. The temporal characteristics in speech are modeled by state transitions in HMM. While the state specific data distribution is modeled by an embedded AANN. The HMMs were used to model individual phones. As shown in Fig. 7 every phone HMM has three states. An AANN is embedded within each state which captures the emission probabilities of that state. The emission probability of input vector is calculated as follows: The feature vector is fed to AANN. The output vector has same dimensions. The Euclidean distance $d$ between input vector $x$ and output vector $y$ is calculated using following formula.

$$d(x,y) = \sqrt{\sum_{i=1}^{l}(x_i - y_i)^2}$$

The probability $p$ of the vector is given as -

$$p = e^{-d}$$

## VI. ISOLATED WORD RECOGNITION USING AANN-HMM FRAMEWORK

The AANN-HMM framework was studied for performance comparison in following experiment.

### A. Speech Database

The database used here consists of 90 isolated words spoken by single male speaker. These words were 0.3 to 0.7 sec. long. They are the instances of the digits one to nine spoken 10 times each. All the data was recorded at 16000 Hz. The Mel-Cepstral features are extracted using frame size of 160 and frame shift of 80. The MFCC features were variance normalized. Each feature vector is 39 dimensional.

### B. Structure of AANN

For our experiments we have used five layer AANN as shown in fig. 2 The five layer AANN performs Nonlinear Principle Component Analysis (NLPCA) [7]. The second and fourth layers of the network have more units than in input layer. The third layer has less number of units than in first or fifth layer. The activation function of layer 3 may be linear or nonlinear, but activation functions at layer 2 and 4 must be nonlinear[7]. Two experiments were performed with different structure of AANNs. In first experiment the structure of AANN was 39L 78N 12N 78N 39L where L stands for linear units while N stands or nonlinear units. In second experiment all hidden layer nodes were increased by factor of $1/3^{rd}$ . The structure used was 39 L 117 N 18 N 117 N 39 L. The choice of above architectures follows from [7] [4].

### C. Training AANN

Initially all the wave files were labeled using HMM based automatic segmenter. The feature vectors pertaining to every state of HMM were segregated. For every state of HMM, one AANN was trained using these segregated feature vectors. Every network was trained for 1000 iterations. The learning rate was kept 0.01 and momentum was 0.6. Training set consisted of $70\%$ data and test set consists of $30\%$ of data.

### D. Testing

Testing was done by comparing performance of GMM-HMM architecture with AANN-HMM framework. The HMMs were trained on 63 $(70\%)$ spoken utterances using Baum-Welch algorithm. The gaussian mixture model used here consisted to 2 gaussians. They are used to model state specific data distribution. These HMM models were then used to decode the data. The phone error rate and word recognition accuracy was calculated from obtained results. Then in next step, all the GMMs were replaced with AANNs which were already trained on state specific data. The decoding was again done on the same data. And the performances of both systems were studied.

### E. results

The performance obtained in first experiment is shown in Table IV. It's seen that here the performance of AANNs was comparable to that of GMMs. The phone error rate obtained after proper tuning of AANN parameters is shown in Table V. Here AANNs clearly performed much better than GMM models. Table VI shows word recognition accuracy obtained using both techniques. The word recognition accuracy obtained by both the techniques was same. But we found that AANN based technique led to better discriminating scores in decoding.

TABLE IV
RESULTS OBTAINED ON ISOLATED 90 WORDS DATABASE. THE AANN MODELS WERE TRAINED ON STATE SPECIFIC DATA. THE NETWORK STRUCTURE WAS 39L 78N 12N 78N 39L

| Decoding Technique | Phone error rate (in percentage) |
|---|---|
| GMM embedded in HMM | 14.6% |
| AANN embedded in HMM | 14.9% |

## VII. CONCLUSION

In this paper, we have explored AANN models as an alternative to GMM for speaker verification and speech recognition studies. The relationship between the training error surface and the input data distribution is used to demonstrate the distribution capturing ability of a five layer

TABLE V

RESULTS OBTAINED ON ISOLATED 90 WORDS DATABASE. THE NETWORK STRUCTURE WAS 39 L 117 N 18 N 117 N 39 L

| Decoding Technique | Phone error rate (in percentage) |
|---|---|
| GMM embedded in HMM | 14.6% |
| AANN embedded in HMM | 13.4% |

TABLE VI

WORD RECOGNITION ACCURACY

| Decoding Technique | Word Recognition Accuracy (in percentage) |
|---|---|
| GMM embedded in HMM | 100% |
| AANN embedded in HMM | 100% |

AANN model. These AANN models differ from other nonlinear extensions of PCA such as principal curves [10]. The number of free parameters in the form of weights are less compared to the number of mixture components in GMM [12]. Studies on background models indicate that the normalization procedure of IBM performs better than that of UBM for AANN-based speaker verification system. The channel and handset effects can be reduced to some extent by decreasing the units in the dimension compression layer and by using the proposed normalize score. Throughout the studies, we have used 19-dimensional static weighted linear prediction cepstral coefficients without any post processing such as smoothing the trajectories [25], [26]. Performance of the AANN-based speaker verification system reported in this paper is comparable with the performance of GMM-based speaker verification systems for matched conditions. The results of isolated word recognition also suggest that AANN-HMM models are useful for speech recognition. Further experiments have to be conducted using AANN-HMM models for large vocabulary continuous speech recognition.

## ACKNOWLEDGMENT

The authors would like to thank Dr. S. Gangashetty and Mr. A. Gopalakrishna for useful discussions and correction of the draft.

## REFERENCES

[1] B. Yegnanarayana, *Artificial Neural Networks*. New Delhi: Prentice-Hall of India, 1999.

[2] H. Bourlard and Y. Kamp, "Auto-association by multilayer perceptrons and singular value decomposition," *Biol. Cybernet.*, vol. 59, pp. 291–294, 1988.

[3] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural networks: Theory and Applications*. New York: John Wiley & Sons Inc., 1996.

[4] S. Haykin, *Neural networks: A comprehensive foundation*. New Jersey: Prentice-Hall Inc., 1999.

[5] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 4–37, Jan. 2000.

[6] M. Bianchini, P. Frasconi, and M. Gori, "Learning in multilayered networks used as autoassociators," *IEEE Trans. Neural Networks*, vol. 6, pp. 512–515, Mar. 1995.

[7] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE*, vol. 37, pp. 233–243, Feb. 1991.

[8] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *IEEE Trans. Neural Networks*, vol. 2, pp. 53–58, 1989.

[9] B. Yegnanarayana, S. P. Kishore, and A. V. N. S. Anjani, "Neural network models for capturing probability distribution of training data," in *Int. Conference on Cognitive and Neural Systems*, (Boston), p. 6 (A), 2000.

[10] E. C. Malthouse, "Limitations of nonlinear PCA as performed with generic neural networks," *IEEE Trans. Neural Networks*, vol. 9, pp. 165–173, Jan. 1998.

[11] NIST, "Speaker recognition workshop notebook," *Proc. NIST 2000 Speaker Recognition Workshop, University of Maryland, USA*, Jun 26-27 2000.

[12] NIST, "Speaker recognition workshop notebook," *Proc. NIST 1999 Speaker Recognition Workshop, University of Maryland, USA*, Jun 3-4 1999.

[13] S. P. Kishore and B. Yegnanarayana, "Speaker verification: Minimizing the channel effects using autoassociative neural network models," in *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, (Istanbul), pp. 1101–1104, 2000.

[14] M. S. Ikbal, *Autoassociative Neural Network Models for Speaker Verification*. MS dissertation, Indian Institute of Technology, Department of Computer Science and Engg., Madras, May 1999.

[15] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, pp. 561–580, Apr. 1975.

[16] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 29, pp. 254–272, Apr. 1981.

[17] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[18] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Amer.*, vol. 55, pp. 1304–1312, Jun. 1974.

[19] D. A. Reynolds, "Comparison of background normalization methods for text–independent speaker verification," in *Eurospeech*, (Greece), pp. 963–966, 1997.

[20] H. Misra, *Development of a Mapping Feature for Speaker Recognition*. MS dissertation, Indian Institute of Technology, Department of Electrical Engg., Madras, May 1999.

[21] J. Oglesby, "What's in a number ? Moving beyond the equal error rate," *Speech Comm.*, vol. 17, pp. 193–208, Aug. 1995.

[22] G. R. Doddington, M. A. Pryzbocki, A. F. Martin, and D. A. Reynolds, "The NIST speaker recognition evaluation - Overview, methodology, systems, results, perspective," *Speech Comm.*, vol. 31, pp. 225–254, Jun. 2000.

[23] R. A. Finan, A. T. Sapeluk, and R. I. Damper, "Imposter cohort selection for score normalization in speaker verification," *Pattern Recognition Lett.*, vol. 18, pp. 881–888, 1997.

[24] K. Kirchhoff, "Dynamic classifier combination in hybrid speech recognition systems using utterance-level confidence values," *ICASSP*, vol. 2, pp. 693–696, 99.

[25] S. van Vuuren, *Speaker Recognition in a Time-Frequency Space*. PhD dissertation, Orgeon Graduate Institute of Science and Technology.

[26] N. Malayath, *Data-Driven Methods for Extracting Features from Speech*. PhD dissertation, Orgeon Graduate Institute of Science and Technology.