Impostor Modeling and Effects of Neural Network Structure on the Performance of Speaker Verification

Abstract

In this paper, we present our work on text-independent speaker verification using linear prediction cepstral coefficients as feature vectors and using artificial neural networks for pattern matching. We specifically address the issue of imposter modeling technique in the context of neural network classifier based speaker verification system. An approach for modeling the imposter characteristics is proposed. Performance of the speaker verification system for different structures of the neural network is compared. We present our observations based on the above experiments.

Keywords: speaker verification; neural networks; feature extraction; imposter modeling;

I. Introduction to Speaker Recognition

Speaker recognition is the process of automatically recognizing the speaker by using speaker specific information present in the speech signal [1] [2] [3] [4]. The generic term speaker recognition refers to the task of speaker identification and speaker verification. The task of speaker verification is to verify the identity claim of the test voice, whereas, speaker identification is to choose the closest match of the test voice with one of the registered speakers. The difference between identification and verification is the number of decision alternatives. In identification, the number of decision alternatives is equal to the number of speakers. However, the identification system can also have a rejection alternative, which is to decide that the test voice belong to none of the registered speakers. But in verification systems there are only two choices, accept or reject, regardless of the
number of users. The performance of an identification system would decrease with the increase in the population of speakers, while it has almost no effect on speaker verification system.

Speaker recognition can be performed in a text-dependent or text-independent mode. In a text-dependent speaker recognition system, a restriction is imposed on the speaker to utter some fixed words or sentences. Such restriction is not imposed in a text-independent speaker recognition system. In this paper, we address some issues related to the development of a text-independent speaker verification system and focus in particular the issue of building an imposter model.

The organization of this paper is as follows: Section II presents an overview of speaker verification system. Section III describes the exaction of features to represent the speaker information. Section IV explains the reason behind choosing neural network for classification and give reasons for the specific architecture chosen in this paper. The implementation of the speaker verification system is described in Section VI and the results are discussed in Section VII.

II. SPEAKER VERIFICATION SYSTEM

A typical implementation of a speaker verification system is shown in Figure 1. Speaker verification by a machine involves three stages. They are, (1) extraction of features to represent the speaker information present in the speech signal, (2) modeling of speaker features, and (3) decision logic to implement the verification task.

The primary task in a speaker recognition system is to extract features capable of representing the speaker information present in the speech signal. The distribution of these
feature vectors is estimated using parametric models or discriminative models such as neural networks are trained. During verification the test utterance is matched with the model of the speaker whose identity is claimed. The match of the claimed model for the test utterance is compared with a threshold to accept or reject the claim.

In verification procedure, we take a binary decision of accepting or rejecting a speaker. In making this decision, two kinds of errors occur. They are False Acceptance (FA) and False Rejection (FR). False acceptance refers to the case of accepting an impostor, while the false rejection refers to the case of rejecting a genuine speaker. A false acceptance of an impostor may cause severe damage to the objective of a secured application such as banking transaction. On the other hand, frequent false rejections (FR) of a genuine speaker would have an annoying effect on the user. Performance of a speaker verification system is typically measured in terms of trade off provided between FA and FR.

III. Features to Represent Speaker Information

Speaker information can be extracted both at the segmental and suprasegmental levels. The segmental features are the features extracted from short (10-30 ms) segments of speech signal. Some of the segmental features are linear prediction cepstral coefficients, melcepstral coefficients, log spectral energy values, etc. [5]. These features represent the short-term spectra of the speech signal. The spectrum of a speech segment is determined primarily by the shape of the vocal tract. The spectral information of the same sound uttered by two persons may differ both in the shapes of their vocal tracts and in the manner in which they produce speech [3]. Comparative studies between spectral features and other features such as the fundamental frequency show that the spectral features seem to provide better discrimination among speakers [6]. In this work, the spectral features are represented by linear prediction cepstral coefficients [7].
A. Preprocessing of Speech Signal

The speech signal $x(n)$ is preemphasized to counteract the spectral roll-off due to the glottal closure in voiced speech [8].

$$x(n) = x(n) - \alpha x(n-1), \text{ where } \alpha = 1.$$ 

Differencing the speech signal in time domain, multiplies the signal spectrum with linear filter to give emphasis to the high frequency components [5].

B. Extraction of Linear Prediction Cepstral Coefficients

The characteristics of the speech signal are assumed to be stationary over a short duration of time (between 10-30 ms) [5]. The differenced speech signal is segmented into frames of 20 ms using a Hamming window with a shift of 10 ms. A 16th order Linear Prediction (LP) analysis is used to capture the properties of the signal spectrum [9]. The recursive relation between the predictor coefficients and cepstral coefficients is used to convert the 16 LP coefficients into 19 LP cepstral coefficients (Appendix A).

IV. Neural Network Classifiers for Speaker Verification Task

Artificial neural network (ANN) models consist of interconnected processing units, where each unit represent the model of an artificial neuron, and the interconnection between two units has a weight associated with it. ANN models with different topologies perform different pattern recognition tasks [10] [11] [12]. For example, a feedforward neural network can be designed to perform the task of pattern classification or pattern mapping, whereas a feedback network can be developed to perform the task of pattern storage or pattern environment storage [11]. In the case of a feedforward network, the weights are adjusted so as to realize the global minimum of the total error for the training data in the weight
space. The weights are adjusted to reduce the error, and hence the training vectors with large error will influence the adjustment to a large extent. The objective in these supervised learning tasks is to achieve generalization from the training data, so that the classification or mapping error for test data is low [13]. The capabilities of these models to discriminate between patterns of different classes motivated us to use them for the task of speaker verification.

Each speaker model is built by training a neural network to discriminate between feature vectors of speaker set and impostor set (see Section 5 for imposter modeling). A five layer neural network, for example $19L38S6L38S2S$, is typically used in this work. Here $L$ refers to a linear unit and $S$ refers to a nonlinear unit. The integer value denotes the number of units in that particular layer. These networks are shown to perform nonlinear discriminant analysis over the feature vectors [12]. The dimension compression layer in the middle is useful to project the data onto a lower dimensional space. As the network gets trained to discriminate using these lower dimensional vectors, significant features are represented in this lower dimensional space like in the discriminant analysis [12] [14]. The network is trained using Backpropagation algorithm in pattern mode [10]. In back propagation learning algorithm the error estimated for each feature vector is propagated backwards towards the input layer, and the weights of all the connections are adjusted in order to minimize the error.

V. Verification Procedure

During verification, the feature vectors extracted from the test utterance is given to the claimant model. A claim is accepted if the average of the output at the speaker node is higher than that of the value at the impostor node. It is known that the performance of the speaker verification system is highly dependent on the imposter modeling (imposter...
model is also referred to as cohort model) [15] [16] [17] [18] [19] [20]. The approach we followed for the selection of impostor data is explained below.

A. Impostor Modeling Algorithm

*Step 1:* Extract the feature vectors for training data of all the speakers. Let $S$ denote the set of speakers.

*Step 2:* To build a impostor model for each speaker say $X$ in the set $S$, repeat the steps 3 to 5.

*Step 3:* For each feature vector $X_i$ of speaker $X$,

*Step 4:* Calculate the Euclidean distance between the feature vector $X_i$ and all of the feature vectors of the speakers in the set $S - X$. The vector that has the least distance is included in the impostor feature set of the speaker $X$.

*Step 5:* Repeat step 4 for all the feature vectors of the speaker $X$.

B. Significance

The set of impostor feature vectors obtained in this fashion is used as impostor data to train a neural network classifier. Note that feature vectors present in the impostor set of a speaker $X$ can be from any of the other speakers. In the worst case, we have the impostor feature vectors of a single speaker, and in other cases, we may have a uniformly distributed selection. The selection of the impostor feature set depends on the nature of the feature vectors of the speaker $X$. 

In traditional impostor modeling methods the nearest impostor (speaker) is calculated [21] [22] [17] [23]. In some methods an arbitrary number of randomly selected speakers are used as impostors. But by using the method presented above, we focus more on impostor feature selection, rather than on the impostor speaker. Our selection procedure of impostor set might include feature vectors of a single speaker, or multiple speakers which is entirely dependent on the nature of speaker data (for which the impostor data set is being selected). The advantage of this method is that, it might model almost all of the impostor feature space surrounding the intended speaker. Thus covers all possible impostor attacks whereas, the method in which the nearest impostor (speaker) is calculated only a small impostor set is modeled and need a careful selection to cover all possible impostor attacks.

VI. Performance of speaker verification system

The speaker verification system is built for 50 users. The duration of the training speech data collected from each user is two minutes, while the duration of six testing speech samples collected from each user is 30 seconds. Speech data collection is done in a lab environment. Recording is done using multimedia facilities of a typical computer and the speech signal is sampled at 8000 Hz. Training and test data are collected from the same microphone. Speaker models are built for each user as mentioned in the above sections.

A total of 300 genuine tests and 14700 impostor tests are conducted. The performance of the speaker verification systems is evaluated for different neural network structures, trained for 1000 and 2000 epochs. The results are shown in Table I and Table II. The first column shows the neural network structure. The integer value denotes the number of nodes in the particular layer. Character L denote a linear unit while S for nonlinear
unit with sigmoid activation. The number of nodes in the middle layer accounts for the compression in the neural network. Figures 2 and 3 show the training error curves graphs of one of the speaker models for 1000 and 2000 epochs respectively.

VII. Observations

A. Comparison of 3-layered and 5-layered networks

The 3-layered networks are the typical neural networks used for pattern classification. We added a compression layer and built the 5-layered network following the capabilities of these networks as discussed in Section IV. The structure of these neural networks is given in Table I and II. The comparison shows a clear better performance by the 5-layered networks over the typical 3-layered networks.

B. Comparison for 1000 and 2000 epochs

Our observations have shown that as number of epochs for which a network is trained is increased from 1000 to 2000, false acceptance has increased while the false rejection has decreased. We think that with the increase in the number of epochs the model might learn better about the genuine speaker’s data, but the increase in false acceptance rate might be due to some outliers present in the speaker’s data set.

C. Comparison for different compression levels

Our observations have shown that the as the compression is increased, the overall performance diminished initially till some point and then it improved before again dropping. The initial reduction in performance is due to the loss of some speaker features due to compression. We think the ensuing improvement is due to the removal of some noise features (that diminish the performance) due to compression. The effect of compressions is graphically shown in Figure 4. Here, the graph is plotted with compression on the X-axis.
and overall percentage error i.e. \((\text{false rejection} + \text{false acceptance}) / 2\) on the Y-axis.

VIII. Conclusion

In this paper, we presented a text-independent speaker verification system. Speaker models were built using neural network classifiers. We presented an approach for imposter modeling which tends to select a imposter feature set from one or many imposter speakers depending on the nature of the speaker data. We evaluated the performance of the speaker verification system for three layer and five layer network structures. We found that the discrimination capabilities of a five layer network with dimension compression layer performs better than that of the three layer network. The speaker verification was evaluated for a speech database of 50 users and the best performance of FA 2.7% and FR 2.3 % was found to be obtained from the the five layer network with four nodes in the dimension compression layer.

Our future work is focussed on experiments with switchboard corpus, and in developing a robust speaker verification system over telephone.

References

In LP analysis of speech, an all-pole model is assumed for the system producing speech signal $s(n)$. A $p^{th}$ order all-pole model assumes that sample value at time $n$ can be approximated by linear combination of past $p$ samples. i.e.,

**APPENDIX**

A: LINEAR PREDICTION ANALYSIS

In LP analysis of speech, an all-pole model is assumed for the system producing speech signal $s(n)$. A $p^{th}$ order all-pole model assumes that sample value at time $n$ can be approximated by linear combination of past $p$ samples. i.e.,
\[ s(n) \approx \sum_{k=1}^{p} a_k s(n-k) \quad (A.1) \]

If \( \hat{s}(n) \) denotes the prediction made by the all-pole model then, the prediction error is given by,

\[ e(n) = s(n) - \hat{s}(n) = s(n) - \sum_{k=1}^{p} a_k s(n-k) \quad (A.2) \]

For a speech frame of size \( m \) samples, the mean square of prediction error over the whole frame is given by,

\[ E = \sum_{m} e^2(m) = \sum_{m} [s(m) - \sum_{k=1}^{p} a_k s(m-k)]^2 \quad (A.3) \]

Optimal predictor coefficients will minimize this mean square error. At minimum value of \( E \),

\[ \frac{\partial E}{\partial a_k} = 0, \quad k = 1, 2, \ldots p. \quad (A.4) \]

Differentiating Eqn A.3 and equating to zero we get,

\[ \mathbf{R} \mathbf{a} = \mathbf{r} \quad (A.5) \]

where, \( \mathbf{a} = [a_1 \ a_2 \cdots a_p]^T \), \( \mathbf{r} = [r(1) \ r(2) \cdots r(p)]^T \), and \( \mathbf{R} \) is a Toeplitz symmetric autocorrelation matrix given by,

\[ \mathbf{R} = \begin{bmatrix}
  r(0) & r(1) & \cdots & r(p-1) \\
  r(1) & r(0) & \cdots & r(p-2) \\
  \vdots & \vdots & \ddots & \vdots \\
  r(p-1) & \cdots & r(0) 
\end{bmatrix} \quad (A.6) \]

Eqns A.5 can be solved for prediction coefficients using Durbin’s algorithm as follows:
\( E^{(0)} = r[0] \) \hspace{1cm} (A.7)

\[ k_i = \frac{r[i] - \sum_{j=1}^{L-1} \alpha_j^{(i-1)} \cdot r[i-j]}{E^{(i-1)}} \quad 1 \leq i \leq p \] \hspace{1cm} (A.8)

\[ \alpha_i^i = k_i \] \hspace{1cm} (A.9)

\[ \alpha_j^i = \alpha_j^{(i-1)} - k_i \cdot \alpha_{i-j}^{(i-1)} \] \hspace{1cm} (A.10)

\[ E^{(i)} = (1 - k_i^2) \cdot E^{(i-1)} \] \hspace{1cm} (A.11)

The above set of equations are solved recursively for \( i = 1, 2, \ldots, p \). The final solution is given by

\[ a_m = \alpha_m^{(p)} \quad 1 \leq m \leq p \] \hspace{1cm} (A.12)

where, \( a_m \)'s are linear predictive coefficients (LPCs).

Cepstral coefficients can be extracted from the predictor coefficients using recursive algorithm as follows.

\[ c_0 = \ln \sigma^2 \] \hspace{1cm} (A.13)

\[ c_m = a_m + \sum_{k=1}^{m-1} \frac{k}{m} \cdot c_k \cdot a_{m-k} \quad 1 \leq m \leq p \] \hspace{1cm} (A.14)

\[ = \sum_{k=1}^{m-1} \frac{k}{m} \cdot c_k \cdot a_{m-k} \quad m > p \] \hspace{1cm} (A.15)
### TABLE I
Performance of the speaker verification system when the network is trained for 1000 epochs.

<table>
<thead>
<tr>
<th>Network Structure</th>
<th>False Acceptance (in %)</th>
<th>False Rejection (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>19L 38S 2S</td>
<td>2.1</td>
<td>5.6</td>
</tr>
<tr>
<td>19L 32S 4L 32S 2S</td>
<td>1.5</td>
<td>6</td>
</tr>
<tr>
<td>19L 38S 4L 38S 2S</td>
<td>1.9</td>
<td>3.6</td>
</tr>
<tr>
<td>19L 38S 6L 38S 2S</td>
<td>1.5</td>
<td>6</td>
</tr>
<tr>
<td>19L 38S 8L 38S 2S</td>
<td>1.9</td>
<td>4.6</td>
</tr>
<tr>
<td>19L 38S 10L 38S 2S</td>
<td>1.6</td>
<td>6.6</td>
</tr>
<tr>
<td>19L 38S 12L 38S 2S</td>
<td>1.7</td>
<td>6.3</td>
</tr>
</tbody>
</table>

### TABLE II
Performance of the speaker verification system when the network is trained for 2000 epochs.

<table>
<thead>
<tr>
<th>Network Structure</th>
<th>False Acceptance (in %)</th>
<th>False Rejection (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td>19L 38S 2S</td>
<td>5.1</td>
<td>2.6</td>
</tr>
<tr>
<td>19L 32S 4L 32S 2S</td>
<td>2.6</td>
<td>2.6</td>
</tr>
<tr>
<td>19L 38S 4L 38S 2S</td>
<td>2.7</td>
<td>2.3</td>
</tr>
<tr>
<td>19L 38S 6L 38S 2S</td>
<td>2.5</td>
<td>4</td>
</tr>
<tr>
<td>19L 38S 8L 38S 2S</td>
<td>2.6</td>
<td>1.6</td>
</tr>
<tr>
<td>19L 38S 10L 38S 2S</td>
<td>2.3</td>
<td>2.4</td>
</tr>
<tr>
<td>19L 38S 12L 38S 2S</td>
<td>1.8</td>
<td>3</td>
</tr>
</tbody>
</table>
Fig. 1. Block diagram of a speaker verification system
Fig. 2. Training error curve of a speaker model for 1000 epochs.
Fig. 3. Training error curve of a speaker model for 2000 epochs.
Fig. 4. Performance of speaker verification system for different compression levels