VOICE RECOGNITION

RICHARD L. KLEVANS
ROBERT D. RODMAN

Voice Recognition

Richard L. Klevans
Robert D. Rodman

Artech House
Boston • London

Library of Congress Cataloging-in-Publication Data
Klevans, Richard L.
Voice recognition / Richard L. Klevans, Robert D. Rodman.
p. cm. — (Artech House telecommunications library)
Includes bibliographical references and index.
ISBN 0-89006-927-1 (alk. paper)
1. Speech processing systems. 2. Automatic speech recognition.
3. Voiceprints. 4. Natural language processing (Computer science)
I. Rodman, Robert D. II. Title. III. Series.
TK7882.2.S65K55 1997
006.4'54—dc21 97-30792
CIP

British Library Cataloguing in Publication Data
Klevans, Richard L.
Voice recognition
1. Automatic speech recognition
I. Title. II. Rodman, Robert D.
006.4'54

ISBN 0-89006-927-1

Cover design by Jennifer L. Stuart

685 Canton Street
Norwood, MA 02062

International Standard Book Number: 0-89006-927-1
Library of Congress Catalog Card Number: 97-30792

10 9 8 7 6 5 4 3 2 1

Contents

Chapter 1 Introduction 1

Speech Synthesis 1

Speech Recognition 3

Speaker Classification 4

Areas of Application for Voice Recognition 6

Design Tradeoffs in Voice Recognition 7

Text-Dependent Versus Text-Independent 7

Ideal Recording Environment Versus Noisy Environment 8

Speaker Verification Versus Speaker Identification 9

Real-Time Operation Versus Off-Line Operation 9

Regarding This Book 10

Intended Readers 10

What Is Covered 11

Why? 12

References 13

Chapter 2 Background of Voice Recognition 15

Voiceprint Analysis 16

Parameter Extraction 20

The Parameter Extraction Process 20

Types of Parameters 21

Evaluation of Parameters 27

Distance Measures 29

Pattern Recognition 32

Voice Recognition in Noisy Environments 54
Summary 56
References 57

Chapter 3 Methods of Context-Free Voice Recognition 61
Voice Recognition in Law Enforcement 61
Forensic Recognition Classification 62
Ideal Voice Recognition 64
A Segregating Voice-Recognition System 72
System Tasks 75
Channel Variation Compensation 94
Software Implementation 99
Logistics of Forensic Speaker Identification 101
Summary 105
References 105

Chapter 4 Experimental Results 107
Test Utterance Length Experiments 107
Large Population Results 112
Filtered Data Test 114
Channel Compensation Tests 116
Average Filter Compensation Technique Experiment 117
Rehumanizing Filter Technique Experiment 119
Secondary Parameters 121
Secondary Parameter Usage 130
Effects of Varying the Cutoff Value 131
Best-Case Secondary Parameter Usage 132
Mock Forensic Cases 135
SBI Case 1 136
SBI Case 2 144
SBI Case 3 147
Summary 148
References 149

Chapter 5 The Future of Context-Free Voice Recognition 151
Rehumanizing Filter Technique Tests 151
Voice-Recognition Databases 153
Medium-Term Goals 156
Long-Term Goals 157
Other Applications 158
Summary 159
References 160
Chapter 6 Conclusions 161
About the Authors 165
Index 167

Background of Voice Recognition 2

This chapter will present a review of the research in the area of voice recognition. Initially, research in this area concentrated on determining whether speakers' voices were unique or at least distinguishable from those of a group of other speakers. In these studies, manual intervention was necessary to carry out the recognition task. As computer power increased and knowledge about speech signals improved, research became aimed at fully automated systems executed on general-purpose computers or specially designed computer hardware.

Voice recognition consists of two major tasks: *feature extraction* and *pattern recognition*. Feature extraction attempts to discover characteristics of the speech signal unique to the individual speaker. The process is analogous to a police description of a suspect, which typically lists height, weight, skin color, facial shape, body type, and any distinguishing marks or disfigurements. Pattern recognition refers to the matching of features in such a way as to determine, within probabilistic limits, whether two sets of features are from the same or different individuals. In this chapter, we will discuss research related to these tasks. The chapter will conclude with a short description of methods for dealing with noise in voice-recognition systems.

VOICEPRINT ANALYSIS

The first type of automatic speaker recognition, called *voiceprint analysis*, was begun in the 1960s. The term *voiceprint* was derived from the more familiar term *fingerprint.* Researchers hoped that voiceprints would provide a reliable method for uniquely identifying people by their voices, just as fingerprints had proven to be a reliable method of identification in forensic situations.

Voiceprint analysis was only a semiautomatic process. First, a graphical representation of each speaker's voice was created. Then, human experts manually determined whether two graphs represented utterances spoken by the same person. The graphical representations took one of two forms: a *speech spectrogram* (called a *bar voiceprint* at the time)—see Figure 2.1—or a contour voiceprint [1]. The former, the more commonly used form, consists of a representation of a spoken utterance in which time is displayed on the abscissa, frequency on the ordinate, and spectral energy as the darkness at a given point.

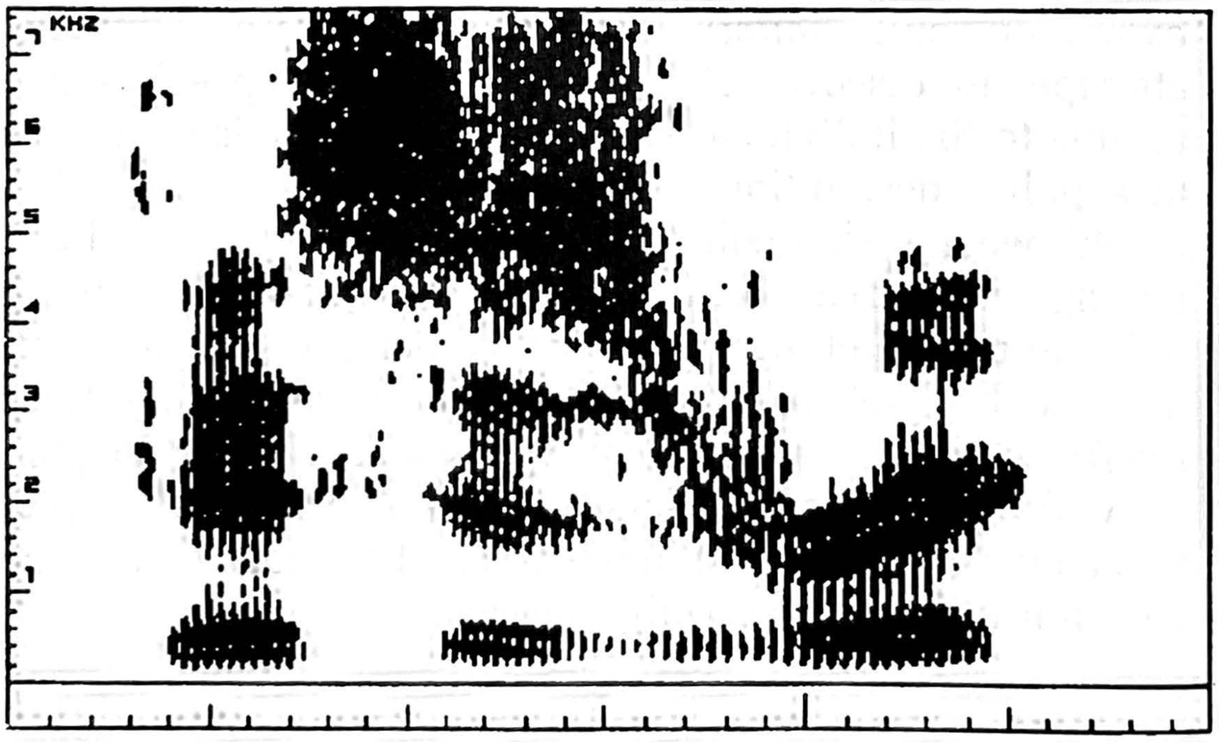

Figure 2.1 Spectrogram of author saying, "This is Rick."

Prior to a voiceprint identification attempt, spectrograms would have been produced by a *sound spectrograph* from recordings of the speakers in question. Typically, the input data for voiceprint analysis consisted of recordings of utterances of 10 commonly used words—such as "the," "you," and "I"—from each speaker in the set to be identified. These 10 words can be thought of as roughly analogous to the 10 fingers used in fingerprint analysis. Human experts determined the identity of speakers by visually inspecting the spectrograms of a given word spoken by several known speakers and comparing those to a spectrogram of the same word spoken by an unknown speaker.

The experts looked for features of the spectrograms that best characterized each speaker. Some commonly used features were absolute formant frequency, formant bandwidths, and formant trajectories. Formants are bands of energy in the spectrogram that are related to the resonant frequencies of the speaker's vocal tract. Therefore, formant locations and trajectories are related to the fixed shapes of the speaker's vocal tract as well as the way in which the speaker manipulates his or her vocal tract during utterances.

The voiceprint identification method described above had many flaws. First, identification was based on the subjective judgment of human experts. Second, multiple voiceprints of a word spoken by one person can vary as much as voiceprints by two different speakers speaking the same word. This phenomenon introduces the general problem of interspeaker versus intraspeaker variance that is of primary concern for all voice-recognition research. A final concern was the vulnerability of the voiceprint identification process to impostors that had been trained to mimic other speakers. Thus, researchers were uncertain about the worth of voiceprint identification. In the 1960s, a number of experiments were performed that addressed these issues. L.G. Kersta reported an error rate of 1% for 2000 identification attempts with populations of 9 to 15 known speakers for each unknown [1]. Richard Bolt summarized the results of several similar studies with widely varying error rates [2]. Some studies reported error rates as high as 21% and others as low as 0.42%. Bolt criticized all the studies as being artificial inasmuch as the experiments consisted of matching tasks. If used as evidence in court, he pointed out, the analysis would be a verification task, in which the experts would have to decide from two sets of voiceprints (one set from the accused, one set from the unknown) whether or not the accused and the unknown person were the same.

The inconsistent experimental evidence caused experts to disagree about the viability of voiceprints. Kersta's original study led him to believe that voiceprint analysis could be as effective as fingerprint analysis:

> Other experimental data encourages me to believe that unique identifications from voiceprints can be made. Work continues, there being questions to answer and problems to solve. . . . It is my opinion, however, that identifiable uniqueness does exist in each voice, and that masking, disguising, or distorting the voice will not defeat identification if the speech is intelligible [1].

A study by Richard Bolt and others a few years later reached the opposite conclusion:

> Fingerprints show directly the physical pattern of the finger producing them, and these patterns are readily discernible. Spectrographic patterns and the sound waves that they represent are not, however, related so simply and directly to vocal anatomy; moreover, the spectrogram is not the primary evidence, but only a graphic means for examining the sounds that a speaker makes [2].

Between 1970 and 1985, the Federal Bureau of Investigation (FBI) made extensive use of spectrogram identification, the results of which were analyzed by Bruce Koenig [3]. The FBI formulated a 10-point procedural protocol dictating how voice comparison was to take place. The Bureau insisted on high-quality recordings, from which spectrograms in the frequency range of 0–4,000 Hz were to be made. FBI technicians examined twenty words pronounced alike (supposedly) for similarities and differences, and these results were supplemented by aural comparisons made by repeatedly and simultaneously playing the two voice samples on separate tape recorders. In the end, the examiner determined whether two exemplars were "no or low confidence," "very similar," or "very dissimilar," and these results were confirmed by two other examiners. Identification of an individual was only claimed in the presence of a sufficiently high percentage of "very similar" determinations.

A survey of the results of 2,000 voice comparisons found that in two-thirds (1,304) of the cases, examiners had no or low confidence; in 318 cases there was a positive identification; and in 378 cases a positive elimination. There was one false identification and two false eliminations. Koenig observes:

> Most of the no or low confidence decisions were due to poor recording quality and/or an insufficient number of comparable words. Decisions were also affected by high-pitched voices (usually female) and some forms of voice disguise [3].

The attempt to use voiceprints in a forensic setting left unanswered many questions about the practicality of using voice to identify individuals uniquely. It became clear that research must be focused on the following goals:

1. Automating the recognition procedures;
2. Freeing recognition procedures from dependency on fixed words;
3. Standardizing testing so improvements in procedures could be measured;
4. Handling noisy signals;
5. Coping with unknown and/or inadequate channels;
6. Dealing with intervoice and intravoice variation both natural and artificial (i.e., disguised voice).

Advances in digital computer hardware in the mid 1980s made achievement of these goals seem possible. The six points enumerated above were the basis of many research programs in voice recognition during subsequent years. These programs will be discussed in the remaining sections of this chapter. Since voice-recognition research progressed along many different paths after the 1960s, a historical perspective is not fully appropriate. Thus,we have partitioned the discussion of research by task: *parameter extraction, distance measurements, pattern recognition techniques*, and *special considerations*.

PARAMETER EXTRACTION

In this section, we will discuss methods of extracting information from speech waveforms. Parameter or feature extraction consists of preprocessing an electrical signal to transform it into a usable digital form, applying algorithms to extract only speaker-related information from the signal, and determining the quality of the extracted parameters.

The Parameter Extraction Process

The preprocessing required by voice-recognition systems uses digital signal processing (DSP) methods that are common to all computer speech systems. First, the sound wave created by an individual's speech is transduced into an analog electrical signal via a microphone. The electrical signal is sampled and quantized, resulting in a digital representation of the analog signal. Typical representations of signals for voice-recognition systems are sampled at rates of between 8 and 16 kHz with 8 to 16 bits of resolution [4].

The digital signal may then be subjected to conditioning. For example, bandpass filtering can be used for attenuating parts of the spectrum that are corrupted with additive noise. Spectral flattening can be used to improve the pitch extraction process by compensating for the effect of the vocal tract on the excitation signal created by the vibrating vocal folds. Many other conditioning techniques have been reported. After the signal has been conditioned, it may then be used as input to an algorithm for parameter extraction (Figure 2.2).

Types of Parameters

The most basic type of parameters used for voice recognition are either quantifiable by a human listener, such as pitch or

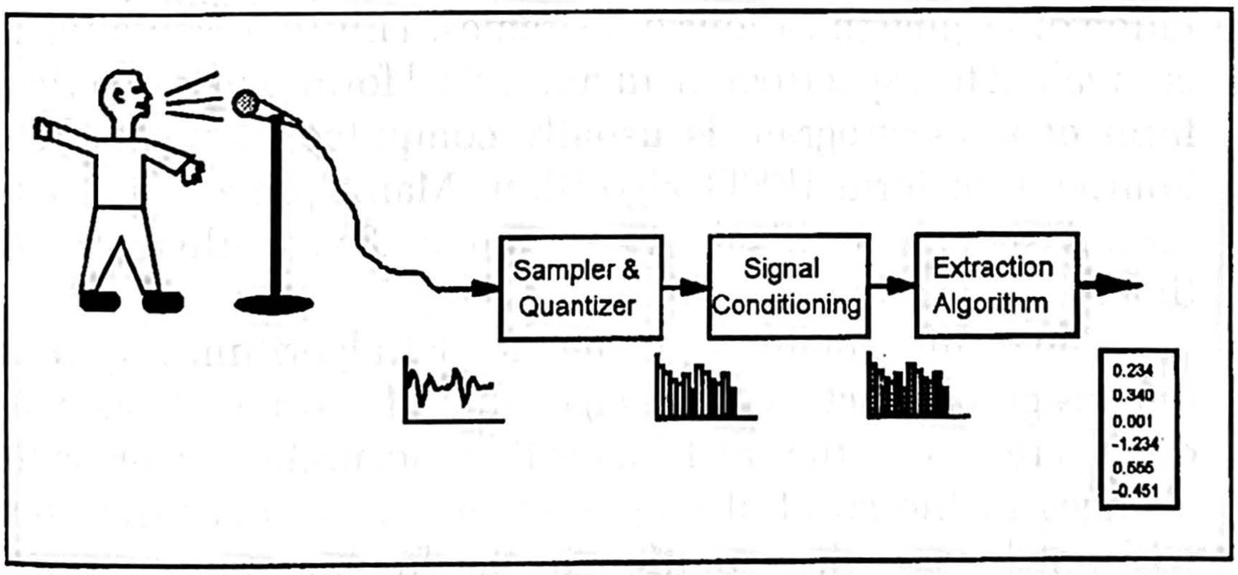

Figure 2.2 The parameter extraction process.

loudness, or have been borrowed from systems for speech coding, recognition, or synthesis.

Pitch

The pitch of a speaker's voice during an utterance is roughly describable by a human listener. The human listener can sense the average pitch and detect changes of pitch during an utterance. Although it is not an easy process, pitch determination can be performed by computer algorithms. Many different algorithms have been devised for pitch extraction [5].

At first glance, pitch appears to be a valuable parameter for speaker identification. For example, a distinction between male voices, female voices, and juvenile voices can be made based mainly on pitch. However, pitch is affected by the speaker's mood and can be modified intentionally by an uncooperative speaker or one with criminal intent.

Frequency Representations

A second simple type of parameter is the frequency representation of a signal in various time frames. This representation is equivalent to a spectrogram in numerical form. The numerical form of a spectrogram is usually computed using the fast Fourier transform (FFT) algorithm. Many processors have been designed specifically to execute FFT algorithms in real time.

The results obtainable using the FFT algorithm vary with the design parameters of the algorithm. If short analysis windows are used, the FFT algorithm accurately represents changes in the spectral energy of the signal over time but will not have high resolution in the frequency dimension. Conversely, if long analysis windows are used, the results will be accurate in the frequency dimension but coarse in the time dimension.[1] Most voice-processing systems use FFTs with moderate-sized analysis windows (approximately 20 ms). The magnitudes of the resulting FFT coefficients are commonly called inverse filter spectral coefficients.

As mentioned earlier, formant frequencies, which can be determined from the frequency representation of a speech signal, are related to the resonant cavities of an individual's vocal tract. Thus, researchers believed that this correlation might be useful for voice recognition. The original research in this area required manual intervention for determining formant frequencies, but soon, automated methods became available [6].

LPC Coefficients

Linear predictive coding (LPC) coefficients are commonly used as features for voice-recognition systems. LPC was developed as an efficient method for representing speech signals and became widely used in many areas of speech processing [7].

In LPC, a parametric representation of speech is created by using past values of the signal to predict future values. The nth value of the speech signal can be predicted by the formula below:

$$\hat{s}_n = \sum_{i=1}^{p} s_{n-i}a_i$$

where s_n is the nth speech sample, the a_k are the predictor coefficients, and $\hat{s}_n$ is the prediction of the nth value of the

[1] This is actually a manifestation of a classical trade-off in physics known as the Heisenberg Uncertainty Principle. Most readers will know it as follows: "one cannot determine both the position and the velocity of an elementary particle with complete accuracy; the more highly determined the one, the less highly determined the other." It is a consequence of the wave description of matter and, in the particular case of digital signal processing, of the wave description of sound.

speech signal. Predictor coefficients can be estimated by an iterative algorithm that minimizes the mean square error between the predicted waveform, $\hat{s}$, and the actual waveform, s. The number of coefficients derived using LPC, p, is a parameter of the algorithm and is roughly related to the number of real and complex poles of the vocal tract filter. With more coefficients, the original signal can be reconstructed more accurately but at a higher computational cost. Typically, 12 coefficients are calculated for speech sampled at 10 kHz [8–10].

Although the LPC predictor coefficients can be used directly as features, many transformations of the coefficients are also used. The transformations are designed to create a new set of coefficients that are optimized for various performance criteria.

The most commonly used transformation is that which derives the anagrammatically named *cepstrum* from the spectrum. The LPC-derived *cepstral* coefficients are defined as follows, where c_i is the ith cepstral coefficient:

$$c_1 = a_1$$

$$c_i = a_i + \sum_{k=1}^{i-1} ((1 - (k/i))a_k c_{i-k}),\ 1 < i \leq p$$

Unlike LPC coefficients, cepstral coefficients are independent and the distance between cepstral coefficient vectors can be calculated with a Euclidean-type distance measure [11].

The reflection coefficients are natural byproducts of the computation of the LPC predictor coefficients. They are defined from the following backward recursion:

$$k_i = b_{i,i}$$

$$b_{j,i-1} = \frac{b_{j,i} + b_{i,i}b_{i-j,i}}{1 - k_i^2}$$

$$b_{j,p} = a_j$$
$$1 \leq j \leq i - 1,\ 1 \leq i \leq p$$

where k_i is the value of ith reflection coefficient, $i = (p, p - 1, \ldots, 1)$, a_i is the ith LPC coefficient, $b_{j,i}$ is a variable within the recurrence relation, and p is the number of LPC coefficients.

The log area coefficients are defined by:

$$g_i = \log\left(\frac{1 - k_i}{1 + k_i}\right) \quad 1 \leq i \leq p$$

where g_i is the ith log area coefficient, k_i is the ith reflection coefficient and $k_i \leq 1$ [12].

Another such transformation is the impulse response function, calculated as follows:

$$h_i = \sum_{k=1}^{p} a_k, h_{1-k} \qquad i > 0$$
$$h_i = 1 \qquad i = 0$$
$$h_i = 0 \qquad i < 0$$

where a_k is the kth LPC coefficient and p is equal to the number of LPC coefficients. The impulse response function is the time-domain output function that would result from inputting an impulse function to a finite duration impulse response (FIR) filter that used the LPC coefficients as the filter coefficients.

Other Parameters

The selection of features, for the most part, is not affected by the type of application. Most text-independent voice-recognition systems currently developed have used the same kinds of features as are used in text-dependent systems. However, some features have been developed specifically to improve performance in noisy environments.

For example, Delta-Cepstrum coefficients are calculated by determining the differences between cepstral coefficients in each time frame. Thus, any constant bias caused by the channel would be removed [11]. The relative spectral-based coefficients (RASTA) use a series of transformations to remove linear distortion of a signal (i.e., filtering). With this technique, the slow-moving variations in the frequency domain are detected and removed. Fast-moving variations—caused by the speech itself—are captured in the resulting parameters [13,14]. The intensity deviation spectrum (IDS) parameters constitute another attempt to remove the frequency characteristics of the transmission channel by normalizing by the mean value at each frequency in the spectrum [15].

Other miscellaneous features have also been suggested: perceptual linear predictive (PLP) coefficients attempt to modify LPC coefficients based on the way human perception and physiology effects sounds [16]. Line spectral pair (LSP) frequencies have also been used as parameters. LSP frequencies are derived from the LPC coefficients and have a rough correlation to formant bandwidths and locations [17]. The partial correlation (PARCOR) coefficients, which are another natural byproduct of LPC analysis, have also been used [18]. Finally, smoothed discrete Wigner distributions (SDWD) attempt to eliminate the problem of time versus frequency accuracy when calculating FFTs. By smoothing the FFT calculation in an efficient manner, the resulting SDWD parameters achieve accuracy in both time and frequency dimensions without a high computation cost. The resulting parameters have been used effectively for voice recognition [19].

The list of features used for voice recognition discussed in this section consists of many parameters that are common to other voice-processing applications as well as some parameters that were devised specifically for the voice-recognition task. Most of these parameters were derived by performing some kind of transformation of the LPC coefficients.

Evaluation of Parameters

To build a successful voice-recognition system, one must make informed decisions concerning which parameters to use. The penalties for choosing parameters incorrectly include poor recognition performance and excessive processing time and storage space. The goal of parameter evaluation should be to determine the smallest set of parameters which contain as much useful information as possible.

The theory of analysis of variance provides a method for determining the relative merits of parameters for voice recognition. Features are identified which remain relatively constant for the speech of a single individual but vary over the speech of different individuals. Typical voice-recognition systems use a set of parameters (features) that may be represented by a vector **W**:

$$\mathbf{W} = [w_1, w_2, \ldots, w_p]$$

where w_1, w_2, etc., are individual features such as LPC coefficients or cepstral coefficients. Numerous vectors can be obtained by performing feature extraction on evenly spaced analysis windows throughout utterances spoken by the individuals to be recognized. Thus, at different time positions in an utterance, the same parameters are calculated.

The F-ratio for each feature, k, in **W** can be determined as follows [6]:

$$F_k = \frac{\text{(Variance of Speaker Means)}}{\text{(Average Within Speaker Variance)}} \qquad (2.1)$$

If s vectors have been collected for each of q number of speakers, then:

$$F_k = \frac{\dfrac{s}{q-1}\displaystyle\sum_{i=1}^{q}(S_{i,k} - U_k)^2}{\dfrac{1}{(s-1)}\displaystyle\sum_{i=1}^{q}\sum_{j=1}^{s}(w_{i,j,k} - S_{i,k})^2} \tag{2.2}$$

$$S_{i,k} = \frac{1}{s}\sum_{j=1}^{s} w_{i,j,k} \tag{2.3}$$

$$U_k = \frac{1}{q}\sum_{i=1}^{q} S_{i,k} \tag{2.4}$$

where $w_{i,j,k}$ is the value of the kth feature for the ith speaker during the jth reference frame. $S_{i,k}$ estimates the value of the kth feature for the ith speaker. The average of the kth feature over all frames of all speakers is represented by U_k.

Features with larger F-ratios will be more useful for voice recognition. However, F-ratios are only valid for the set of data from which they were calculated. Features that appear to be useful for one set of speakers may be worthless for another set of speakers. To calculate meaningful F-ratios, a large population with a large number of examples from each speaker must be used.

Other methods for evaluating the usefulness of features exist. For example, the feature effectiveness criterion (FEC) is defined by Shridhar as follows [10]:

$$FEC = \sum \text{Interspeaker distances} - \sum \text{Intraspeaker distances}$$

Parameters with higher FEC values are more desirable since high interspeaker distances are favorable for discrimination and low intraspeaker distances are favorable for speaker variability. Another method for choosing which features to use in a voice-recognition system is simply to use recognition error rates of the system when different features are used as input. By using the same input speech data and pattern matching algorithm (algorithms will be discussed later in this chapter), the performance of different sets of parameters may be evaluated by comparison of recognition scores. Better parameters will yield better recognition scores. A broad range of testing must be used to prevent overtraining, which occurs when the system parameters are varied slightly in an attempt to achieve better performance on a specific set of input data, while the performance of the system actually drops for more general input data.

Distance Measures

Distance measures refer to methods of calculating differences between parameter vectors. Typically, one of the vectors is calculated from data of the unknown speaker while the other vector is calculated from that of a known speaker. However, some pattern-matching techniques require that vectors from the same speaker be compared to each other to determine the expected variance of the speaker in question. Descriptions of how distance measures are used will be presented later in this chapter.

Many different distance measures have been proposed, and deciding which one to use is as difficult as determining which set of parameters to use. Often, a method is chosen simply because it yields favorable results and/or compensates for the ineffectiveness of certain parameters within a feature vector.

Most distance measures are variations of either the Euclidean or Manhattan distance between two vectors.

Euclidean:

$$d(a,b) = \left(\sum_{i=1}^{p} (a_i - b_i)^2 \right)^{1/2}$$

Manhattan:

$$d(a,b) = \sum_{i=1}^{p} |a_i - b_i|$$

where a_i and b_i are ith components of the two vectors to be compared and p is the number of features to compare.

The Euclidean and Manhattan distance measures are not appropriate for comparing two vectors of LPC coefficients since the coefficients are not independent. However, the likelihood ratio distortion, which only applies to LPC coefficients, can be used. It is defined as follows:

$$d_{LR}(a,b) = \frac{\mathbf{b}^T \mathbf{R}_a \mathbf{b}}{\mathbf{a}^T \mathbf{R}_a \mathbf{a}} - 1$$

where **a** and **b** are vectors of LPC predictor coefficients, $\mathbf{R}_a$ is the Toeplitz autocorrelation matrix (a byproduct of the calculation of the predictor coefficients) associated with **a**, and T is transpose [20]. The log likelihood distance can be computed as follows:

$$d_{LLR} = \log(d_{LR})$$

These two distance measures are effective ways of comparing vectors of LPC predictor coefficients.

Since cepstral coefficients are the most commonly used type of voice-recognition parameter, several distance measures for cepstral coefficients have been suggested. Most of these distance measures are simple variations of the weighted cepstral distance:

$$d(a,b) = \left(\sum_{i=1}^{p} [f_i(a_i - b_i)]^2 \right)^{1/2}$$

where, again, p is the number of features and f_i is the weighting function [21]. Several weighting functions have been suggested:

Uniform:

$$f_i = 1$$

Expected difference:

$$f_i = \frac{1}{E[a_i - b_i]}$$

where E is the expected difference between two features determined from a population of speakers.

Inverse variance:

$$f_i = \frac{1}{\mathrm{var}(a_i)}$$

Uniform without first coefficient:

$$f_i = \begin{cases} 0 \text{ if } i = 1 \\ 1 \text{ if } 1 < i \leq p \end{cases}$$

The expected difference and inverse variance weighting functions attempt to maximize the F-ratio of each feature. The uniform without first coefficient function discounts the first coefficient, which has been shown to contain little information for speaker recognition [21].

The number of different distance measures is as great as the number of different extracted parameter types. Some distance measures were designed for a specific type of parameter. Others were chosen to maximize the F-ratios of any given feature. However, most were chosen simply because of their favorable performance with specific pattern matching algorithms.

Pattern Recognition

Pattern recognition in voice-recognition systems consists of developing a database of information about known speakers (training) and determining if an unknown speaker is one of the known speakers (testing). The result of the pattern recognition step is a decision about an unknown speaker's identity. In the previous sections, we discussed feature extraction and distance measures. In this section, algorithms that use these features and distance measures for making voice-recognition decisions will be explained.

Testing Voice-Recognition Systems

To compare the relative performance of the different pattern-recognition techniques, a brief discussion on the testing of voice-recognition systems is necessary. Typically, the relative performance of voice-recognition systems is based on the error rates for either verification or identification tasks. Unfortunately, error rates can be misleading owing to the large number of variables involved in the testing process. Error rates are affected by the amount of training data (the length and number of training utterances), the amount of testing data, the number of known speakers, the quality of data (amount of noise in the speech signals), and other factors. Thus, comparing error rates reported in the literature of two voice-recognition systems may be misleading.

Many studies report the error rates of various systems with the same input conditions as baselines for comparisons. Then the error rates are more meaningful. In this section, we will report error rates only from studies in which baseline error rates are also available or in which the effects of changing a variable in the test conditions, such as test utterance length, are examined. To make comparisons of test results more meaningful, researchers have begun using standardized databases for testing voice-recognition systems. For example, the

TIMIT database[2], originally designed for testing speech recognition systems, can also be used for testing voice-recognition systems. It contains speech from 420 speakers—230 male and 190 female—from throughout the United States with the speakers segregated into eight dialect regions (DR1-DR8). For each speaker, there are recordings of 10 sentences, two of which (SA1 and SA2) are the same for all 420 speakers. The recordings are of excellent quality, with 16 bits of resolution, sampled at 16 kHz.

Several other databases specifically designed for voice recognition are gaining acceptance as standard test databases. Among these are the SPIDRE, YOHO, and KING databases [22]. Unfortunately, the currently available databases do not meet the needs of all researchers, so many nonstandard test databases are still used.

Pattern-Recognition Techniques

Pattern-recognition processes consist of training and testing. During training, a model of each known speaker must be created. Each model consists of a set of features extracted from utterances spoken by the individual. The exact form of the model will depend on the nature of the pattern-recognition algorithm used. During testing, a similar model is created for the unknown speaker. To make a decision, the pattern-recognition algorithm compares the model of the unknown speaker with models of known speakers. The basic structure of the voice-recognition pattern-recognition process is shown in Figure 2.3.

Many different types of speaker models and decision methods will be discussed in the sections that follow, such as long-term feature averages, vector quantization, hidden Markov models, neural networks, and segregating techniques.

[2] Available from the Linguistic Data Consortium (LDC) at the University of Pennsylvania, 441 Williams Hall, Philadelphia, PA 19104-6305. Tel. (215) 898-0464. E-mail: ldc@unagi.cis.upenn.edu

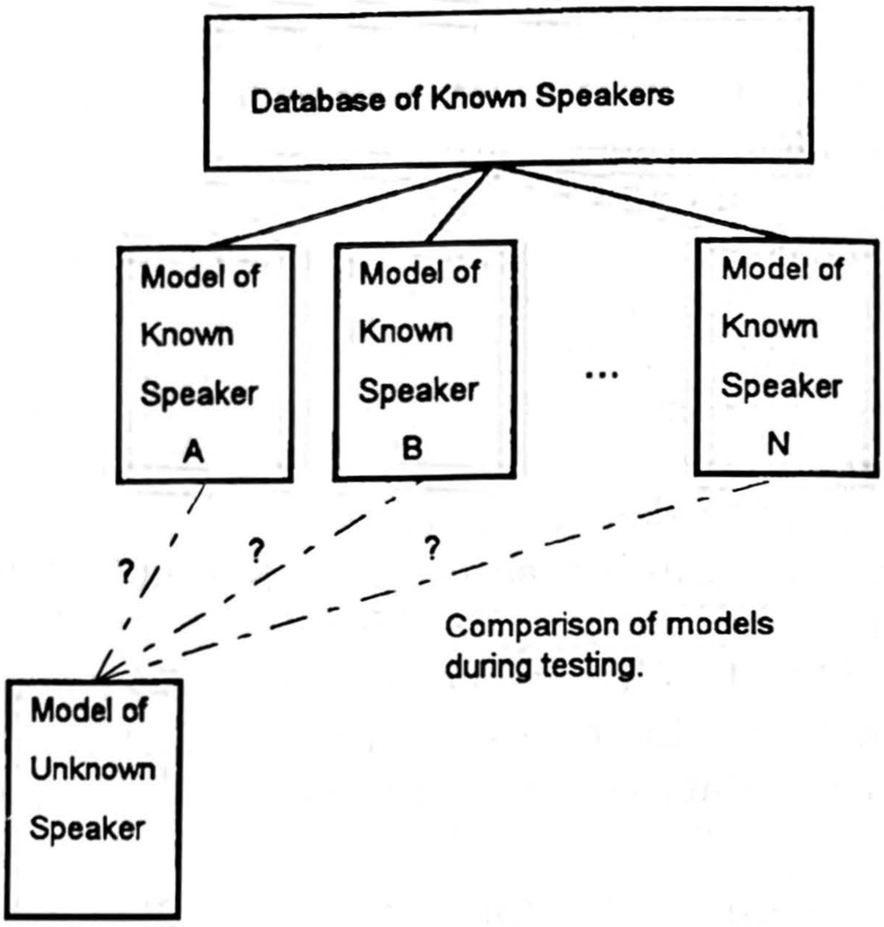

Figure 2.3 Voice-recognition pattern-recognition structure.

Long-term Averaging. One of the first modeling techniques proposed was long-term averaging of features. In this technique, a large number of feature vectors is obtained for each known speaker. The average and variance of each component of the feature vector are computed for all the examples from an individual. Thus, the model for each known speaker consists of two vectors: a vector of the average values of the example vectors and a vector of the variances. A similar model is made for an unknown speaker. The variance vectors are used for weighting each component of the average vectors in a manner related to the F-ratio of the features [23].

For closed-set identification, the decision is made by finding the model of a known speaker whose average vector is closest to the average vector of the unknown speaker. The distance between the two average vectors is computed using a weighted Euclidean distance measure. For verification, a decision criterion must be established. If the distance between the unknown and the known speaker is greater than the threshold value, the unknown is rejected. The criterion value must be adjusted so that the false acceptance error rate is as low as possible without increasing the false rejection error rate above an intolerable level.

The long-term averaging technique was designed for text-independent voice recognition. Its accuracy is highly dependent on the duration of the training and test utterances, which must be sufficiently long and varied. With shorter utterances, the intraspeaker variance increases owing to differences in the content of the utterances. For example, a short training utterance might contain a sentence in which mainly low vowels were used, such as "the cat sat on the mat." If the test utterance contained mainly high vowels instead, as in "he eats peas, beets, and kiwis," then the long-term average vectors for the two utterances would be different despite the fact that the vectors were generated from the same person's speech. Using the long-term averaging technique [24], an error rate of 80% was reported for 0.06 seconds of test data, 34% for 2.5 seconds of test data, and 6% for 40 seconds of test data. These experiments were all performed with 20 seconds of training data.

Researchers have used the long-term averaging approach with several different kinds of features, such as inverse filter spectral coefficients, pitch, and cepstral coefficients [25]. The inverse filter spectral coefficients were the most commonly used despite their susceptibility to channel variations [10,25].

The voice-recognition technique of using long-term averages of features yields favorable results when long utterance lengths are used and channel variance is small. Of course, its performance would be better if the training and testing text were the same, since the intraspeaker variance due to the content of the utterances would be reduced. Unfortunately, it is not always possible to control the training and testing utterances.

Vector Quantization. In the long-term averaging approach, each speaker's model consisted of a single cluster of data represented by an average and variance vector. However, this approach yields a high variance if the data actually contains multiple clusters. Since human speech is composed primarily of vowels, it is natural to expect clusters in a set of feature vectors. Each cluster is the result of the speaker generating the same sound each time a given vowel is pronounced.

Vector quantization (VQ) is an effective method of segregating data into clusters and determining the centroids of those clusters. VQ reduces a set of n k-dimensional vectors into a codebook of N centroid vectors where $n \gg N$. Linde, Buzo, and Gray developed an efficient algorithm for determining the codebook vectors [26]. The basic algorithm is specified as follows:

1. Initialize the N codebook vectors uniformly throughout the vector space by analyzing the sample data.
2. Partition the n training vectors into N groups by determining to which centroid each training vector is closest. Any distance measure may be used at this step.
3. Calculate the average distortion for each input vector using the same distance measure that was used in step (2). If the change since the last iteration of the average distortion is less then some threshold, e, then terminate.
4. Determine the new centroids of each of the partitions of input vectors and store these new values in the codebook. Go to (2).

VQ was originally designed for speech transmission systems to reduce the bandwidth of signals. Instead of transmitting all the bits necessary to represent the k-dimensional vector, only the codebook entry number of the centroid closest to the vector would need to be transmitted. Thus, a sequence of codebook entry numbers could be transmitted to represent an entire utterance. At the receiving end, an approximation of the original vector could be constructed by looking up the codebook entry of each number in the sequence. The codebook itself would only need to be transmitted once at the beginning of the message.

VQ can be used in many different ways in automatic speaker recognition. In some systems, VQ is used only to compress the data. In other systems, the segregation of vectors is used as a preprocessing step. However, the most common use of VQ is as a pattern-recognition method itself.

When VQ is used as a pattern-recognition method, a codebook is created for each known speaker by applying the VQ algorithm shown above to a set of feature vectors from the known speaker's training utterances. For text-independent identification, a comparison of the unknown's vectors is made with the codebooks of each known speaker. For each codebook, the minimum distortion of each vector in the test data to one of the vectors in the codebook is accumulated. The name of the codebook with the smallest accumulated distortion is returned as the identity of the unknown speaker. For text-independent verification, a similar procedure is followed. The accumulated distortion between the unknown and the codebook for the person to be verified is determined. If the distortion is above the criterion value, the unknown is rejected [20].

A similar approach may be used for text-dependent identification and verification. First, a codebook is created for each speaker using utterances of a prescribed text. Then, a sequence of codebook entries is determined for the same utterance—that is, encoded using the codebook. This sequence is called a *template*. During testing, the same prescribed text is spoken by the unknown person. For identification, the unknown is compared with all the stored sequences of codebook entry numbers. The accumulated distortion between the unknown and each stored sequences is calculated. The stored sequence with the lowest distortion is determined to be the same as the unknown [27]. For verification, a similar process is used with a criterion value for the distortion. During both identification and verification, a time-alignment process is used to remove variations in speaking rate of the prescribed text that might otherwise cause unwarranted distortion. Thus, using VQ for text-dependent voice recognition is very similar to the speech recognition of isolated words using dynamic time warping.

Using the VQ method described above introduces a new variable that affects performance—codebook size. With a codebook of only one vector, this technique is similar to long-term averaging of feature vectors. With larger codebooks, a speaker's voice can be better characterized, but at significant computational expense. F. K. Soong reports error rates of 20% for codebooks of size 4, 10% for size 8, and 2% for size 64 for identification based on utterances of 10 digits (thus only quasi-text-independent) in a noise-free environment [20].

VQ is a useful technique for automatic speaker recognition because of its ability to reduce the size of a data set dramatically with very little loss in accuracy.

Hidden Markov Models. Hidden Markov models (HMMs) are a useful method for modeling both the stationary and transient properties of a signal. They are appropriate for modeling speech because some speech sounds are sustained, such as vowels, while others are ephemeral, such as stop consonants, and the transitions between them are short periods of rapid change. Since HMMs are probabilistic by nature, they are able to represent accurately signals that exhibit such diverse behavior.

The basic structure of a HMM is a set of states with transitions between each state. For each transition from a given state, a probability of taking that transition is assigned. The sum of the probabilities of all transitions from a state must equal one. At each state, a symbol is outputted. The symbol to be outputted is also determined probabilistically. Thus, each state contains a probability distribution of the possible output symbols. These models are called "hidden" because the sequence of states is not directly observable. It can only be probabilistically deduced from the sequence of output symbols, which is all that can be observed. HMMs display the *Markov* property since the probability of taking any transition is not based on any previous behavior but only on the current state of the system.

Formally, an HMM may be defined as follows [28]:

N = The number of states in the model;

M = The number of output symbols in the model;

$Q = \{q_1, q_2, \ldots, q_N\}$, the states in the model;

$A = \{a_{ij}\}$, $a_{ij} = \Pr(q_j \text{ at } t + 1 | \, q_i \text{ at } t)$,
the state transition probability distribution;

$B = \{b_j(k)\}$, $b_j(k) = \Pr(v_k \text{ at } t \,|q_j \text{ at } t)$, the output
symbol probability distribution at state j,
where $\{v_k\}$ is the set of output symbols;

$\pi = \{\pi_i\}$, $\pi_i = \Pr(q_i \text{ at } t = 0)$, the initial state distribution.

The formal model for the HMM shown in Figure 2.4 would be:

$N = 2$

$M = 2$

$Q = \{q_1 = \text{“State 1”}, q_2 = \text{“State 2”}\}$

$A = \{a_{11} = 0.5, a_{12} = 0.5, a_{21} = 0.7, a_{22} = 0.3\}$

$B = \{b_1(\mathrm{O}_1)\text{=}0.1, b_1(\mathrm{O}_2)\text{=}0.9, b_2(\mathrm{O}_1)\text{=}0.3, b_2(\mathrm{O}_2)0.7\}$

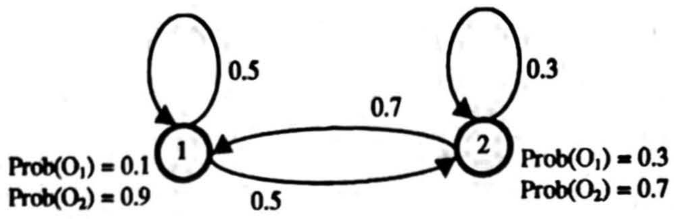

Figure 2.4 Two-state ergotic HMM.

The π distribution was not shown in Figure 2.4; however, if we assume that starting in state 1 or state 2 is equally likely, then:

$$\pi = \{\pi_1 = 0.5,\ \pi_2 = 0.5\}$$

Using this model, an output sequence, $O = \{o_0, o_1, \ldots o_{t-1}\}$, can be created by first choosing an initial state according to the initial state distribution π. Then, for each time step, a symbol is outputted according to the distribution B and a transition is taken according to distribution A for the current state. A random number generator is necessary for determining the outcome of each choice. If a generator that returns evenly distributed numbers in the range $0.0 \leq x < 1.0$ is used, then a value returned from the random number generator <0.5 would indicate starting in state 1, while a value ≥ 0.5 would indicate starting in state 2. Transition and output choice may be determined in a similar fashion.

Many topologies of HMMs are possible. HMMs that contain transitions to and from every state with nonzero probability are called *ergotic* models, since they often exhibit so-called ergotic behavior; that is, the probability that each state will be revisited approaches 1 as time increases, and revisits do not necessarily occur at periodically spaced intervals [28].

Other topologies may be formed by forcing certain transitions of an ergotic model to have probabilities of zero. For example, the HMM in Figure 2.5 would have:

$$a_{13} = 0,\ a_{21} = 0,\ a_{31} = 0,\ a_{32} = 0$$

Notice that state 3 of the left-to-right HMM in Figure 2.5 is an *absorbing state* that cannot be exited once entered. The use of left-to-right, circular (Figure 2.6), and other topologies have been described in the literature [29,30].

There are three basic problems related to HMMs:

1. *The recognition problem:* Given an output sequence and a model, what is the probability that the model could have created the sequence?
2. *The sequence problem:* Given an output sequence and a model, what is the most likely sequence of states that could have created the output sequence?
3. *The training problem:* Given an output sequence and a topology, how can the parameters of a model—that is, the probability distributions for transitions and outputs—be adjusted to maximize the probability that the model created the output sequence?

An algorithm that solves the *recognition problem* can be used for recognition tasks by comparing new data to the models of known signals. A solution to the *sequence problem* can

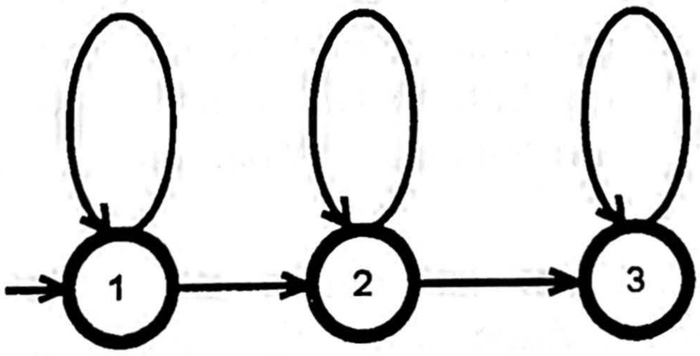

Figure 2.5 Left-to-right HMM.

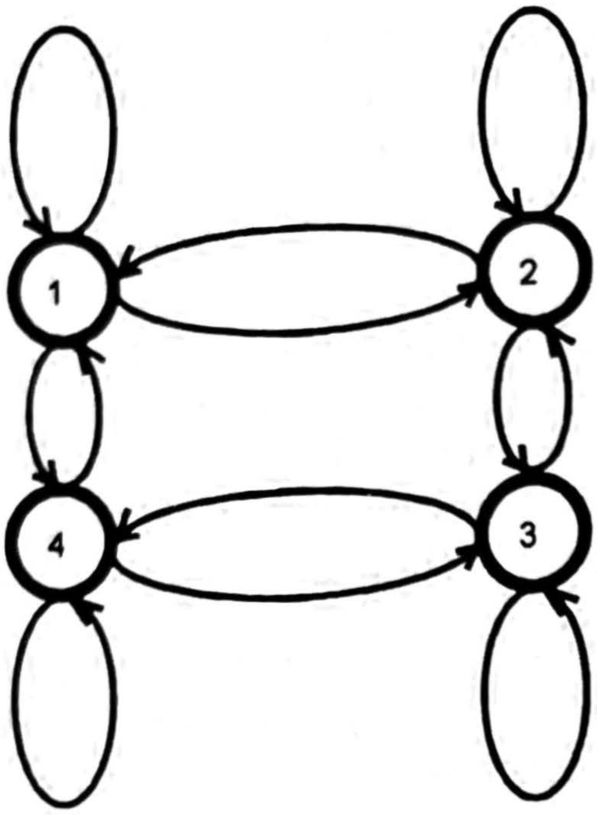

Figure 2.6 Circular HMM.

be used for applications in which each state has a specific meaning. For example, each state might be related to a broad phonetic category, such as vowels or plosives. Finally, the solution to the *training problem* will allow us to train a HMM for a given input sequence. Efficient recursive and iterative algorithms have been developed for solving these three problems [28].

The models described thus far have assumed that discrete output symbols were used. These are called *discrete HMMs.* However, the output sequence need not consist of a set of discrete symbols. *Continuous HMMs* use a probability distribution for each component of the parameter vectors at each state.

Text-Dependent Voice Recognition. When HMMs are used for text-dependent voice recognition, the process is similar to the way in which HMMs are used for small-vocabulary, isolated-word recognition. One HMM is trained for each individual uttering the prescribed text, using the solution to the training problem. When an unknown sample of speech is to be recognized, the solution to the recognition problem is used for determining which model of a known speaker has the highest probability of generating the unknown sample.

HMMs used for text-dependent recognition may be either discrete or continuous. If the models are discrete, then a vector quantizer is used to convert each feature vector into the index number of the vector in the codebook that most closely matches the feature vector. Thus, the number of output symbols for the model, M, is equal to the size of the codebook. If the model is continuous, then the distributions of the actual feature vector values are used for B and the output at each state will be a feature vector.

Text-dependent voice-recognition systems (and small-vocabulary, isolated-word recognition systems) typically use left-to-right HMMs. The ordering of states models the order of speech events during the prescribed text. Since the prescribed text is spoken during both training and testing, the ordering of speech events should be the same. Left-to-right HMMs are also able to model the duration of speech events by using the transitions from each state back to itself. (See Figure 2.5.)

Many experimental studies have tested the performance of text-dependent HMM-based voice-recognition systems. In 1989, an error rate of 4.6% was reported for an HMM-based system using data recorded over long-distance telephone lines [31]. As a baseline, the paper reported an error rate of 6.2% for a template-based system using the same input. Other experiments resulted in an error rate of 3.5% for an HMM-based system using 1.1-second test utterances with 66 seconds of training data versus 6.7% for a template-based approach using the same input data [11].

Text-Independent Voice Recognition. HMMs have also been used for text-independent voice recognition. In these systems, either ergotic or circular models are used. The states are trained to represent different phonetic classes, such as strong vocalization, silence, or nasal [11]. Since more transitions are possible in these models than in left-to-right models, no ordering of speech events is imposed. Therefore, during training and testing, the individuals may speak any text.

During training, the parameters of a model are adjusted to represent best the salient features of each person's speech. If enough training data is used, the model parameters will stabilize. An HMM is trained for each known speaker. During testing, the solution to the recognition problem is used to determine which model is most likely to generate the unknown input.

Experimental results show that text-independent voice-recognition systems using HMMs can be expected to perform slightly better than systems using VQ. One experimenter reported identification error rates of 4.4% for a system based on continuous HMMs versus 4.6% for a VQ system [32]. In the same experiment, a system based on discrete HMMs produced an identification error rate of 11.7%. These experiments were performed with 40 seconds of training data and 20 seconds of test data with 36 speakers. Another experimenter reported similar results for verification tasks [33]. In all, experimental results show that HMM-based systems perform slightly better than text-independent systems based on VQ.

In the applications of HMMs described so far, one HMM was used to represent the speech of each individual. These types of models may be called *utterance unit models*, since each model represents the entire utterance given by the individual. Rosenberg proposed using *subword unit models* for voice recognition. The subword could either be phone-like units (PLUs), in which a phonetic transcription of the training data is required, or acoustic segment units (ASUs), in which segmentation of speech segments is performed automatically [34]. Recognition of a speaker would be performed by recognizing a series of subword units. This method would be similar to large-vocabulary, continuous-speech recognition, in which models of words are the concatenation of subword units. A voice-recognition system based on subword unit models could be either text-dependent or text-independent.

Preprocessing. HMMs have also been used as a preprocessor for voice-recognition systems that segregate feature vectors by phonetic category. The details of segregating systems will be discussed later.

Summary of HMMs. HMMs have been successfully used in voice recognition. Utterance unit models have been used for both text-dependent and text-independent recognition systems. Other research has been carried out to examine the use of subword unit models for recognition. HMMs are also used in the preprocessing stages of voice-recognition systems that segregate feature vectors by phonetic category.

Neural Networks. Artificial neural networks (NNs) are computational models that attempt to emulate the human brain by a topology that resembles interconnected nerve cells. NNs are capable of modeling nonlinearity and can be used for many different tasks, such as classification, associative memory, and clustering. This versatility has allowed them to solve problems in areas as diverse as computer vision, process control, and medical diagnosis [35]. The main drawback of neural networks is their long training time. Although knowledge about neural networks is still in an early stage, their application to automatic speaker recognition is significant.

An NN consists of a collection of neurons (also called perceptrons or nodes) that are connected by weighted pathways. Each neuron is a processing element that has many inputs, performs one function, and produces one output. The computation performed by a typical neuron consists of taking the sum of its inputs and using that value as the argument to a nonlinear function. (See Figure 2.7.)

This nonlinear function is called the *activation function* of the neuron. The most commonly used activation function is the *sigmoid*:

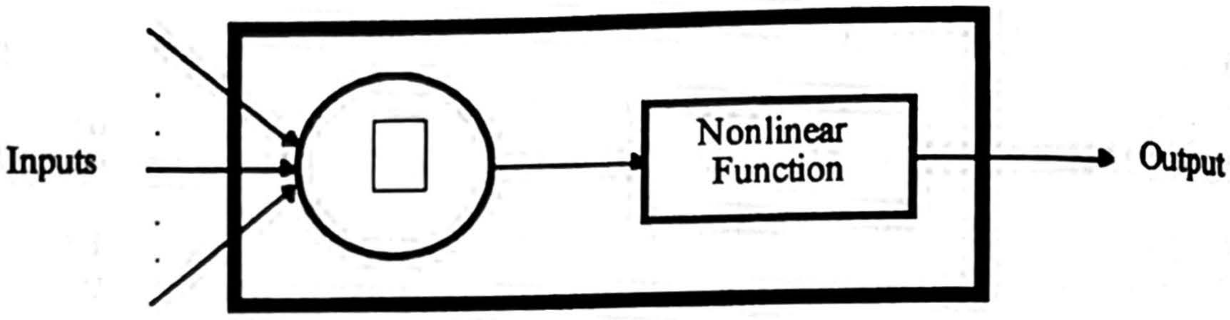

Figure 2.7 A typical neuron.

$$f(x) = \frac{1}{1 + e^{-\lambda x}} \tag{2.5}$$

where λ is > 0. The constant λ determines how "hard" the activation function limits the input. The "hardness" of the activation may be viewed graphically as the slope of the transition in Figure 2.8. Steeper slopes indicate a harder activation function, which results in faster changes in the output value in response to changes in the input value.

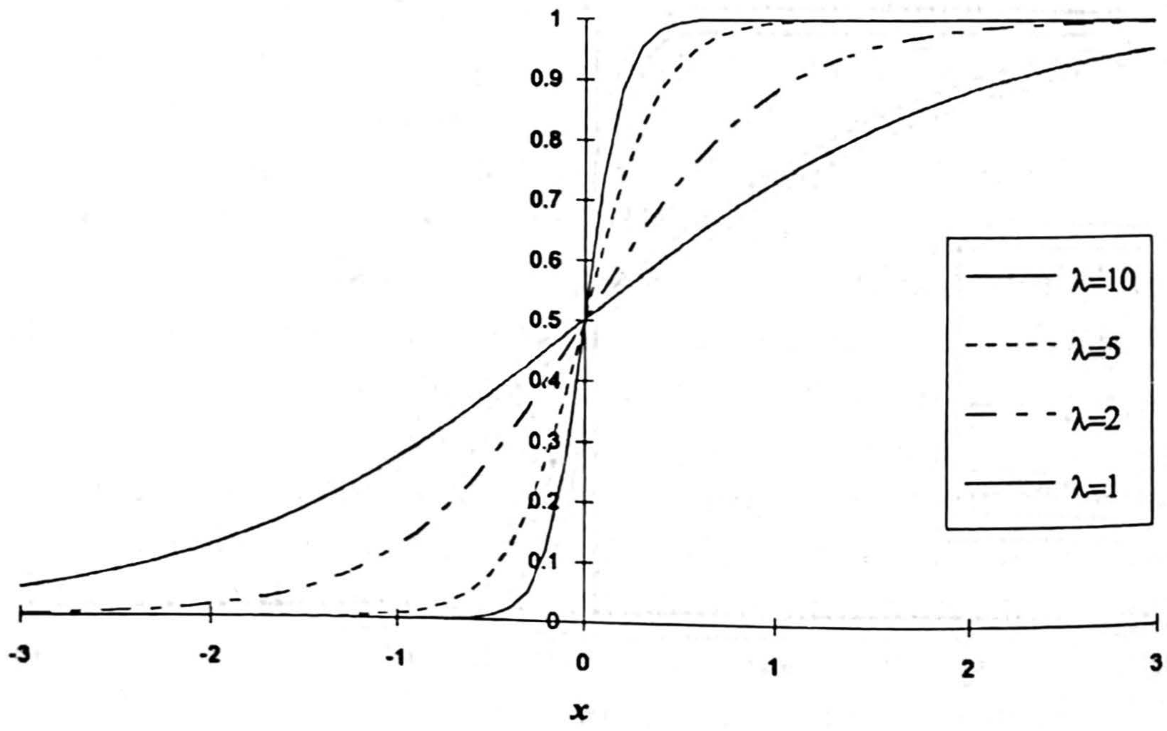

Figure 2.8 The sigmoidal activation function.

The most commonly used neural networks, called *multilayer feedforward networks*, consist of multiple layers of neurons connected by weighted pathways. The networks as a whole have multiple inputs and one or more outputs. The pathways connect neurons from one layer of the network to the next layer closer to the output layer.

The network shown in Figure 2.9 has three input neurons $\{z_1, z_2, z_3\}$ and two output neurons, $\{o_1, o_2\}$. This network is fully connected—that is, the output of each neuron is used as an input for every neuron in the next layer of the network. This network contains one hidden layer of three neurons $\{y_1, y_2, y_3\}$. Since each neuron has only one output value, this value may be given the same name as the neuron. Thus, the output value of neuron z_1 is also called z_1. The weights are shown as $v_{i,j}$ for the first layer and $w_{j,k}$ for the second layer.

The scheme suggested in Figure 2.7 may be used for determining how to calculate the output values for each neuron. The input neurons do not perform any calculation but simply pass through the input values to the next layer. The output of y_1 can be calculated as follows:

$$y_1 = f\left(\sum_{j=1}^{3} v_{1,j} z_j\right)$$

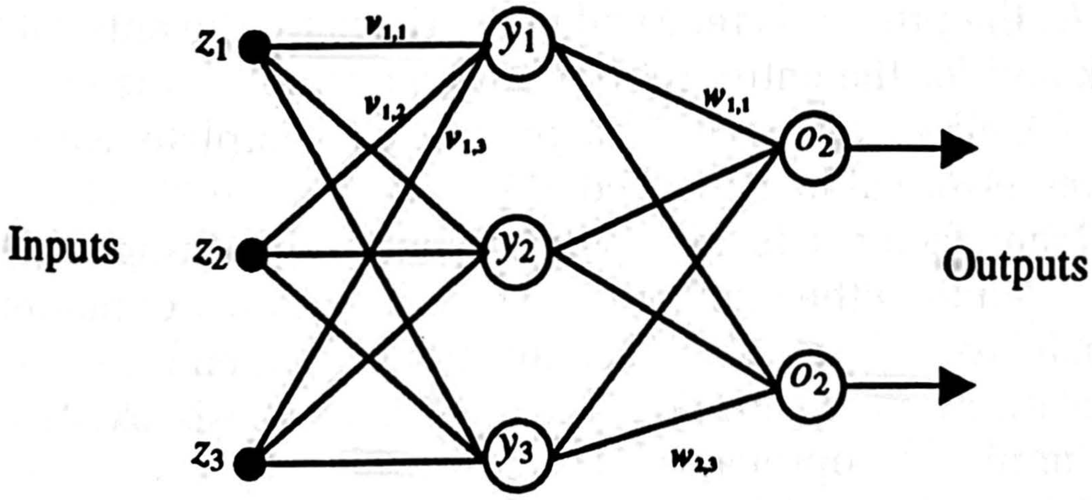

Figure 2.9 A multilayer feedforward neural network.

where $f(\,)$ is the activation function for the neuron, $v_{1,j}$ is the weight of the connection from neuron j, and z_j is the output of neuron j. The outputs of the other neurons can be calculated in a similar fashion. The process of calculating the output values of the network by evaluating the outputs at each layer is called *feedforward recall.*

Training NNs. Like HMMs, NNs must be trained before they are useful. The common way to train a multilayer network is to use error back-propagation training. Before a network is trained, its weights are initialized to small random values. A set of input values and expected output values for those inputs is presented to the network. An iterative gradient descent approach is used for adjusting weights to minimize the differences between the desired output values and the output values calculated by the network using current weights and input.

For each training example, the error at the output layer is fed back to the previous layer and then fed back, layer by layer, until the input layer is reached. Each time an error value is fed back, the weights at that connection are adjusted slightly to reduce the error. A constant, α, determines how large a change in weights can be made during each iteration. This constant is called the *learning rate.*

As the process is repeated many times for the entire network and for the entire set of training examples, the output errors tend to decrease. The process is complete when a desired error value is reached [35].

Many factors affect the performance and training time of NNs, including the number of layers, the number of neurons in each layer, the number of connections between layers, and the learning rate. Currently, no theoretical basis exists for determining the optimal values for these variables. Thus, they are determined by trial-and-error or by heuristics.

Another concern about NNs is the possibility of overtraining. Overtraining occurs when too many training iterations are performed on a network, causing the network to learn coincidental patterns in the input. Overtraining causes a network to generalize poorly.

Voice Recognition With NNs. One of the first experimental voice-recognition systems using neural networks was proposed by Oglesby [36]. In this system, a feedforward network was created for each known speaker. Each network contained one output that was trained to be active (output approximately equal to 1) for input belonging to the speaker whom the network was supposed to represent. The output was trained to be inactive (output approximately equal to 0) for input belonging to other speakers. Thus, the training data included positive examples from the speaker in question as well as negative examples from other speakers in the population. The input to this network was a vector of 10 LPC-derived cepstral coefficients.

For speaker identification testing, each input vector was fed forward through the networks of all the known speakers. The output values for each network were accumulated. The network with the highest accumulated score was deemed to be the best match. For speaker verification, the input vectors for the unknown were fed forward through the network belonging to the individual wishing to be verified. If the average output value were greater than a threshold value, the unknown speaker was accepted.

Oglesby performed many tests to determine optimal parameters of NNs. He determined that single-hidden-layer networks outperformed double-hidden-layer networks (error rates of 8% versus 10%). Oglesby also determined that a large number of hidden nodes increased performance. The best error rate was obtained with 16 input nodes and 128 hidden nodes. Oglesby claimed that for small model sizes, his neural network approach outperformed VQ systems [36].

Rudasi and Zahorian presented two other possible strategies for using NNs to perform voice recognition [37]. Their first strategy was to use one large network with one output per known speaker. During training, examples from each speaker were presented to the network. When a particular speaker's input vectors were presented to the network, the expected value for the output corresponding to that speaker was set to one. All other outputs were set to zero.

During identification testing, the average output scores for all of the unknown input vectors were determined. The best match was the speaker with the highest average output value. Verification can be performed by comparing average output values to a criterion value.

The "one large network" strategy has several problems. Although it will perform well for small populations of speakers, performance and training times for large populations would be less than satisfactory. When new speakers are added to the population, the entire network must be retrained. Rudasi and Zahorian's second strategy was to use small *binary networks*, each of which was required to make a distinction between only two speakers, thus taking modularity to an extreme. Although there were now many more networks, the training time for each network was very short. If the population included N speakers, then $N * (N - 1)/2$ binary networks were required. Since each network was responsible for only a small portion of the overall classification, the binary networks could be highly specialized and offer much better performance than a large network.

Binary classifiers may be used in two different ways. The simplest method is to process the unknown data with all the binary networks. The output of each network can be considered to be a vote. The vote need not be a hard decision but can be a score reflecting confidence in the decision. The known speaker with the most votes is determined to be the best match. This approach is called *global soft-decision search*.

A second approach is to perform a *binary tree search*. This approach requires a series of elimination rounds similar to tournaments in sporting events. The winners of each round proceed to the next round of competition until only one winner is left. The binary tree search requires less computation than the global soft-decision search. However, the two methods will produce the same answer if all the binary networks are 100% accurate. Although binary classifiers are valuable for closed-set identification problems with large populations, they cannot be used for verification tasks.

Time-Delayed NNs. In the NN models described so far, the input was given in terms of short-time input vectors—that is, one frame of features at a time. This approach neglects the transient information that may be useful for voice recognition. Earlier in the chapter, we mentioned how HMMs captured transient information that could not be represented with VQ models. Similarly, time-delayed NNs (TDNNs) were developed to capture transient information with an NN approach.

The input for TDNNs is actually a series of feature vectors called a *frame*. Bennani describes a system in which the input consists of a 25-window-wide analysis frame in which each input window overlaps the previous window by 24 of the 25 points [38].

The TDNN system described by Bennani was not a fully connected network. The input neurons were positioned as an array (25 time units × 16 features). Each row of neurons in the hidden layer was connected to only five consecutive rows (in the time-step dimension) of input neurons. Bennani's system had a 21 × 12 neuron hidden layer. Bennani's identification error rate was 2% for 20 speakers using five TIMIT database sentences for training and five for testing.

Summary of NNs. As more knowledge is gained about NNs, their applicability to voice recognition is sure to increase. Several approaches have been tested: large discriminating networks, binary networks, and TDNNs. Because of the promising results, research in this area is expected to continue.

Segregating Systems. Segregating voice-recognition systems treat the text-independent speaker-recognition task as a two-step process. First, all input vectors are segregated into groups based on certain spectral characteristics, generally ones associated with particular classes of speech sounds. Then, for each group of vectors, the vectors from each known individual are compared with vectors from the unknown. The basic idea is to compare a specific sound class produced by the unknown speaker with the same sound class produced by the known speakers. Since segregating systems require two steps, two questions must be answered during their design: "How are vectors segregated?" and "How are vectors within each group compared?" Through the answer to the first question, the answer to the second becomes apparent. Variations on techniques discussed previously, such as VQ and HMMs, are used for solving these problems.

In the first study proposing a segregating approach, vowel samples were segregated manually [39]. Vowel classification was performed by human operators assisted by computer programs that identified likely locations for vowels. LPC coefficients were calculated at the steady state position of each vowel. Then, the vectors were compared using a weighted Euclidean distance measure.

Fakotakis discussed a system in which vowels are located by looking for peaks in the short-time energy contours of utterances [40]. Cepstral coefficients are calculated at each vowel locus and used as input to a vector quantizer that segregates the feature vectors. Each vector is placed in the group corresponding to the nearest centroid in the codebook. Wang describes a similar segregating voice-recognition system that uses VQ for an entire utterance, not just the vowels [41].

If VQ is used for segregation and the distance from the centroid is used for comparison, then the method behaves like the VQ recognition process described earlier. Thus, most VQ segregating systems use distance measures weighted by the variance within each group of vectors [41].

Another method for segregating vectors uses HMMs. Savic describes a method in which ergotic HMMs are used for representing broad phonetic categories such as vowels or fricatives [42]. Each state in the HMM is associated with a category. To segregate the vectors of an utterance, the solution to the HMM sequence problem (for example, "What is the most likely sequence of states for a sequence of observations?") is used. Thus, each vector is assigned to one of the states in the HMM. To compare vectors within each group, a Bayesian classifier is used. Therefore, the mean vector and covariance matrix is required. The final score is a weighted summation of the scores from each group.

Matsui suggests a system in which vectors are segregated using either a voiced/un-voiced classifier or an ergotic HMM for determining broad phonetic categories. Within each group, VQ is used for performing recognition. Thus, each known speaker has several codebooks, one for each category [43].

Kao discusses a similar system [44]. However, in his implementation, a speaker-independent continuous-speech recognizer is used for segregating vectors. The output of the recognizer is a hypothesized phonetic category. VQ codebooks for each speaker are created for each possible category.

Many segregating voice-recognition systems have been developed. These systems use a combination of techniques, such as VQ and HMMs. However, as of this writing, segregating systems that use neural networks have not been discussed in the literature. This would appear to be a promising area of research.

Miscellaneous Pattern-Matching Techniques. In preceding sections, many voice-recognition pattern-matching techniques have been discussed. These techniques include long-term feature averaging, VQ, HMMs, NNs, and segregating systems.

Many other miscellaneous pattern-matching techniques have been discussed in the literature. For example, principal component analysis is a technique for optimizing dynamic time warping systems [45]. Similarly, orthogonalization has been used for improving the performance of long-term averaging recognizers [12]. Several statistical detection approaches have been examined, such as probability density functions [46,47], trajectory space comparisons [48], k-nearest neighbors [15,49], discriminator counting [15,49], Gaussian mixture models [50], Fourier Bessel functions [51], and probabilistic acoustic maps [52,53].

Unfortunately, comparing the performance of voice-recognition systems is difficult owing to variations in testing. Error rates are affected by many features, such as noise, training time, testing time, number of speakers, and definition of free text.

A common feature of all voice-recognition pattern-matching techniques discussed in this section is that they perform a considerable amount of data reduction, essentially some kind of averaging. Data reduction is necessary to extract the salient features of an individual's speech and also to make the recognition process computationally feasible.

VOICE RECOGNITION IN NOISY ENVIRONMENTS

Overcoming the difficulties associated with performing voice recognition in noisy environments is a primary concern. For voice recognition to be successful when performed over telephone lines—perhaps the most important general application area—voice-recognition systems need to be relatively impervious to noise. Several studies in this area have been completed [54–56]. Experiments over radio channels have also been performed [47,57].

The main difficulty with noisy environments is not the noise itself, but the variations in the noise. An extreme example occurs when a voice-recognition system is trained on clean speech—that is, speech containing no noise—and is tested on noisy speech. Error rates are bound to be higher when any aspect of the signal processing is changed between training and testing. These include the type of microphone, the amount of ambient noise, and the transmission medium.

If the same noise appears in both training and testing, however, it will not be a factor unless the signal-to-noise ratio (SNR) is low.

Typical channel variations are changes in the amount of additive noise, bandpass filtering, and phase distortion applied to the actual speech signal. To make voice-recognition systems immune to such variations, several techniques have been devised, including selection of features with immunity to channel variations and preprocessing of signals to separate the noise components of a signal from the actual speech component.

In an earlier section of this chapter, the feature selection process was described, and several types of features used for voice recognition were listed. In addition to the techniques for feature selection already described, some features have been used or invented for use based solely on their immunity to channel variation.

For example, fundamental frequency (pitch) and formant frequencies have been used because they are not affected by additive noise or phase distortion and are affected only slightly by the bandpass filtering typical of communication channels [58,59]. Cepstral coefficients are used because they are unaffected by linear distortion.

Delta-type features have also been used. These are calculated by determining the difference between successive vectors and using the difference vectors as features. Delta-type features automatically remove the bias from a signal.

Other features mentioned in the literature specifically for their immunity to channel variance are clipped autocorrelation coefficients [54] and relative spectral perceptual linear predictive (RASTA-PLP) methods [13,14].

To separate an additive noise component of a signal from the components useful for voice recognition, the communication channel must first be characterized. The simplest method for characterizing a channel in text-dependent applications is to determine the average value of all features over the entire utterance. Then, the average vectors can be subtracted from each vector in the utterance to normalize it [60]. This technique is also useful for compensating for some of the bandpass filtering. However, if the utterances are short, some of the speaker-dependent information will also be removed from the signal.

Another method is to characterize the additive noise from segments of the utterance where no speech is present, ensuring that actual speech will not be mistakenly classified as stationary channel noise. Alternately, Wang suggests averaging channel characteristics over many different types of channels instead of over time [41].

Filtering signals during preprocessing can be useful for removing variance in the bandpass filtering behavior of the communication channel. Successful results have been achieved by attenuating certain spectral regions during preprocessing that are likely to be attenuated by some, but not all, communication channels, thus achieving normalization [44,61]. The same effect results from attenuating cepstral coefficients, a process called *liftering* [13].

Gish suggests that the channel should be modeled statistically as a Gaussian random vector, which can then be incorporated into the classifier, assuming that a Gaussian probability distribution function classifier (GPDF) is used [55]. By choosing features that are immune to channel effects or by removing channel effects before features are extracted, voice-recognition systems can be designed to perform better in noisy environments.

SUMMARY

In this chapter, previous research in voice recognition was presented. The early use of voiceprints was discussed. The different types of features, distance measures, and pattern-matching techniques were explained. Finally, studies on performing voice recognition in noisy environments were summarized.

References

[1] Kersta, L. G., "Voiceprint Identification," *Nature*, Vol. 196 No. 4861, 29 December 1962, pp. 1253–1257.

[2] Bolt, Richard H. et al., "Identification of a Speaker by Speech Spectrograms," *Science*, Vol. 166, Oct. 1969.

[3] Koenig, Bruce E., "Spectrographic Voice Identification: A Forensic Survey," JASA, Vol. 79 No. 6, June 1986, pp. 2088–2090.

[4] Tetschner, W. *Voice Processing*, 2nd ed., Norwood, MA: Artech House, 1993.

[5] Rabiner, Lawrence R., and Schafer, Ronald W., *Digital Processing of Speech Signal*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1978.

[6] Pruzansky, S., and Mathews, M. V., "Talker-Recognition Procedure Based on Analysis of Variance," *JASA* 36, 1964, pp. 2041–2047.

[7] Atal, B. S., "Speech Analysis and Synthesis by Linear Prediction of Speech Wave," *JASA* 47, 65(A), 1970.

[8] Atal, B. S., "Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification," *JASA*, Vol. 55, June 1974, pp. 1304–1312.

[9] Fasolo, L. and Mian, G. A., "A Comparison Between Two Approaches to Automatic Speaker Recognition," *ICASSP*, 1978, pp. 275–278.

[10] Shridhar, M. et al., "Text-Independent Speaker Recognition Using Orthogonal Linear Prediction," *ICASSP*, 1981, pp. 197–200.

[11] Rosenberg, Aaron E. et al., "Connected Word Talker Verification Using Whole Word Hidden Markov Models," *ICASSP*, 1991, pp. 381–384.

[12] Mohankrishnan, N. et al. "A Composite Scheme for Text-Independent Speaker Recognition," *ICASSP*, 1982, pp. 1653–1656.

[13] Kao, Yu-Hung et al., "Robustness Study of Free-Text Speaker Identification and Verification," *ICASSP*, 1993, pp. 379–382.

[14] Openshaw, J.P. et al., "A Comparison of Composite Features Under Degraded Speech in Speaker Recognition," *ICASSP*, 1993, pp. 371–374.

[15] Nakasone, H. and Melvin, C., "Computer Assisted Voice Identification System," *ICASSP*, 1988, pp. 587–590.

[16] Xu, L. et al., "The Optimization of Perceptually Based Features for Speaker Identification," *ICASSP*, 1989, pp. 520–523.

[17] Liu, Chi-Shi et al., "Study of Line Spectrum Pair Frequencies for Speaker Recognition," *ICASSP*, 1990, pp. 277–280.

[18] Attili, Joseph B. et al., "A TMS32020-Based Real-Time, Text-Independent Automatic Speaker Verification System," *ICASSP*, 1988, pp. 599–602.

[19] Wilber, JoEllen and Taylor, Fred J., "Consistent Speaker Identification via Wigner Smoothing Techniques," *ICASSP*, 1988, pp. 591–593.

[20] Soong, F. K. et al., "A Vector Quantization Approach to Speaker Recognition," *ICASSP*, 1985, pp. 387–390.

[21] Velius, George, "Variant of Cepstrum-Based Speaker Identity Verification," *ICASSP*, 1988, pp. 583–586.

[22] Godfrey, J. et al., "Public Databases for Speaker Recognition and Verification," *ESCA Workshop on Automatic Speaker Recognition, Identification, and Verification*, 1994.

[23] Markel, John D. and Davis, Steven B., "Text-Independent Speaker Identification from a Large Linguistically Unconstrained Time-Spaced Data Base," *ICASSP*, 1978, pp. 287–289.

[24] Wrench, E. H., "A Realtime Implementation of a Text-Independent Speaker Recognition System," *ICASSP*, 1981, pp. 193–196.

[25] Doddington, George R., "Speaker Recognition—Identifying People by Their Voices," *Proceedings of the IEEE*, Vol. 73 No. 11, Nov. 1985, pp. 1651–1663.

[26] Linde, Y., Buzo, A., and Gray, R., "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, Vol. COM-28 No. 1, Jan. 1980, pp. 84–95.

[27] Buck, Joseph et al., "Text-Dependent Speaker Recognition Using Vector Quantization," *ICASSP*, 1985, pp. 391–394.

[28] Rabiner, Lawrence R. and Juang, B.H., "An Introduction to Hidden Markov Models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Jan. 1986, pp. 4–16.

[29] Zheng, Yuan-Cheng and Yuan, Bao-Zong, "Text-Dependent Speaker Identification Using Circular Hidden Markov Models," *ICASSP*, 1988, pp. 580–582.

[30] Lee, K.-F., *Automatic Speech Recognition*, Boston, MA: Kluwer Academic Publishers. 1989.

[31] Naik, Jayant M. et al., "Speaker Verification over Long Distance Telephone Lines," *ICASSP*, 1989, pp. 524–527.

[32] Matsui, Tomoko and Furui, Sadaoki, "Comparison of Text-Independent Speaker Recognition Methods Using VQ Distortion and Discrete/Continuous HMMs," *ICASSP*, 1992, pp. 157–160.

[33] Tishby, N., "On the Application of Mixture AR Hidden Markov Models to Text Independent Speaker Recognition," *IEEE Trans. Signal Processing*, SP-39, 1991, 563–570.

[18] Attili, Joseph B. et al., "A TMS32020-Based Real-Time, Text-Independent Automatic Speaker Verification System," *ICASSP*, 1988, pp. 599–602.

[19] Wilber, JoEllen and Taylor, Fred J., "Consistent Speaker Identification via Wigner Smoothing Techniques," *ICASSP*, 1988, pp. 591–593.

[20] Soong, F. K. et al., "A Vector Quantization Approach to Speaker Recognition," *ICASSP*, 1985, pp. 387–390.

[21] Velius, George, "Variant of Cepstrum-Based Speaker Identity Verification," *ICASSP*, 1988, pp. 583–586.

[22] Godfrey, J. et al., "Public Databases for Speaker Recognition and Verification," *ESCA Workshop on Automatic Speaker Recognition, Identification, and Verification*, 1994.

[23] Markel, John D. and Davis, Steven B., "Text-Independent Speaker Identification from a Large Linguistically Unconstrained Time-Spaced Data Base," *ICASSP*, 1978, pp. 287–289.

[24] Wrench, E. H., "A Realtime Implementation of a Text-Independent Speaker Recognition System," *ICASSP*, 1981, pp. 193–196.

[25] Doddington, George R., "Speaker Recognition—Identifying People by Their Voices," *Proceedings of the IEEE*, Vol. 73 No. 11, Nov. 1985, pp. 1651–1663.

[26] Linde, Y., Buzo, A., and Gray, R., "An Algorithm for Vector Quantizer Design," *IEEE Transactions on Communications*, Vol. COM-28 No. 1, Jan. 1980, pp. 84–95.

[27] Buck, Joseph et al., "Text-Dependent Speaker Recognition Using Vector Quantization," *ICASSP*, 1985, pp. 391–394.

[28] Rabiner, Lawrence R. and Juang, B.H., "An Introduction to Hidden Markov Models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Jan. 1986, pp. 4–16.

[29] Zheng, Yuan-Cheng and Yuan, Bao-Zong, "Text-Dependent Speaker Identification Using Circular Hidden Markov Models," *ICASSP*, 1988, pp. 580–582.

[30] Lee, K.-F., *Automatic Speech Recognition*, Boston, MA: Kluwer Academic Publishers. 1989.

[31] Naik, Jayant M. et al., "Speaker Verification over Long Distance Telephone Lines," *ICASSP*, 1989, pp. 524–527.

[32] Matsui, Tomoko and Furui, Sadaoki, "Comparison of Text-Independent Speaker Recognition Methods Using VQ Distortion and Discrete/Continuous HMMs," *ICASSP*, 1992, pp. 157–160.

[33] Tishby, N., "On the Application of Mixture AR Hidden Markov Models to Text Independent Speaker Recognition," *IEEE Trans. Signal Processing*, SP-39, 1991, 563–570.

[34] Rosenberg, Aaron E. et al., "Sub-Word Unit Talk Verification Using Hidden Markov Models," *ICASSP*, 1990, pp. 269–272.

[35] Zurada, Jacek W., *Introduction to Artificial Neural Systems*, St. Paul, MN: West Publishing Co., 1992.

[36] Oglesby, J. and Mason, J. S., "Optimization of Neural Models for Speaker Identification," *ICASSP*, 1990.

[37] Rudasi, Laszlo and Zahorian, Stephen A., "Text-Independent Talker Identification with Neural Networks," *ICASSP*, 1991, pp. 389–392.

[38] Bennani, Younes and Gallinari, Patrick, "On the Use of TDNN-Extracted Features Information in Talker Identification," *ICASSP*, 1991, pp. 385–388.

[39] Pfeifer, Larry L., "New Techniques for Text-Independent Speaker Identification," *ICASSP*, 1978, pp. 283–286.

[40] Fakotakis, N., Tsopanoglou, A., and Kokkinakis, G., "A Text-Independent Speaker Recognition System Based on Vowel Spotting," *Speech Communication*, Vol 12, 1993, pp. 57–68.

[41] Wang, H. et al., "A Novel Approach to Speaker Identification over Telephone Networks," *ICASSP*, 1993, pp. 407–410.

[42] Savic, Michael and Gupta, Sunil K., "Variable Parameter Speaker Verification System Based on Hidden Markov Modeling," *ICASSP*, 1990, pp. 281–284.

[43] Matsui, Tomoko and Furui, Sadaoki, "A Text-Independent Speaker Recognition Method Robust against Utterance Variations," *ICASSP*, 1991, pp. 377–380.

[44] Kao, Yu-Hung et al., "Free-Text Speaker Identification over Long-Distance Telephone Channels Using Hypothesized Phonetic Segmentation," *ICASSP*, 1992, pp. 177–180.

[45] Naik, Jayant M. and Doddington, George R. "High Performance Speaker Verification Using Principal Spectral Components," *ICASSP*, 1986, pp. 881–884.

[46] Schwartz, R. et al. "The Application of Probability Density Estimation to Text-Independent Speaker Identification," *ICASSP*, 1982, pp. 1649–1652.

[47] Wolf, J. et al. "Further Investigation of Probabilistic Methods for Text-Independent Speaker Identification," *ICASSP*, 1983, pp. 551–554.

[48] Gong, Yifan and Haton, Jean-Paul, "Text-Independent Speaker Recognition by Trajectory Space Comparison," *ICASSP*, 1990, pp. 285–288.

[49] Higgins, A. L. et al., "Voice Identification Using Nearest-Neighbor Distance Measure," *ICASSP*, 1993, Vol. 2, pp. 375–378.

[50] Reynolds, D. A. and Rose, R. C., "An Integrated Speech-Background Model for Robust Speaker Identification," *ICASSP*, 1992, pp. 185–188.

[51] Gaganelis, D. A. and Frangoulis, E. D., "A Novel Approach to Speaker Verification," *ICASSP*, 1991, pp. 373–376.

[52] Tseng, Belle L. et al., "Continuous Probabilistic Acoustic Map for Speaker Recognition," *ICASSP*, 1992, pp. 161–164.

[53] Chang, Harry M., "Augmented Phonetic Map for Voice Verification," *ICASSP*, 1992, pp. 169–172.

[54] Ney, H., "Telephone-Line Speaker Recognition Using Clipped Autocorrelation Analysis," *ICASSP*, 1981, pp. 188–192.

[55] Gish, H. et al., "Investigation of Text-Independent Speaker Identification over Telephone Channels," *ICASSP*, 1985, pp. 379–382.

[56] Gish, H. et al., "Methods and Experiments for Text-Independent Speaker Recognition over Telephone Channels," *ICASSP*, 1986, pp. 865–868.

[57] Krasner, M. et al., "Investigation of Text-Independent Speaker Identification Techniques under Conditions of Variable Data," *ICASSP*, 1984.

[58] Hunt, Melvyn J., "Further Experiments in Text-Independent Speaker Recognition over Communication Channels," *ICASSP*, 1983, pp. 563–566.

[59] Federico, A. et al., "A New Automated Method for Reliable Speaker Identification and Verification over Telephone Channels," *ICASSP*, 1987, pp. 1457–1460.

[60] Rosenberg, A. E. and Soong, F. K., "Recent Research in Automatic Speaker Recognition," *Advances in Speech Signal Processing*, New York: Marcel Dekker, Inc., 1992.

[61] Rose, R. C. et al., "Robust Speaker Identification in Noisy Environments Using Noise Adaptive Speaker Models," *ICASSP*, 1991, pp. 401–404.

ISBN: 0-89006-927-1

Artech House Publishers BOSTON • LONDON