Speaker Recognition

Douglas O'Shaughnessy

INTRODUCTION

VOCAL communication between people and computers includes the synthesis of speech from text, automatic speech recognition (speech-to-text conversion), and the identification of speakers by analysis of their voices. This tutorial article deals with the third task, automatic speaker recognition (ASR). Given a speech input, the objective of ASR is either to output the identity of the person most likely to have spoken (from among a known population) or to verify whether the speaker is who he claims to be. There are two primary applications for ASR: 1) verifying a person's identity prior either to admission to a secure facility or to a transaction over the telephone, and 2) linking a person to a voice in police work [1]. While fingerprints or retinal scans are usually more reliable ways of verifying a person's identity claims, voice identification has the convenience of easy data collection over the telephone. Many financial institutions, as well as companies furnishing limited access to computer data bases, would like to provide automatic customer service by telephone. Since personal number codes (keyed on a telephone pad) can be lost, stolen, or forgotten, ASR (if sufficiently reliable) can provide a viable alternative.

ASR may be viewed as a complement to speech recognition, where the latter attempts to decode the linguistic message (or text) underlying an utterance, rather than the identity of the speaker. In automatic speech recognition, variation due to different speakers in speech signals corresponding to the same spoken text is often viewed as "noise" to be either eliminated by speaker normalization or (more commonly) accommodated through the use of different stored spectral patterns for different speakers. When the task is to identify the person talking rather than what he is saying, the speech signal must be processed to extract measures of speaker variability instead of segmental features.

The acoustic aspects that differentiate voices are difficult to separate from signal traits which reflect the identity of the sounds (i.e., the abstract linguistic units called *phonemes*). There are two sources of variation among speakers: 1) differences in vocal cords and vocal tract shape, and 2) differences in speaking style. The latter includes variations in both target vocal tract positions for phonemes and dynamic aspects of speech, such as speaking rate. There are no acoustic cues specifically or exclusively dealing with speaker identity. Most of the parameters and features used in speech analysis contain information useful for the identification of both speaker and spoken message. The two types of information, however, are coded quite differently. To a certain degree, segmental information occurs sequentially in a series of *phones* (the acoustic realizations of phonemes). A simplistic speech recognizer could make independent phoneme decisions (from a small range of about 40 phonemes), based on spectral patterns, at a typical speaking rate of 12/s. More practical recognizers attempt to account for contextual effects (e.g., coarticulation) and analyze the speech signal over time ranges longer than individual phones.

For ASR, however, only one decision is made, based on parts or all of an input *test* utterance, and there is no simple set of acoustic cues which reliably distinguishes speakers. Speaker recognizers typically utilize long-term statistics averaged over whole utterances or exploit analyses of specific sounds. The latter approach is common in *text-dependent* applications, where utterances of the same text are used for training and testing; the statistical average method is often used in *text-independent* cases, where training and testing involve utterances from different texts. While only one decision is necessary in speaker recognition, the set of choices can vary widely (unlike the fixed set of 40 phonemes for speech recognition). Some applications need only a binary decision (i.e., is the person who he claims to be?), whereas others require selecting which of N stored voices matches a test input. In some police applications, the list of N suspects can be large.

VERIFICATION VS. IDENTIFICATION

There are two related but different areas of voice recognition: *automatic speaker verification* (ASV) [2] and *automatic speaker identification* (ASI) [3]. (For discussions applying to both ASV and ASI, the more general term automatic speaker recognition (ASR) will be used.) Both ASI and ASV use a stored data base of reference patterns, or *templates*, for N known speakers, and similar analysis and decision techniques are employed. ASV is the simpler task, since it only requires comparing the test pattern against one reference pattern and involves a binary decision whether the test speech matches the template of the claimed speaker. Speakers known to the system who claim

0740-7467/86/1000–0004/$01.00 ©1986IEEE

their true identity are called *customers*, while others are *impostors*. ASI, on the other hand, requires choosing which of the N voices known to the system best matches a test voice. Since N comparisons and decisions are necessary, the error rate can be much higher for ASI than for ASV. Furthermore, the test speaker's voice may not be among the N stored patterns, in which case a "no match" decision is required.

While the worst performance for both ASV and ASI is 0 percent correct, simple guessing yields 50 percent for ASV (assuming an input of equal numbers of customers and impostors), but only $100/N\%$ for closed-set ASI (assuming the test speaker is known to the system, and each speaker is equally likely). There are two classes of errors: *false rejections* and *false acceptances*. A false rejection occurs when the system incorrectly: 1) rejects a true speaker in ASV, or 2) claims a "no match" in ASI. A false acceptance occurs when the system incorrectly: 1) accepts an impostor during ASV, or 2) identifies the wrong person during ASI.

Comparing test and training utterances for speaker identity is much simpler when the underlying texts of the utterances are the same. The straightforward application of speech recognition methods (e.g., template matching) to ASR is only possible for cooperative speakers, who train the system and later test it with the same words. This text-dependent case, using the same text for training and testing, permits the simple comparison of word templates and occurs frequently in ASV applications, but rarely for ASI. In forensic work, speakers are often uncooperative, training may be done surreptitiously, and the test and training texts are often not the same. Having different texts for training and testing, the text-independent case can still use template matching, but much different information must be stored in the templates than when test and reference templates are simply repetitions of the same word. In general, text-independent templates contain long-term statistical data. Error rates for text-independent recognition are considerably higher than for a comparable text-dependent case. To achieve good results for text-independent ASR, much more speech data are usually needed for both training and testing than for text-dependent ASR. Training often exceeds 30 s per speaker, and test utterances are usually longer than 5 s. On the other hand, the performance of text-dependent systems is highly correlated with the vocabulary that is chosen.

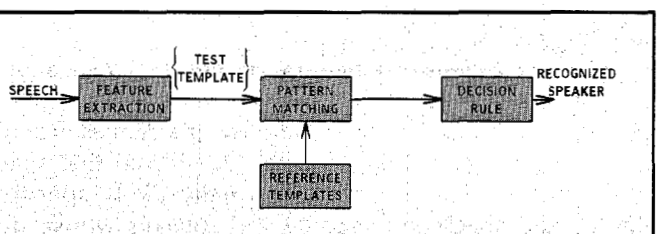

Figure 1. Traditional pattern recognition model for speaker recognition.

RECOGNITION TECHNIQUES

ASR is an example of a pattern recognition task, and uses standard pattern recognition techniques, such as those in the fields of robotics (image identification) or data communications (converting analog signals to digital information at a receiving modem). In essence, ASR requires a mapping between speech and speaker identity so that each possible input waveform is identified with its corresponding speaker. All pattern recognition tasks, including ASR, utilize two phases: *training* and *recognition*. Performed off-line and often combining manual and automatic methods, the training phase establishes a reference memory or dictionary of (speech) template patterns, which are assigned (speaker) labels. The automatic (and usually real-time) recognition phase attempts to assign a label to an unknown input test pattern.

In theory, a speaker recognizer could be as simple as a large dictionary where each entry is a stored waveform associated with a speaker identity. Given an input utterance, this dictionary would be searched to find an exact match (or perhaps a close match, using some similarity measure), and the system would output the corresponding identity. However, even assuming a one-to-one correspondence between utterances and speakers, this approach is grossly impractical due to the immense memory and calculation search time required for even the simplest applications. Much more efficient procedures are necessary in practice.

Applying pattern recognition methods to ASR involves several steps: normalization, parameterization, feature extraction, a similarity comparison, and a decision (Fig. 1). The first three steps comprise the front end of the recognizer and concern information reduction or elimination of redundancies in the input data sequence, very much like what is done in reducing bit rate for speech coders [4]. Since the objective is to extract a speaker identity, and not to preserve sufficient information to reproduce the speech (as in speech coding) or to determine the text message (as in speech recognition), data reduction can eliminate many aspects of the speech signal which affect naturalness and intelligibility.

The initial normalization step attempts to eliminate variability in the input speech signal due to environment (e.g., background noise, recording level). The simplest form of normalization adjusts maximum signal amplitude to a standard level to account for variations in recording level, distance from the microphone, original speech intensity, and loss in transmission. Such variations are assumed to be constant or slowly changing, which permits updating an amplitude scaling factor (by which the received signal is multiplied) at long intervals, corresponding typically to utterances bounded by easily identifiable pauses.

Major data reduction occurs in converting the signal into parameters and features. Acoustic *parameters* derive directly from standard speech analysis and coding methods (e.g., linear predictive (LPC) coefficients, amplitudes of filter bank outputs, fundamental frequency—F0), while *features* denote the outputs of a further optional reduc-

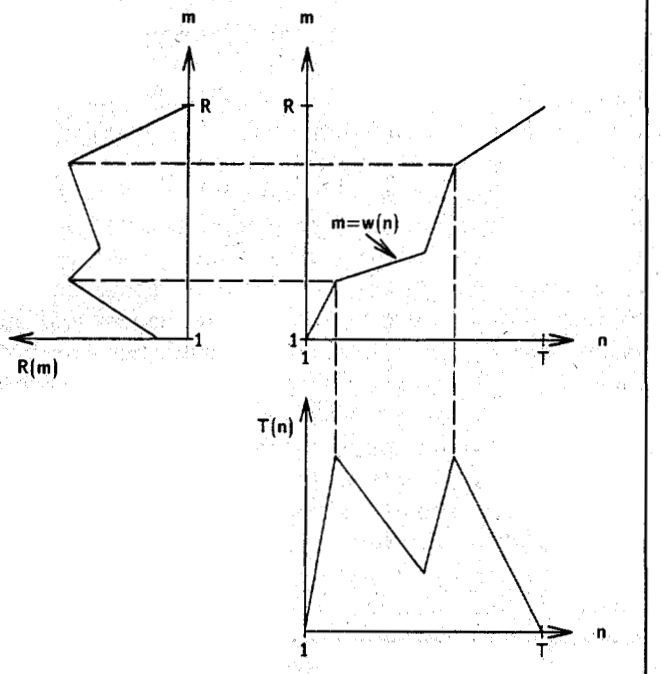

Figure 2. Example of nonlinear time alignment of a test $T(n)$ pattern and reference $R(m)$ pattern.

tion step (e.g., locating formants in spectral patterns) [5]. To parameterize the speech signal efficiently, a standard speech model is used, which separates excitation and vocal tract response [5]. Excitation is typically represented in terms of a voicing decision, overall amplitude, and an F0 estimate (during speech identified as voiced). There is less agreement on which spectral parameters to use, although most recognizers represent the spectral envelope with about 8–14 coefficients. Common parameters are: LPC coefficients, mel-based cepstral coefficients, channel energies in a channel vocoder, or some form of reduced discrete Fourier transform. They all attempt to capture in a few parameters enough spectral information to identify speakers.

The focus of the recognition process is a comparison between templates of parameter/feature representations of an unknown or test speech signal and of a reference signal. In ASI, the test template (derived from the test signal) is usually compared with all reference templates stored in data memory, but the memory can sometimes be partitioned for more efficient search procedures. The comparison involves a measure of how similar the test and reference templates are. The reference template most closely matching the test is usually chosen, yielding an output of the speaker identity corresponding to that reference. However, if the match is relatively poor or if other references provide similar matches, a decision can be postponed pending new input and the speaker asked to repeat his utterance.

The decision to accept or reject usually depends on a threshold: if the distance between a test and a reference template exceeds a threshold, the system rejects a match. Depending on the costs of each type of error, an overall cost can be minimized by biasing the decisions in favor of the least costly error. A low threshold is preferred because false acceptances are usually more expensive (e.g., admitting an impostor to a secure facility might be disastrous, while excluding some authorized personnel is usually only annoying). In financial transactions authorized by telephone voice, the amount of the transaction might be set inversely proportional to the confidence the system has in its decision (e.g., how small the template distance is).

The template representation of a small portion or *frame* of speech using N features or parameters can be viewed as an N-dimensional vector (or point in N-dimensional feature space). A memory of reference templates is established during training when each speaker utters a controlled vocabulary, and acoustic segments are converted into features identified with that speaker. To represent utterances of words or sentences, N may include time variation in the features. In general, feature templates are M-dimensional vectors (or $L \times N$ matrices), where $M = LN$ and L N-dimensional vectors are extracted at uniformly spaced intervals for each utterance. A fixed value of L for all utterances is assumed here, which implies linear time normalization. The problems of time normalization and aligning templates are discussed later.

Euclidean and Mahalanobis distances

The similarity between two templates can be viewed as inversely proportional to their separation distance in N- or M-dimensional space. One standard measure is a quadratic distance, which for two N-dimensional templates $\mathbf{x}$ and $\mathbf{y}$ is:

$$d(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^{T}\mathbf{W}^{-1}(\mathbf{x}-\mathbf{y}) \qquad (1)$$

where $\mathbf{W}$ is a positive-definite matrix which allows different weighting for individual features of the template, depending on their utility in identifying the speakers in the feature space. The common Euclidean distance sets $\mathbf{W}$ to be the identity matrix $\mathbf{I}$, whereas the general Mahalanobis distance sets $\mathbf{W}$ to be the autocovariance matrix corresponding to the reference vector. (Individual matrices could be used for each reference template, but an average $\mathbf{W}$ is more commonly used without significant accuracy loss, because: 1) it is difficult to obtain accurate estimates for speaker-individual $\mathbf{W}$ from limited training data; and 2) using one $\mathbf{W}$ matrix is efficient.) Other distance measures, such as correlations [6] and city-block distances, are sometimes used in ASR, but may give inferior results.

The Mahalanobis distance has origins in statistical decision theory, where each utterance may be viewed as a point in N-dimensional space and the utterances describe a multivariate probability density function in that space. Assuming recognition among equally likely speakers, Bayes' rule specifies choosing the speaker whose density was most likely to have generated the test utterance. Because of the difficulty of estimating density functions from a limited amount of training data, a parametric form

of density, such as a Gaussian, which can be simply and fully described by a mean vector μ and a covariance matrix **W**, is often assumed.

The density of a feature vector **x** for speaker i would be

$$P_i(\mathbf{x}) = (2\pi)^{-M/2}|\mathbf{W}|^{-1/2} \exp[-(1/2)(\mathbf{x} - \mu_i)^T \mathbf{W}^{-1}(\mathbf{x} - \mu_i)], \quad (2)$$

where μ_i is the mean of **x** for speaker i's utterances and $|\mathbf{W}|$ is the determinant of **W**. Given a test vector **x** for recognition, speaker j is selected in ASI if

$$P_j(\mathbf{x}) > P_i(\mathbf{x}) \quad \text{for all speakers } i \neq j . \quad (3)$$

(For ASV, speaker j is accepted if and only if $P_j(\mathbf{x})$ exceeds a threshold.) Applying a (monotonic) logarithm transformation and eliminating terms constant across words (i.e., a common $|\mathbf{W}|$), Eqs. 2–3 reduce to minimizing the Mahalanobis distance of Eq. 1, using μ_i in place of **y**. The simpler Euclidean distance, which sets **W** equal to the identity matrix, trades off optimal recognition accuracy for fewer calculations and fewer parameters to be estimated. The Euclidean distance is optimal only if the **N** parameters are mutually independent and have equal variances (i.e., contribute equally to the distance measure).

Ideally, repetitions of the same speech segment (e.g., pronunciations of a phoneme in different linguistic contexts) would yield consistent feature measurements and therefore small *clusters* in the feature space, and different speech segments would provide very distinct measurements and hence widely separated points in the space. The best features show little variance for utterances from a single speaker and large variance for utterances from different speakers. For the Mahalanobis distance, the elements along the diagonal of **W** reflect these variances, with large **W** values for good features; small values for poor features discount their weighting in calculating an overall distance. It simplifies calculation considerably if the features are orthogonal (allowing a diagonal **W**) or orthonormal (allowing $\mathbf{W} = \mathbf{I}$), but a *principal component* or Karhunen-Loeve transformation on the parameters to achieve orthogonality significantly increases calculation in determining the test template. As the number of templates in memory grows with large populations, however, distance computations dominate ASI computation, suggesting the utility of reducing templates to a small set of independent features.

For both speech and speaker recognition, data reduction of the input speech via parameterization and feature extraction is important for efficiency. While template matching and distance measures are common to both applications, the reference templates may store quite different information for speech and speaker recognition. ASR templates emphasize speaker characteristics rather than word information. Just as template memory grows linearly with vocabulary size in speech recognition, ASR memory expands similarly with the number of speakers. Memory grows with both vocabulary and population size, but ASR employs much smaller vocabularies (e.g., a few digits) than most speech recognition systems. Alternative ASR, based on feature extraction rather than matching parametric templates, has small memory requirements with no correspondence to vocabulary, but is usually more difficult to implement than template ASR.

Timing considerations

In discussing distance measures, a comparison of stationary sounds was implicitly assumed, where a single feature representation of each utterance (from one frame or from an average of several frames) would suffice. Most attempts at ASR, however, use vocabularies which involve sequences of different acoustic events. The test and reference utterances are subdivided in time, yielding sequences of parameter vectors. Most commonly, each speech signal is divided into equal-duration (possibly overlapping) frames of about 10–30 ms, each producing a vector.

Since automatic segmentation of utterances into meaningful linguistic units (e.g., phones, syllables) is difficult, templates are usually compared frame-by-frame, which leads to alignment problems. Utterances are generally spoken at different rates, even for a single speaker repeating the same word. Thus, test and reference utterances normally have different durations. One way to allow linear frame-by-frame comparison is to normalize the interval between frames so that a common number of frames is used for all templates. For example, if a typical word duration is 400 ms, and a time resolution of 20 frames/word is desired, the frame intervals would exceed 20 ms for words longer than 400 ms and be proportionately less for shorter words. Such *linear time normalization*, or *warping*, can be accomplished either through frame interval adjustment before parameterization or through decimation/interpolation of the feature sequence.

Accurate time alignment is crucial for good ASR performance. Matching templates corresponding to the same speaker results in a small distance when parallel acoustic segments in the two templates are compared. Linear warping is insufficient in general to align speech events, because the effects of speaking rate change are nonlinear: vowels and stressed syllables tend to expand/contract more than consonants and unstressed syllables. Thus linear warping of two utterances of the same "word" often aligns acoustic segments from different phones. For example, the main difference between a long and a short version of the word "sues" occurs in the /u/ duration. Linearly compressing the long version to the same number of frames as the short one would align some frames at the start and end of /u/ in the long version with frames from /s/ and /z/ in the short version. A large distance results for these frames, since vowels and fricatives are very different spectrally. If enough frames are misaligned, the overall distance for the word may be large enough to reject a positive match decision, even if they represent the same word spoken by one person.

Dynamic time warping

Most high-performance speaker (and speech) recognizers address the problem of alignment by nonlinearly

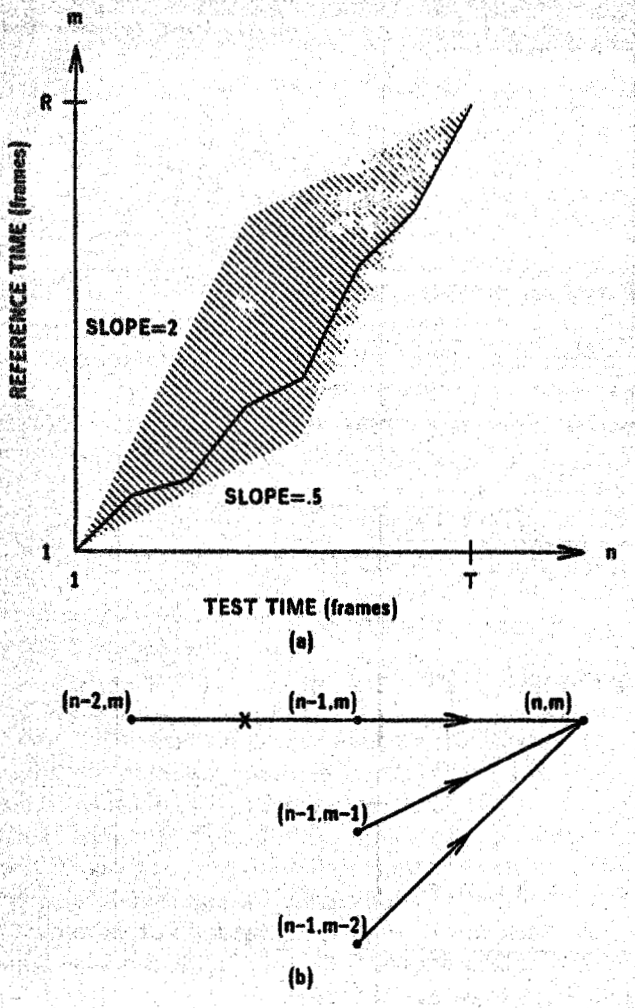

Figure 3. a) Typical DTW plot illustrating the optimal warp path $w(n)$ mapping the test time axis n into the reference axis m. If the template endpoints are forced to match, and the warping is restricted to lie within slopes of 1/2 and 2, the shaded area shows the subset of points that is considered; b) Permitted transitions to the grid point (n, m). Since two horizontal transitions are not allowed, the step marked "X" is illegal.

warping one template in an attempt to align similar acoustic segments in the test and reference templates. The procedure, called *dynamic time warping* (DTW), combines alignment and distance computation through a dynamic programming procedure [7–8]. Deviations from a linear frame-by-frame comparison are allowed if the distance for such a frame pair is small compared to other local comparisons. In the absence of specified segment boundaries, DTW aligns templates by finding a time warping that minimizes the total distance measure, which sums the frame distances in the template comparison.

The underlying assumptions of basic DTW are that: 1) global variations in rate for a speaker uttering the same word on different occasions can be handled by linear time normalization; 2) local rate variations (which render linear normalization inadequate) are small and best dealt with using distance penalties known as *local continuity constraints;* 3) each frame of the test utterance contributes equally to recognition; and 4) a single distance measure applied uniformly across all frames is adequate. The first two assumptions appear reasonable, but the latter two have recently led to considerable refinements in DTW [9].

Consider two patterns **R** and **T** of R and T frames each, corresponding to the reference template and test template, respectively. DTW finds a warping function $m = w(n)$, which maps the time axis n of the test template into the time axis m of the reference template (Fig. 2). Proceeding frame-by-frame through the test template, DTW searches for the best frame in the reference template against which to compare each test frame. The warping curve derives from the solution of an optimization problem

$$D = \min_{w(n)} \left[\sum_{n=1}^{T} d(\mathbf{T}(n), \mathbf{R}(w(n))) \right], \tag{4}$$

where each $d(,)$ term is a frame distance between the nth test frame and the $w(n)$th reference frame. D is the minimum distance measure corresponding to the 'best path' $w(n)$ through a grid of $T \times R$ points (Fig. 3).

Theoretically, $T \times R$ frame distances must be calculated for each template comparison, corresponding to matching each test frame against every reference frame. In practice, constraints restrict the search space so that typically only about 25–35% of the matches are performed. This nonetheless leads to a significant increase in computation: a linear frame-by-frame comparison for a typical 25-frame template requires only 25 distance calculations, while DTW needs about 150–200. Since DTW calculation usually increases as the square of T (compared to calculation proportional to T for a linear path), computation becomes excessive for long templates involving several words at a time. Computation can be limited by restricting the warp path to stay within a 'window' of $\pm X$ frames of the linear path, which leads to about $2XT$ distance calculations. However, X must often be expanded for good recognition of longer utterances.

For a point $[n, m]$ in the grid, the minimum accumulated distance $D_a(n, m)$ from the start (point [1, 1]) can be recursively defined:

$$D_a(n,m) = d(T(n), R(m)) + \min_{k \le m} [D_a(n-1,k)p(n-1,k)], \tag{5}$$

where p represents a penalty for deviating from the linear path or for violating continuity constraints, and d is the frame distance at point $[n, m]$. Some systems use a binary penalty function p (e.g., =1 for an acceptable path or $=\infty$ for an unacceptable one [10]), while others allow multivalued penalties which increase gradually as the path deviates further from the linear ideal [51]. Limiting the search to the range $k \le m$ reflects an assumption that the warp path should be monotonic. The range is usually limited further by the continuity constraints, such as those in Fig. 3b, which insist that the reference index m advance at least one frame every two test frames, and that at most one

reference frame can be skipped for each test frame. Extrapolating these local constraints to the global grid in Fig. 3a, minimum and maximum slopes of 1/2 and 2, respectively, restrict the grid points that must be examined to the shaded region.

Differences between speech and speaker recognition in template matching

Since typical speech parameters and features contain both segmental and speaker information, ASR systems often use methods and templates identical to those for speech recognition, except that the templates are indexed by speakers rather than by words. In general, reference templates are stored for each speaker saying one or more utterances, each typically containing one or more isolated words. To minimize memory and run-time computation, simple systems use one template per speaker clustered (averaged) over repetitions of one word during a training period. Performance can be improved (at increased cost) by storing templates for several words and/or several repetitions of the same word without clustering.

When compared to speech recognition, ASR has a potential computational advantage during template matching. Text-dependent ASR always compares templates of the same words, whereas in speech recognition the same words are only compared when the test and reference templates happen to match. Since variation in spectral patterns is usually much greater across different words by the same speaker than across different speakers for the same word, ASR warp paths have simpler shapes than those in speech recognition. As in speech recognition, speaking rate variability necessitates nonlinear time alignment when comparing speaker templates, but simple metrics may be used in ASR to find the warp path. For example, in addition to LPC or spectral distances typically used to find a warp path in speech recognition, an energy distance (e.g., the absolute value of the difference between the energies in two template frames) is often used to align ASR templates [2, 12, 13] (Fig. 4). The recognition decision employs more precise spectral distances, but such computation-intensive measures are only needed along the warp path specified by the simple energy distance. Utterances of isolated monosyllabic words may allow very simple linear alignment, e.g., lining up the frames of maximum energy between templates.

Statistical or dynamic features

Since acoustic cues to a speaker's identity are spread throughout each of his utterances, many systems utilize templates of averaged parameters rather than the full time sequence of parameters used in speech recognition. This statistical approach is most useful in text-independent cases, since the time sequences of training and test utterances do not correspond. The simplest statistical approach conceptually (although not computationally) takes long-term averages of speech parameters over all available data from each speaker to yield one mean vector template.

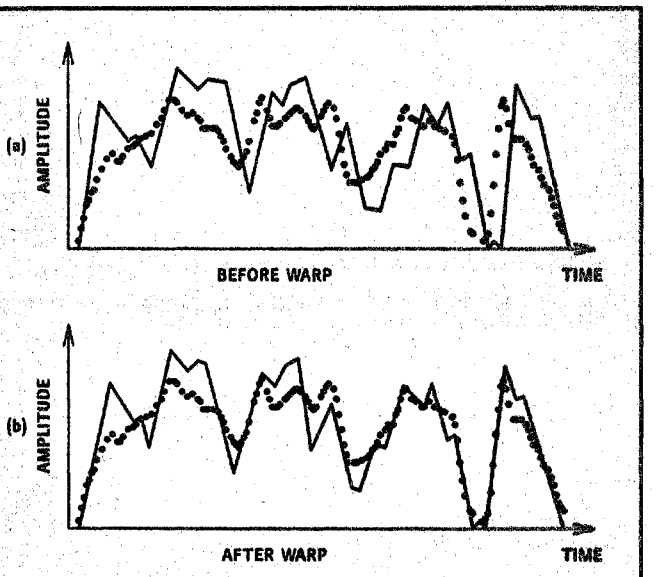

Figure 4. Time alignment of the amplitude contours for two templates. The contour from the test utterance (solid curve) is aligned to that of the reference utterance (dotted curve). The upper panel shows the curves after linear warping to align endpoints, while the bottom one shows the curves after nonlinear alignment using an amplitude distance. After alignment, the utterance is divided into 20 equal sections, where each section contributes to an overall distance measure in proportion to its usefulness. The recognition parameters used were F0, intensity, and two widely-spaced LPC coefficients. (After Rosenberg ©IEEE [2].)

Long-term spectra can yield high accuracy for normal speech and even for speech spoken under stress, but not for disguised speech [14]. Unfortunately, the long test utterances needed to obtain averages (usually more than 30 s) usually preclude this approach from use in real-time applications.

Statistical averaging has also been applied successfully in text-dependent cases to reduce run-time computation by using templates of few dimensions. Compared to DTW computation, which increases as the square of template duration, computation of statistics (e.g., moments, covariances) over an utterance typically increases only linearly with utterance length. Furthermore, calculation of statistical means requires no multiplications. For example, similar recognition results on 40-frame words have been obtained with either standard DTW template matching or a single distance measure involving a 20-dimension vector employing features of F0 and 12 LPC coefficients [15].

Vector quantization

Since recognition via long-term statistics is often impractical for real-time, text-independent applications, one might try to identify specific sounds in the test speech and compare them with stored sounds for each speaker. However, automatic phone segmentation and recognition of continuous speech is difficult. Alternative techniques for text-independent applications have recently been devel-

oped which attempt to compare parallel test and reference phones without explicitly locating them. One such technique, vector quantization (VQ), has been successfully applied to both speech recognition and ASR in similar ways and for similar reasons: e.g., to avoid the problem of segmenting speech into meaningful subunits. VQ is a coding technique typically used to lower transmission bit rate. Applied to ASR, VQ provides an alternative to DTW. The data-reduction efficiency of VQ in parameterizing speech is paramount for speech coding applications and is useful in ASR to minimize memory. As in speech recognition, however, the primary advantage of VQ for ASR lies in the codebook approach to determining the similarity between utterances.

In VQ speech coding, frames of speech are typically represented by k (spectral) parameters, which are coded together as a block or vector [16]. If the vector elements are correlated in some way, such coding is more efficient than treating the k parameters individually. A properly chosen set of 1024 spectra (2^{10} for a 10-bit VQ) should be able to adequately represent the envelope spectra for virtually all possible speech sounds.

The disadvantage of VQ lies in increased complexity in the coder analysis. After the normal analysis is complete (yielding k scalar parameters for a given analysis frame), the coder must then determine which k-dimensional vector from among a set of M possibilities stored in a *codebook*, corresponds most closely to the set of scalar parameters. A distance measure (e.g., Euclidean or Mahalanobis distance) is used as a criterion in both the design and operation of the codebook.

The key issues in implementing VQ concern the design and search of the codebook. Codebook creation involves the analysis of a large training sequence of speech, typically a few minutes long [17], sufficiently varied as to contain examples of phonemes in many different contexts. An iterative design procedure is used to converge upon a locally optimal codebook (optimal in the sense that the average distortion measure is minimized across the training set). Compared to scalar coding, the major additional complexity of VQ lies in the time necessary to search the codebook for the appropriate codeword to represent a given speech vector. In a *full codebook search*, the vector for every frame is compared with each of the M codewords, requiring M distance calculations (each having k squaring operations and $2k - 1$ additions, in the case of the simple Euclidean distance). The design of the codebook is a one-time problem, for which large computation is acceptable, if the codebook is to be used for a long time. However, for real-time coder applications, the cost of a full codebook search must be balanced against improved system performance with larger k.

A separate VQ codebook is usually designed for each combination of speaker and vocabulary word, based on one or more utterances of the word. Each test template is evaluated by all codebooks, and the speaker corresponding to the codebook which yields the lowest distance measure is selected as the ASI output. (For ASV, the codebook distortion is compared to a threshold.) In its simplest form, codebooks have no explicit time information, either in terms of temporal order or relative durations, since the codebook entries are not ordered and can derive from any part of the training words. However, implicit durational cues are partially preserved because the entries are chosen to minimize average distance across all training frames, and frames corresponding to longer acoustic segments (e.g., vowels) are more frequent in the training data. Such segments are more likely to specify codeword positions than the less frequent consonant frames, especially in small codebooks.

Codebooks of more than 1000 entries appear necessary for good speech coding, but smaller codebooks suffice for recognition. One ASR study designed a 1000-entry codebook so that the entire speech space of 12 LPC coefficient dimensions was covered with relatively fixed separation ("radius") between entries [18]. To simplify computation, only the most frequently occurring 400 entries, which included 90 percent of the training speech frames within a radius of an entry, were used to represent the speaker templates. 100-s training samples from each of 11 speakers were used to select 40 of the 400 entries which best represented each speaker. An entry was selected if: 1) its average percentage occurrence in the speaker's training data exceeded the mean appearance in the training data for all other speakers; and 2) the entry's percentages were stable with increasing amounts of training data. The basic idea is to find spectra commonly used in the general speech of each speaker which are distinctive to his or her voice. The 40 entries for each speaker covered 25–40 percent of his or her training data, but only 7–12 percent of the data for other speakers. Thus, during recognition, the unknown speech was vector-quantized with the 400 entries, and the speaker model whose 40 entries most overlapped with the set of entries for the unknown was selected. Recognition accuracy with test utterances of 10 s was 96 percent, which demonstrated that high performance is possible with short test utterances in text-independent applications.

Besides avoiding segmentation and allowing short test utterances, VQ is computationally efficient as compared to storing and comparing large amounts of template data in the form of individual spectra. Thus, VQ can be useful for text-dependent as well as text-independent recognition. One recent experiment employed isolated digits (0–9) for both the training and testing of 100 speakers, with one codebook per speaker [19]. Recognition error decreased substantially as a function of both codebook size and test utterance duration, with errors below 2 percent for 10-digit tests and 64-word codebooks (Fig. 5). Increasing codebook size raises computation but decreases the likelihood of error by reducing the standard deviations of the distortions. The performance increase with duration depends on the degree of correlation among words in the test utterance: when the test utterance of 10 different digits was replaced by a single digit repeated 10 times, error increased dramatically. While this study was partially text-independent (i.e., the speakers were free to say the

digits in any order), true text-dependent recognition allows incorporation of timing information into the distance measures to lower error rates.

In summary, ASR via VQ can yield high accuracy in both text-dependent and independent cases, with relatively short test utterances. As in speech recognition, VQ often has the advantage of smaller reference memory than word templates in the DTW approach. For example, in the first study above, 400 LPC vectors covering all speakers are stored, and each speaker needs storage of only 40 indexes. The second study requires 64 vectors per speaker. Both of these are less than with the storage for DTW (25–40 vectors/word × 10 digits).

Cepstral analysis

Among transformations of LPC parameters (e.g., reflection coefficients, log-area ratios), the cepstral representation has been suggested as superior for ASR [20]. Cepstral coefficients yield high recognition accuracy, and are invariant to fixed linear spectral distortions from recording and transmission (e.g., telephone) environments. Excellent results have been demonstrated through template matching patterns of 18-dimensional cepstral vectors [13]. As an example, the process shown in Fig. 6 inputs all-voiced sentences of 6–7 short words and calculates ten cepstral coefficients every 10 ms via LPC. (Performance was similar whether the cepstra were calculated directly with Fourier transforms or with LPC, but the LPC method was twice as fast.) The mean value for each coefficient over time is subtracted from each coefficient function, which yields a signal that minimizes environmental and intra-speaker effects. The coefficients in each 90-ms section of the utterance are then expanded into orthogonal polynomials so that each coefficient is represented by the slope of its function, in addition to the coefficient itself. An 18-element feature vector for each 10-ms frame consists of the 10 cepstral coefficients plus 8 of the 10 polynomial coefficients. Another cepstral study suggested that low-order coefficients are more effective for ASR than high-order ones [21].

Orthogonal LPC parameters

In the search to isolate speaker-dependent features from speech parameters that also carry segmental information, LPC coefficients have been frequently examined because they are covenient to calculate and usually model speech spectra well. Besides the standard ASR methods utilizing direct LPC coefficients or cepstral features, another LPC transformation known as *orthogonal linear prediction* has demonstrated some recognition success

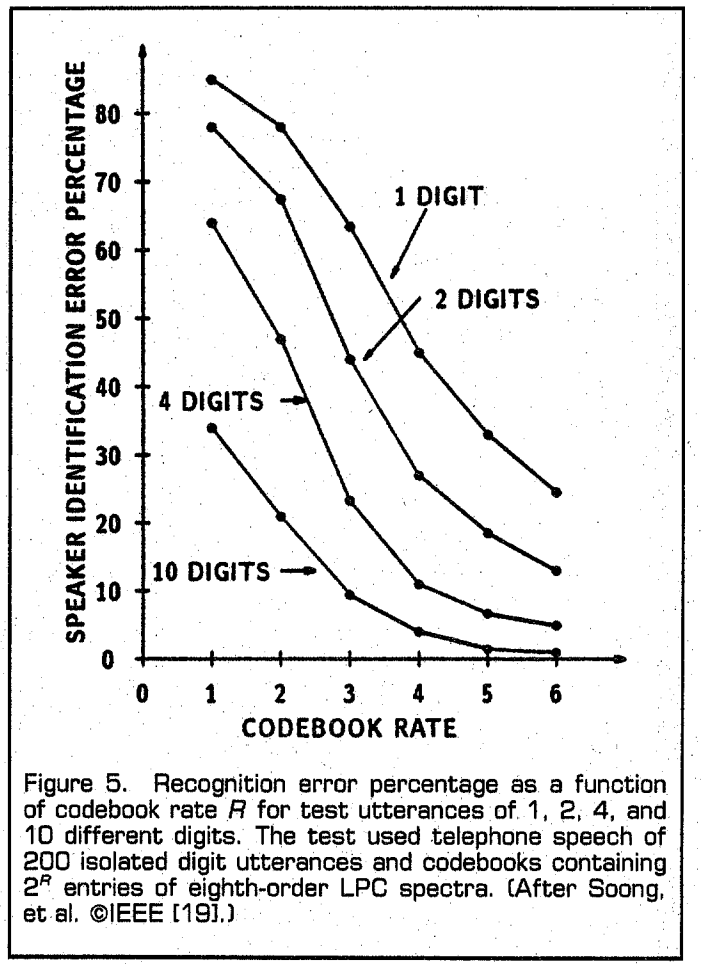

Figure 5. Recognition error percentage as a function of codebook rate R for test utterances of 1, 2, 4, and 10 different digits. The test used telephone speech of 200 isolated digit utterances and codebooks containing 2^R entries of eighth-order LPC spectra. (After Soong, et al. ©IEEE [19].)

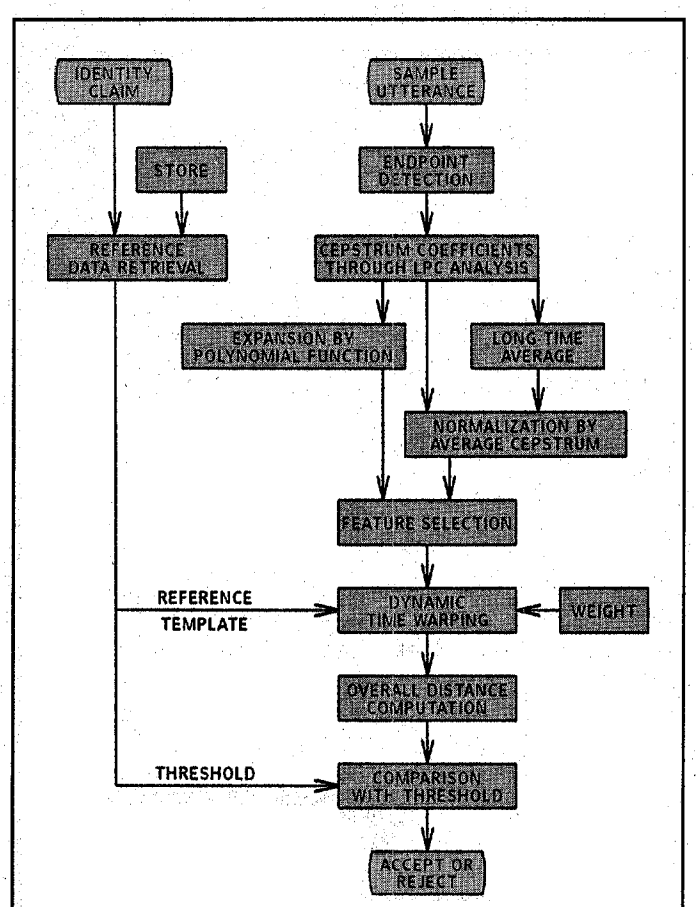

Figure 6. Block diagram of a text-dependent recognizer using orthogonal polynomials based on tenth-order LPC cepstra. (After Furui ©IEEE [13].)

[22,23]. Orthogonal transformations are usually intended to concentrate information from a set of p parameters into a smaller set by rotating and scaling p-dimensional space so that: 1) the revised parameters become independent; and 2) most of the variance (i.e., information) rests in a low-order subset of the revised parameters. Except for the increase in computation, such an approach might be useful for speech coding or recognition. For ASR, however, it was proposed that, while the low-order, high-variance orthogonal LPC coefficients would reflect phonemic information, the high-order, low-variance ones might contain cues dependent on speakers and on the recording environment [22]. Using 12th order LPC on 4-kHz speech, the variances of the last seven orthogonal coefficients are very small compared to those of the first five, indicating that the latter reflect segmental information (e.g., voicing and the first three formants), which vary throughout an utterance, whereas the high-order coefficients contain information that does not change during the utterance. If the latter parameters were different between speakers uttering the same sentence, they would be useful for ASR.

Using only six coefficients, one study recognized 94.4 percent of 21 speakers [22]. When the system was trained and tested using different utterances (i.e., the text-independent case), accuracy only dropped to 94 percent. However, since other recognizers using LPC coefficients have demonstrated higher accuracy without the need for orthogonal calculations, the orthogonal approach has not seen further development since 1981.

FEATURES WHICH DISTINGUISH SPEAKERS

For simplicity, most ASR systems use standard speech parameters such as 8–12 LPC coefficients or 17–20 band-pass filter bank energies [3,24]. However, viewing ASR as a problem of separating probability densities in N-dimensional space, better results and lower computation may be obtained by more careful selection of the parameters or features that comprise the space. Ideally, the space should use a few independent features that have similar small intraspeaker variances and large interspeaker variances, which lead to compact, widely separated clusters for individual speakers. Independent features eliminate calculations for the off-diagonal elements of **W** in Eq. 1, and features of equal weight permit a Euclidean distance. In practical terms, the features should also be easy to measure, be stable over time, change little in different environments, and be unsusceptible to mimicry [25].

One way to select acoustic features for ASR is to examine what features correlate with human perceptions of voice similarity. When multidimensional scaling analysis is applied to such judgments, the following features account for most of the speaker variance: F0 and the first three formant frequencies F1, F2, F3 [26], word duration, and speaker sex and age [27]. Since sex and age are not acoustic features, it appears that F0, timing cues, and spectral cues (e.g., the first three formants) are the most likely candidates for speaker recognition.

The two sources of speaker variation, physiological and behavioral differences, lead to two types of useful features. *Inherent* features are relatively fixed for a speaker and depend upon the anatomy of his vocal tract. While they can be affected by health conditions (e.g., colds which congest the nasal passages), inherent features are less susceptible to the mimicry of impostors than *learned* features. The latter refer to the dynamics of vocal tract movement—i.e., the way a speaker talks. While learned features can be used to distinguish people with similar vocal tracts, impostors usually find it easier to fool recognizers based on learned features than those based on inherent features [2]. Statistical features based on long-term averages reflect inherent features more than learned ones and are suitable for text-independent ASR.

Effectiveness measures

One common measure of effectiveness for individual features is the *F-ratio* [25,28] which compares inter- and intra-speaker variances:

$$F = \frac{\text{variance of speaker means}}{\text{mean intraspeaker variance}} \tag{6}$$

The numerator is large when values for the speaker-averaged feature are widely spread for different speakers, and the denominator is small when feature values in utterance repetitions by the same speaker vary little. (The denominator averages intraspeaker variances over all speakers.) While features with higher F-ratios do not guarantee fewer recognition errors, the F-ratio has successfully helped design ASR systems. The F-ratio tends to be high for features where one or two speakers are very different from the rest, which suggests that F-ratios are most useful in eliminating poor features rather than choosing the best [29]. Generalizing the F-ratio to M parameters and including effects of feature interdependence, a *divergence* measure is defined as

$$D = < [\mu_i - \mu_j]\mathbf{W}^{-1}[\mu_i - \mu_j]^T >_{i,j}, \tag{7}$$

where $<>_{i,j}$ represents averaging over all speakers $1 \le i,j \le N$.

Techniques to choose features

A more direct way to evaluate the utility of features for ASR involves probability-of-error criteria in a *knock out* procedure [28]. Starting with a set of L features, all L subsets of $L-1$ features are used in a recognition system to determine which subset yields lowest error. The feature not used in this best subset is viewed as the least useful feature and is "knocked out" of consideration. The process is repeated with $L-1$ subsets of $L-2$ features, etc., leading to a ranking from worst to best features. The single feature in the last round is the "best". One study examined a total of 92 features, including formant frequencies and bandwidths for vowels, resonances during nasals and fricatives, F0 statistics and dynamics, and some timing measurements (e.g., formant trajectory slopes in diph-

thongs, voice-onset time for stops) [28]. Among the most useful features were F2 in nasals, F2-F3-F4 in vowels, and mean F0.

For text-independent applications where long-term feature averages are used, a dynamic programming evaluation procedure [30] can find a set of features with better recognition accuracy than either a knock-out feature set or a set of LPC reflection or cepstral coefficients (using the same number of features). A more computationally efficient way to select a set of features than the knock-out method is the *add-on* procedure [29]. Initial recognition tests are done with each of L features, one at a time, selecting the best single feature. Then tests with two features, including the best one, select the second-best feature. The cycle repeats until the desired number of features has been chosen, or until recognition error falls below a threshold.

Spectral features

Spectral features, such as formants in specific sounds, tend to be very useful for ASR. Formants in retroflex vowels and nasals [6], in particular, are said to yield good recognition performance. Vowels, nasals, and fricatives (in decreasing order) are commonly recommended for ASR because: 1) they are relatively easy to identify in speech signals; and 2) their spectra contain features which reliably distinguish speakers. Nasals are useful because the nasal cavities of different speakers are distinctive and not easily modified (except via colds). However, the difficulties of phone segmentation in text-dependent cases and phone identification in text-independent applications have led most recognizers to avoid examining specific sounds and to use long-term spectral averages or general spectral distance measures during template matching.

Prosodic features

Both speech and speaker recognition rely primarily on spectral features, but ASR makes more use of prosodics (F0, in particular) than current speech recognition systems do. Mean F0 averaged over all test data from an unknown speaker frequently seves as a simple feature to classify speakers coarsely into broad groups (e.g., male adults, female adults, children). The dynamics of F0 as measured in a contour over time, however, may be a more powerful feature in text-dependent ASR [12, 25, 2], although some measures of F0 appear to be highly variable over different recording sessions [28].

One study sampled F0 at 40 equal intervals in an all-voiced utterance averaging 2 s, and then compressed those 40 features via a Karhunen-Loeve transformation into 20, which accounted for 99.5 percent of the variance [31]. With these 20 features, the Mahalanobis distance attained 97 percent recognition for ten speakers. Using the first four moments of the F0 values (statistical, not dynamic, measures) yielded 78 percent recognition. These results demonstrated both the utility of dynamic F0 data and the Mahalanobis distance. To put the utility of F0 in perspective, however, a feature set of 12 cepstral parameters needed only 0.5 s of these 2-s utterances to achieve 98 percent accuracy [20]. Thus, a set of spectral features is more powerful than F0 for ASR. While F0 dynamics appear to be useful, some studies have found only mean F0 to be a reliable speaker cue over time [28].

F0 is sometimes combined with energy to provide two text-dependent ASR features which are relatively independent and simpler to obtain than spectral features [2]. ASR rarely exploits temporal features, however, even though it is intuitively clear that phone durations and speaking rate are distinctive aspects of speakers. Lack of knowledge of how speakers use durations is evident in poor durational models for speech synthesis, and it is not surprising that duration is seldom used in ASR. One recent study had good results by avoiding modeling phone durations, simply calculating 40 statistical timing measures dealing with "speech bursts" (periods when energy exceeds a threshold) [32]. For a fixed text, the pattern of such bursts reflects speaking style in terms of rate and segment durations. Other studies have shown the usefulness to ASR of word durations, voice onset time [25], and formant slope in diphthongs [28].

SYSTEM DESIGN

In speech recognition, task difficulty is relatively easily measured in terms of vocabulary size, allowed syntactic structure, and confusability of words based on phonemic similarity. For ASR, however, the "vocabulary" is a set of speakers, whose characteristics are much more difficult to describe than the phonemic compositions of words. We know that a vocabulary of "yes" — "no" is easier to recognize than one of "B" — "P", but only human perceptions of voice similarity can provide a measure of the inherent difficulty of discriminating two voices. Comparing ASR experiments using different sets of speakers is often unreliable, since one study may use a very homogeneous set of speakers (e.g., one sex, narrow age range, raised in a limited geographical area), while another could use a heterogeneous set (e.g., males and females of varying ages, different dialects). The latter, of course, yields much higher recognition accuracy. Another factor limiting the utility of many ASR studies is that few employ more than 100 speakers [2, 11].

Data collection

In addition to speaker selection, the time span over which speech is collected is of crucial importance to ASR performance. Speaking style often changes substantially over time. Experiments using reference and test data from the same recording session usually yield high recognition accuracy, which is misleading, since practical applications often compare test data with reference data that was obtained much earlier. Performance usually decreases (often dramatically) as the interval between training and testing increases [15]. Reference data must be updated periodically for best results [13]. Some systems consider each recognized utterance as new training data, revising the reference templates to reflect changes in a person's speaking style over time.

In any pattern-recognition task, training and test data should be kept separate. If the same utterances are used to train and test a speaker recognizer, artificially high accuracy often results [21]. When training and test data are in common, it is difficult to determine whether the system has been designed to take advantage of specific speaker characteristics which may not be reliable for new data. Given K utterances/speaker as data, a common procedure trains the system using $K - 1$ as data and one as test, but repeats the process K times, treating each utterance as test once. Technically, this "leave-one-out" method designs K different systems, but it verifies the accuracy of the system design using a limited amount of data, while avoiding the problems caused by common train/test data.

Sequential decision strategy

Most ASV applications require real-time processing, where the system responds immediately to accept or reject a speaker. Such systems may employ a sequential decision procedure, in which borderline decisions are postponed pending further test input. Rather than use a single threshold to accept or reject, two thresholds divide the distance range into three choices: accept if the distance falls below the lower threshold; reject if it's above the higher threshold; and ask for more input if the first distance lies between thresholds. Such an approach allows shorter initial test utterances and thus faster response time, while avoiding errors in close cases [13].

For example, one well tested ASR system samples utterances of four monosyllables every 10 ms with 17 bandpass filters uniformly spaced from 300 to 3000 Hz. Templates consist of 24 17-dimensional spectral vectors: six from each syllable, spaced at 20-ms intervals centered around the time of maximum energy in each syllable. In one evaluation using 50 speakers and 70 impostors, the error rate was 1.6 percent with one test phrase, but dropped to 0.42 and 0.23 percent with two and three phrases, respectively [2]. A sequential decision method averaged 1.3 phrases per test; customers were rejected 0.3 percent of the time, while the impostor acceptance rate was only 1 percent.

Multiple stage recognition

In ASI, computation and response time usually increase linearly with population size (i.e., the number of speakers whose templates are stored), because each speaker's template must be examined. One way to minimize computation is to set up a hierarchy of reference patterns so that speakers are clustered into groups which can be rapidly identified from a test utterance. For example, speakers could be classified according to mean F0, so that only a relatively small subset of speakers whose F0 is close to that of the test need be examined further. The concept is very similar to the idea of *cohorts* in speech recognition [33,34], where reliable coarse segmentation and feature extraction eliminate most of the vocabulary from consideration, and where finer spectral analyses chose the response from among the remaining alternatives. This multistage method may also help in recognizing speakers with varying dialects.

Effects of different communication channels

Since many ASR applications involve telephone speech or speech subject to other environmental distortions, the effects on recognition accuracy due to environment must be examined. It was noted earlier that cepstral speech representations have the advantage of being invariant under linear distortions, suggesting that variation in cepstral coefficients about their means might be good ASR features. Among the distortions that a telephone link introduces into a speech transmission is bandpass filtering, which preserves speech only in the range 300–3200 Hz. Successful ASR on such limited-spectrum speech has been demonstrated using filtered logarithmic spectra [23] and blind deconvolution [18].

Telephone distortions, however, go beyond band-limiting, and subsequent studies using real telephone speech indicate significant problems for ASR. Typical long-distance telephone links have an average signal-to-noise ratio of 27 dB [35]. Because different links demonstrate large variation compared to variability during transmission over one link, it is important for telephone ASR to train and test using different links [21]. One text-independent study, using ten speakers and cepstral features, found error rate increasing from 17 percent to 56 percent when test data were drawn from a different link than the training data [35]. Subtracting the mean over a recording session from each feature to yield channel-invariant features reduced this degradation, but also decreased net performance because such a normalization eliminated useful speaker information. Multichannel error rate was reduced to 32 percent by incorporating a Gaussian model to account for transmission effects. More complex recognition methods have been suggested for high-noise radio channels, but they too still yield high error rates [36]. Since an error rate of one-third is unacceptable for virtually all applications, significant problems remain for text-independent telephone ASR.

In addition to handling telephone speech, ASR has recently been attempted on digitally coded speech, which is becoming popular for voice transmission. One text-independent recognition study tested several types of coded 4-kHz speech using 20 speakers and LPC feature vectors consisting of 10 LPC reflection coefficients and 10 cepstral coefficients [37]. Compared to a 95 percent accuracy for uncoded speech, performance generally fell with bit rate, with 2.4 kbit/s LPC-coded speech yielding 80 percent. These results are comparable with that of human recognition of coded voices [38]. Waveform coding gave relatively poor results because quantization noise degradation led to poor LPC modeling. A recognition method using features based on parameters other than LPC might do better on waveform-coded speech. For example, ASV performance using only F0 and gain features appears to be unaffected by either telephone or coding (LPC or ADPCM) distortions [39], which suggests the utility of prosodics as robust recognition features.

SPEAKER RECOGNITION BY HUMANS

In human speaker recognition, people can reliably identify voices with which they are familiar. About 2–3 s of speech suffices to identify a voice, although performance decreases for unfamiliar voices. Speaker recognition is one area of artificial intelligence where machines can exceed human performance: using short test utterances and a large number of speakers, ASR accuracy often exceeds that of humans. This is especially true for unfamiliar speakers, where the "training time" for humans to learn a new voice well is very long compared to that for machines. Constraints on how many unfamiliar voices a subject can retain in short-term memory usually limit studies of human speaker recognition to about 5–10 speakers. Such small speaker sets lead to large statistical variation from one set to another, given that distinctiveness and degree of familiarity of voices often vary widely across speakers. While perceptually rated scales of "distinctiveness" (i.e., whether a voice stands out in a crowd) appear to have little correlation with ability to recognize a voice, recognition performance using both uncoded and LPC-coded speech increases dramatically with more familiarity between listener and speaker [38, 40].

A review of human speaker recognition in [41] notes that many studies of 8–10 speakers (work colleagues of the listening subjects) yield more than 97 percent accuracy, if a sentence or more of the test speech is available. Performance falls to about 54 percent (but still significantly above chance levels) when duration is short (e.g., less than 1 s) and/or distorted (e.g., severely high- or lowpass-filtered). One study of 29 familiar speakers had 31, 66, and 83 percent recognition with one word, one sentence, and 30 s of speech, respectively [42]. Performance also falls significantly if training and test utterances are processed through different transmission systems [39]. A study using voices of 45 famous people in 2-s test utterances found only 27 percent recognition in an open-choice test, but 70 percent correct if listeners could select from six choices [41]; if the utterances were increased to 4 s, but played *backwards* (which distorts timing and articulatory cues), 57.5 percent accuracy resulted. Widely varying performance on the backwards task suggested that cues to voice recognition vary from voice to voice, and that voice patterns may consist of a set of acoustic cues from which listeners select a subset to use in identifying a voice.

Recognition often falls dramatically when speakers attempt to disguise their voices [40] (e.g., 59–81 percent accuracy depending on the disguise vs. 92 percent for normal voices in one study [43]). This is reflected in ASR, where accuracy decreases when mimics act as impostors [44]. Humans appear to handle mimics better than machines do, easily perceiving when a voice is being mimicked (e.g., 90 percent accuracy [45]). If the target voice is known to the listener, he often *associates* the mimic voice with it, but does not *confuse* them. Certain voices are more easily mimicked than others, which lends further evidence to the theory that different acoustic cues are used to distinguish different voices [41].

Voiceprint identification

The ability to identify speakers via *voiceprints* (spectrograms of their voices) has been of particular interest in forensic work. Despite some evidence to the contrary [1], most researchers feel that spectrogram reading has not been demonstrated to reliably identify speakers [46]. Experts seem to be able to attain a certain degree of ability to match reference spectrograms to test ones by the same speaker, but performance often degrades substantially if speakers disguise their voices. For example, one study of 15 speakers used spectrograms of nine different monosyllabic words excerpted from different sentences. With normal voices, experts achieved 57 percent accuracy; when speakers spoke very slowly, accuracy fell 14 percent; when they used free disguise, recognition was only 22 percent [47]. Furthermore, certain speakers were considerably more difficult to identify than others.

CONCLUSION

It is useful to examine the lack of commercial success for ASR compared to that for speech recognition. Both speech and speaker recognition analyze speech signals to extract F0 and spectral parameters such as LPC or cepstral coefficients. Furthermore, both often employ similar template matching methods, the same distance measures, and similar decision procedures. Speech and speaker recognition, however, have different objectives: selecting which of M words was spoken vs. which of N speakers spoke. Speech analysis techniques have primarily been developed for phonemic analysis, e.g., to preserve phonemic content during speech coding or to aid phoneme identification in speech recognition. Our understanding of how listeners exploit spectral cues to identify human sounds exceeds our knowledge of how we distinguish speakers. For text-dependent ASR, using template-matching methods borrowed directly from speech recognition yields good results in limited tests, but performance decreases under adverse conditions that might be found in practical applications. For example, telephone distortions, uncooperative speakers, and speaker variability over time often lead to accuracy levels unacceptable for many applications.

Studies that have suggested detailed phonetic features useful for ASR, such as specific formants in certain phones, may eventually result in improved ASR, but the problems of "phone-spotting" have led to few studies exploiting such specific features. As speech recognition techniques improve, ASR may adopt them to yield parallel improvements. Even more so than in speech recognition, statistical methods have dominated ASR research. Only techniques using template or VQ matching or distances involving long-term averages of speech features have been tried in recent years. High recognition accuracy comes from gathering a large number of speech features and evaluating their utility in ASR via the weighting matrix in a Mahalanobis distance. These methods yield no insight into the human speaker recognition process, but serve ASR objectives (to a certain extent).

In the near future, practical applications are most likely for ASV using text-dependent techniques. The more difficult task of ASI is further away from commercial use because of the tendency of increasing error probability as population size increases. Higher ASI accuracy is needed for large-population recognition, whereas current techniques appear to yield sufficient accuracy for ASV applications. For further reading, there are several good reviews of the ASR field [2, 3, 24, 48, 49, 50].

REFERENCES

[1] Tosi, O., *Voice Identification: Theory and Legal Applications,* Baltimore, MD, Univ. Park Press, 1979.

[2] Rosenberg, A., "Automatic speaker verification," *Proc. IEEE,* 64, pp. 475–487, 1976.

[3] Atal, B., "Automatic recognition of speakers from their voices," *Proc. IEEE,* **64,** pp. 460–475, 1976.

[4] Flanagan, J., Schroeder, M., Atal, B., Crochiere, R., Jayant, N., and Tribolet, J., "Speech coding," *IEEE Trans. Comm.,* **COM-27,** pp. 710–736, 1979.

[5] Rabiner, L., and Schafer, R., *Digital Processing of Speech Signals,* Englewood Cliffs, NJ, Prentice Hall, 1979.

[6] Li, K., and Hughes, G., "Talker differences as they appear in correlation matrices of continuous speech spectra," *J. Acoust. Soc. Am.,* **55,** pp. 833–837, 1974.

[7] Itakura, F., "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. ASSP,* **ASSP-23,** 67–72, 1975.

[8] Sakoe, H., and Chiba, S., (1978) "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. ASSP,* **ASSP-26,** pp. 43–49, 1978.

[9] Ney, H., "The use of a one-stage dynamic programming algorithm for connected word recognition," *IEEE Trans. ASSP,* **ASSP-32,** pp. 263–271, 1984.

[10] Rabiner, L., and Levinson, S., "Isolated and connected word recognition — theory and selected applications," *IEEE Trans. Comm.,* **COM-29,** pp. 621–659, 1981.

[11] Das, S., and Mohn, W., "A scheme for speech processing in automatic speaker verification," *IEEE Trans. Audio and Electroac.,* **AU-19,** pp. 32–43, 1971.

[12] Lummis, R., "Speaker verification by computer using speech intensity for temporal registration, *IEEE Trans. Audio and Electroac.,* **AU-21,** pp. 80–89, 1973.

[13] Furui, S., "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. ASSP,* **ASSP-29,** pp. 254–272, 1981.

[14] Hollien, H., and Majewski, W., "Speaker identification by long-term spectra under normal and distorted speech conditions," *J. Acoust. Soc. Am.,* **62,** pp. 975–980, 1977.

[15] Furui, S., "Comparison of speaker recognition methods using statistical features and dynamic features," *IEEE Trans. ASSP,* **ASSP-29,** pp. 342–350, 1981.

[16] Gray, R., "Vector quantization," *IEEE ASSP Magazine* **1,** pp. 4–29, 1984.

[17] Abut, H., Gray, R., and Rebolledo, G., "Vector quantization of speech," *IEEE Trans. ASSP,* **ASSP-30,** pp. 423–435, 1982.

[18] Li, K., and Wrench, Jr., E., "An approach to text-independent speaker recognition with short utterances," *Proc. IEEE Intern. Conf. ASSP,* pp. 555–558, 1983.

[19] Soong, F., Rosenberg, A., Rabiner, L., and Juang, B., "A vector quantization approach to speaker recognition," *Proc. IEEE Intern. Conf. ASSP,* pp. 387–390, 1985.

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

[21] Hunt, M., "Further experiments in text-independent speaker recognition over communications channels," *Proc. IEEE Intern. Conf. ASSP,* pp. 563–566, 1983.

[22] Sambur, M., "Speaker recognition using orthogonal linear prediction," *IEEE Trans. ASSP,* **ASSP-24,** pp. 283–289, 1976.

[23] Bogner, R., "On talker verification via orthogonal parameters," *IEEE Trans. ASSP,* **ASSP-29,** pp. 1–12, 1981.

[24] Bricker, P., and Pruzansky, S., "Speaker recognition," in *Contemporary Issues in Experimental Phonetics,* N. Lass, ed. (Academic Press: NY), pp. 295–326, 1976.

[25] Wolf, J., "Efficient acoustic parameters for speaker recognition," *J. Acoust. Soc. Am.,* **51,** pp. 2044–2056, 1972.

[26] Matsumoto, H., Hiki, S., Sone, T., and Nimura, T., "Multidimensional representation of personal quality of vowels and its acoustical correlates," *IEEE Trans. Audio and Electroac.,* **AU-21,** pp. 428–436, 1973.

[27] Walden, B., Montgomery, A., Gibeily, G., Prosek, R., and Schwartz, D., "Correlates of psychological dimensions in talker similarity," *J. Speech Hear. Res.* **21,** pp. 265–275, 1978.

[28] Sambur, M., "Selection of acoustic features for speaker identification," *IEEE Trans. ASSP,* **ASSP-23,** pp. 176–182, 1975.

[29] Goldstein, U., "Speaker-identifying features based on formant tracks," *J. Acoust. Soc. Am.,* **59,** pp. 176–182, 1976.

[30] Cheung, R., and Eisenstein, B., "Feature selection via dynamic programming for text-independent speaker identification," *IEEE Trans. ASSP,* **ASSP-26,** pp. 397–403, 1978.

[31] Atal, B., "Automatic speaker recognition based on pitch contours," *J. Acoust. Soc. Am.,* **52,** pp. 1687–1697, 1972.

[32] Johnson, C., Hollien, H., and Hicks, J., "Speaker identification utilizing selected temporal speech features," *J. Phonetics,* **12,** pp. 319–326, 1984.

[33] Sekey, A., "Building a model for large vocabulary isolated word recognition," *Speech Tech.,* pp. 71–81, Aug./Sept. 1984.

[34] Kaneko, T., and Dixon, N. R., "A hierarchical decision

approach to large-vocabulary discrete utterance recognition," *IEEE Trans. ASSP,* **ASSP-31,** pp. 1061–1072, 1983.

[35] Gish, H., Karnofsky, K., Krasner, M., Roucos, S., Schwartz, R., and Wolf, J., "Investigation of text-independent speaker identification over telephone channels," *Proc. IEEE Intern. Conf. ASSP,* pp. 379–382, 1985.

[36] Krasner, M., Wolf, J., Karnofsky, K., Schwartz, R., Roucos, S., and Gish, H., "Investigation of text-independent speaker identification techniques under conditions of variable data," *Proc. IEEE Intern. Conf. ASSP,* 18B.5., pp. 1–4, 1984.

[37] Everett, S., "Automatic speaker recognition using vocoded speech," *Proc. IEEE Intern. Conf. ASSP,* pp. 383–386, 1985.

[38] Schmidt-Neilsen, A., and Stern, K., "Identification of known voices as a function of familiarity and narrow-band coding," *J. Acoust. Soc. Am.,* 77, pp. 658–663, 1985.

[39] McGonegal, C., Rosenberg, A., and Rabiner, L., "The effects of several transmission systems on an automatic speaker verification system," *Bell Sys. Tech. J.,* **58,** pp. 2071–2087, 1979.

[40] Hollien, H., Majewski, W., and Doherty, E.T, "Perceptual identification of voices under normal, stress and disguise speaking conditions," *J. Phonetics,* 10, pp. 139–148, 1982.

[41] Van Lancker, D., Kreiman, J., and Emmorey, K., "Familiar voice recognition: patterns and parameters—Part I: Recognition of backward voices," *J. Phonetics,* **13,** pp. 19–38, 1985.

[42] Ladefoged, P., and Ladefoged, J., "The ability of listeners to identify voices," *UCLA Working Papers in Phonetics,* **49,** pp. 43–51, 1980.

[43] Reich, A., and Duke, J., "Effects of selected vocal disguises upon speaker identification by listening," *J. Acoust. Soc. Am.,* 66, pp. 1023–1028, 1979.

[44] Rosenberg, A., and Sambur, M., "New techniques for automatic speaker verification," *IEEE Trans. ASSP,* **ASSP-23,** pp. 169–176, 1975.

[45] Reich, A., "Detecting the presence of vocal disguise in the male voice," *J. Acoust. Soc. Am.,* **69,** pp. 1458–1461, 1981.

[46] Bolt, R., Cooper, F., David, E., Jr., Denes, P., Pickett, J., and Stevens, K., "Speaker identification by speech spectrograms: some further observations, *J. Acoust. Soc. Am.,* **54,** pp. 531–537, 1973.

[47] Reich, A., Moll, K., and Curtis, J., "Effects of selected vocal disguises upon spectrographic speaker identification," *J. Acoust. Soc. Am.,* **60,** pp. 919–925, 1976.

[48] Hecker, M., "Speaker recognition: an interpretive survey of the literature," *ASHA Monographs,* **16,** 1971.

[49] Corsi, P., "Speaker recognition: A survey," in *Automatic Speech Analysis and Recognition,* J.-P. Haton, (Ed.) Dordrecht, Holland, D. Reidel, pp. 277–308, 1982.

[50] Doddington, G., "Speaker recognition—identifying people by their voices," *Proc. IEEE,* **73,** pp. 1651–1664, 1985.

[51] Davis, S., and Mermelstein, P., "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. ASSP,* **ASSP-28,** pp. 357–366, 1980.

Douglas O'Shaughnessy received the BSc and MSc degrees in 1972 and the PhD degree in 1976 from the Massachusetts Institute of Technology, all in electrical engineering and computer science. Since 1977, he has been at INRS-Telecommunications, University of Quebec, where he is now an associate professor, and at McGill University, as an auxiliary professor teaching communications. He has worked on English and French synthesis-by-rule and modelling of intonation. His main interests lie in speech synthesis, coding, and recognition. He is the author of a text, 'Speech Communications,' to be published by Addison-Wesley in early 1987.