Feature Selection via Dynamic Programming for Text-Independent Speaker Identification

RONALD S. CHEUNG, MEMBER, IEEE, AND BRUCE A. EISENSTEIN, SENIOR MEMBER, IEEE

Abstract—**Dynamic programming is applied to the selection of feature subsets in text-independent speaker identification. Each feature is long-term averaged in order to reduce its variability to text information. The resulting subset of features shows a lower average identification error in comparison to that of the "knock-out" strategy, the cepstral coefficients, and the PARCOR coefficients.**

I. Introduction

THE ULTIMATE goal of all speaker recognition studies is to devise an automatic time-independent, unbiased system that can duplicate the human ability to perform fast, accurate, and text-independent recognition of speakers. Though this ability seems common and natural to us, the use of machines to do the same task is nontrivial. The general approach is to extract some acoustic attributes from one's speech and compare them with a reference set previously stored in the machine's library. If there is a close resemblance between the test and reference features, the speaker is said to be recognized. In text-dependent speaker recognition where the test and reference features are obtained from the same text material, meaningful comparisons between the two sets of attributes can be made after aligning the speech utterances using some time normalization or registration schemes [1], [2]. However, the same situation does not hold for the text-independent case where the test and reference text bear no linguistic relationship to each other. It is because these acoustic attributes derived from speech not only signify the inter-speaker variations, but also are functions of the speech text. Hence, the success of a text-independent automatic speaker recognition system depends on the extraction of a set of acoustic properties that can characterize each speaker independent of the speech text.

By time averaging acoustic attributes of speech over different speech text, it was shown recently that some features (e.g., pitch, gain, and reflection coefficients) exhibited large inter-speaker variability regardless of the speech context [3]. In N-dimensional space, the feature set obtained from long-term averaging clustered around the mean value which characterized the speaker. Moreover, the variance of the cluster was also shown to decrease with a longer averaging period yielding better separability of speakers. Hence, long-term averaging of acoustic features seems suitable for text-independent speaker recognition. Since not all acoustic features of speech are useful in distinguishing speakers, a selection procedure has to be formulated to retain only those that yield the best results.

To select a subset of k best features among the entire set N, the optimal method is to consider all combinations of N objects taken k at a time, $\binom{N}{k}$, and exhaustively search for the best one. Unfortunately, implementation of such a search scheme requires an enormous amount of computation, especially for large N and intermediate values of k. In practice, suboptimal schemes, such as the search without replacement [4], sometimes known as the "knock-out" strategy [5], are more often used. These algorithms begin with evaluating the N given features one at a time and "knocks out" the most effective one. Then this feature is coupled one at a time with the remaining $(N-1)$ attributes in the set and these feature pairs are evaluated resulting in the knock-out of the best pair of features. This process is iterated until a subset of k features is obtained. The above scheme, though computationally efficient, $[k \cdot (N-(k-1)/2)$ searches; see Appendix A] suffers the inherent disadvantage that the resulting subset which contains the best individually selected properties is not necessarily the optimal subset of features.

Recently, dynamic programming has been applied to feature selection in pattern recognition. Nelson *et al.* [6] employed a multistage decision process to choose features that had acquisition costs associated with them. His method was to pick those features that maximized the total stage return subject to a predefined cost constraint. However, in most feature selection applications, the acquisition cost of features is either unknown or meaningless and such a scheme cannot be utilized. Chang [7] suggested two dynamic programming procedures to perform feature subset selection. His techniques were similar to the knock-out strategy but without the pitfall of the latter one. Instead of a complete "knock-out" of the optimum features from the rest of the selection process, Chang's algorithms also considered subsets consisting of one optimum feature coupled with some of the unselected ones. This paper discusses a similar dynamic programming procedure and the application of it to select features in text-independent speaker identification. The "goodness" of these features is evaluated using a statistical distance measure known as the divergence [8], [20]. The chosen feature set is then employed in conjunction with a linear classifier for identifying speakers. The results obtained are shown to favorably compare to those obtained from the use of the knock-out strategy, the cepstral coefficients, and the PARCOR coefficients [9].

Manuscript received September 15, 1977; revised April 28, 1978.
R. S. Cheung is with GTE Sylvania Inc., Needham Heights, MA 02194.
B. A. Eisenstein is with Drexel University, Philadelphia, PA 19104.

0096-3518/78/1000-0397$00.75

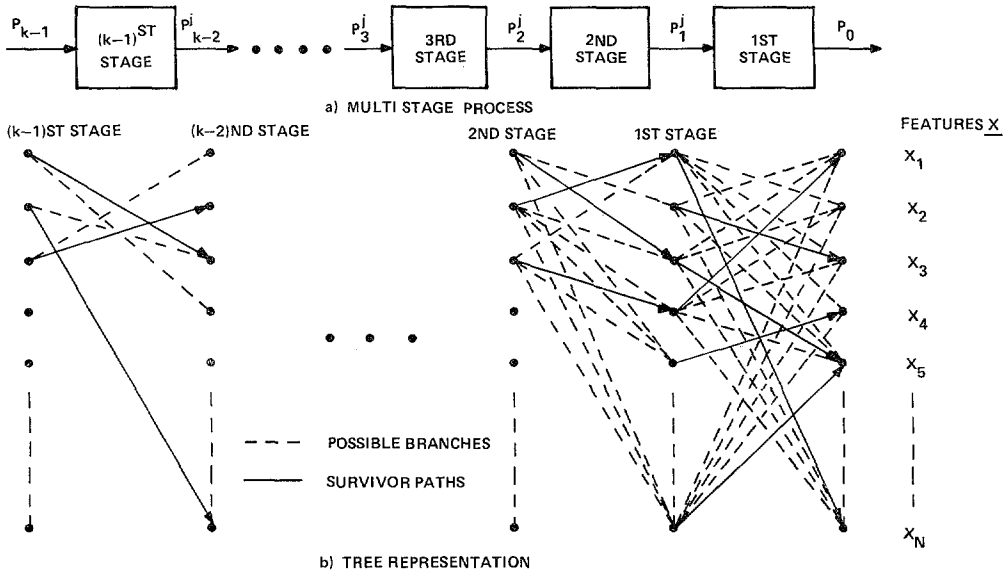

Fig. 1. Feature subset selection using dynamic programming.

II. The Dynamic Programming Approach

Dynamic programming is a multistage optimization technique as shown in Fig. 1(a) that makes use of the Principle of Optimality [10] which states: whatever the initial state and decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. When applied to the selection of features, the principle in conjunction with a functional equation permits the choice of attributes that have the maximum effectiveness.

Let $X = (x_1, x_2, \cdots, x_N)$ be the set of N available features[1], $\boldsymbol{P}^i_n = (p^j_1, p^j_2, \cdots, p^j_n)$ be one of the N possible subsets selected after n stages and p^j_n represents a feature in X. For every x_j at the nth stage, the subset $\boldsymbol{P}^j_n$ is picked such that

$$\lambda_n(\boldsymbol{P}^j_n) = \max D_n(\boldsymbol{P}^j_{n-1}, x_j), \qquad i = 1, 2, \cdots, N, \quad x_j \notin \boldsymbol{P}^i_{n-1} \tag{1}$$

where $(\boldsymbol{P}^i_{n-1}, x_j)$ represents a feature subset formed by augmenting $\boldsymbol{P}^i_{n-1}$ with x_j as follows:

$$(\boldsymbol{P}^i_{n-1}, x_j) = (p^i_1, p^i_2, \cdots, p^i_{n-1}, x_j). \tag{2}$$

D_n is defined as a feature effectiveness measure and λ_n is the maximum effectiveness measure over a collection of subsets as defined in (1). Since there are N such x_j's at each stage, N subsets $\boldsymbol{P}^i_n$ are generated. The optimum subset $\boldsymbol{P}_n$ at the end of n stages can be obtained from the $\boldsymbol{P}^j_n$'s as follows:

$$\lambda_n(\boldsymbol{P}_n) = \max \lambda_n(\boldsymbol{P}^j_n), \qquad j = 1, 2, \cdots, N. \tag{3}$$

A block diagram of the above dynamic programming algorithm is shown in Fig. 2. In the beginning, $\boldsymbol{P}^i_0$ is initialized to be x_i as shown:

$$\boldsymbol{P}^i_0 = (x_i) \qquad \text{for } i = 1, 2, \cdots, N. \tag{4}$$

Then for each x_j in stage 1, $(N-1)$ subsets can be formed by pairing x_j with $\boldsymbol{P}^i_0$ for all $x_j \notin \boldsymbol{P}^i_0$ as shown in (2). The effectiveness D_1 of these subsets is computed and compared as shown in (1) resulting in $\boldsymbol{P}^j_1$. The same procedures are iterated for all x_j resulting in N such subsets $\boldsymbol{P}^j_1$, $j = 1, \cdots, N$. The process is repeated for stages $n = 2, 3, \cdots, k-1$ and at the end of the $(k-1)$st stage, the optimal subset $\boldsymbol{P}_{k-1}$ of k features can be obtained from the $\boldsymbol{P}^j_{k-1}$ for $j = 1, 2, \cdots, N$ as shown in (3).

An alternative view of the dynamic programming procedure is to use a tree search method as shown in Fig. 1(b). In this representation, the features x_i are depicted by the nodes of the tree. Subsets generated as shown in (2) can be interpreted as paths or branches joining the nodes of subsequent stages. Hence, for each input node, the algorithm calls for a sequential evaluation of all branches connecting the input to nodes of other stages. The most effective branch or survivor path, for the given initial node, results. Duplicating the same steps for the rest of the nodes in the initial stage yields survivor paths starting from each of the N nodes. The feature selection process reduces to the tracing of survivor paths through the stages. Proceeding in the above manner, N survivor paths connecting k nodes are shown at the end of the $(k-1)$st stage. Then the optimal path for the $(k-1)$ stages corresponds to the most effective one among the N survivors at the $(k-1)$st stage.

In all dynamic programming algorithms, optimal results can only be achieved if the k-stage problem as given in Fig. 1(a) can be decomposed into k subproblems. Moreover, it can be shown that if the effectiveness criterion, D_n, is a monotonic, nondecreasing function of n shown as

$$D_n(\boldsymbol{P}^j_n) \geq D_{n-1}(\boldsymbol{P}^j_{n-1}) \qquad \text{for any } p^j_n, \tag{5}$$

and D_n can be separated into two parts, one corresponding to the history of the process up to the $(n-1)$st stage and the other corresponding to the behavior of the process at the nth stage as

$$D_n(\boldsymbol{P}^j_n) = f(D_n(p^j_n) D_{n-1}(\boldsymbol{P}^j_{n-1})), \tag{6}$$

where f is a real-value function relating $D_n(p^j_n)$ and $D_{n-1}(\boldsymbol{P}^j_{n-1})$ to $D_n(\boldsymbol{P}^j_n)$, decomposition of the multistage process is guaranteed [19]. With the above feature effectiveness measure, the sub-

[1]The quantities shown in boldface italic type are vectors.

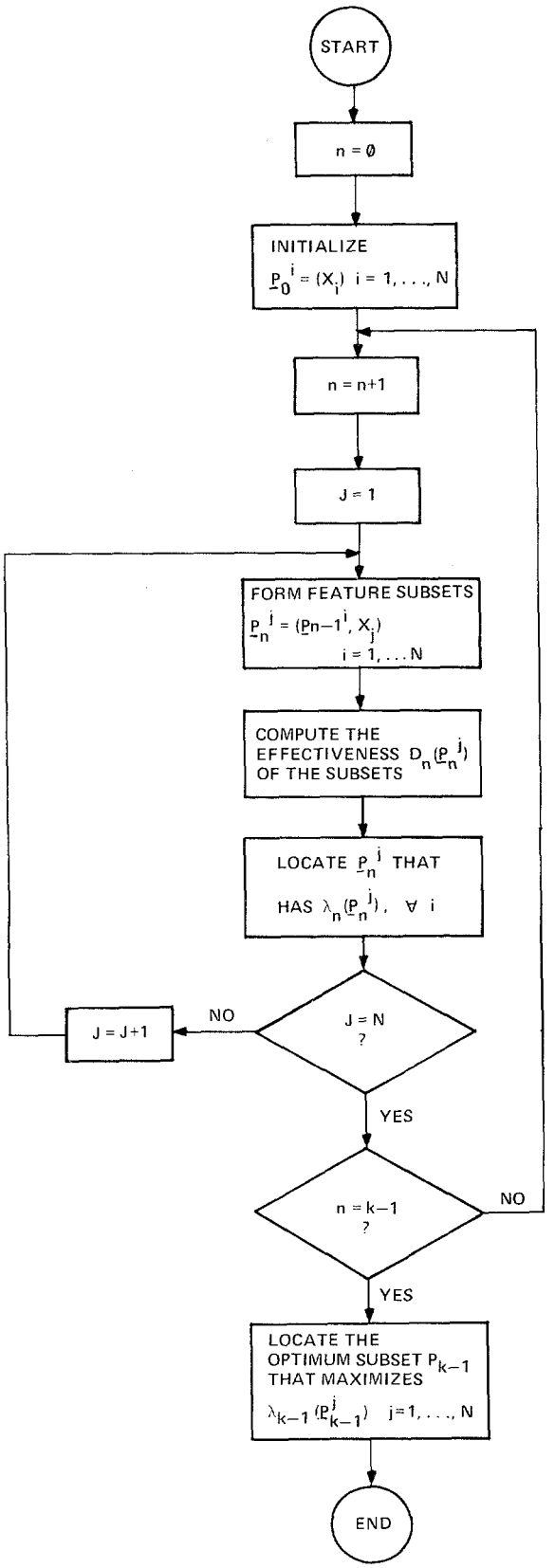

Fig. 2. Block diagram of feature subset selection using dynamic programming.

set selected by the dynamic programming procedure may not necessarily contain features that are individually the best as with the knock-out strategy, but will instead result in the most effective attribute subset of size k.

The number of searches required in the dynamic programming method is given by

$$N_{DP} = N\sum_{i=1}^{k-1}(N-i) = N(k-1)\left(N-\frac{k}{2}\right) \tag{7}$$

which is substantially less than the exhaustive search for large N and intermediate values of k. The improvement ratio R defined as N_{ES}/N_{DP} is shown in Appendix B to be bounded by

$$R < \frac{1}{k!}\cdot\frac{N^{k-2}}{(k-1)} \qquad \text{for } N > k \tag{8}$$

where N_{ES} is the total number required by the exhaustive search method.

III. The Feature Effectiveness Criterion

In automatic speaker identification, the features used are measured from the talker's speech, and each measurement of these attributes can be represented by a point in the N-dimensional feature space. Repeating the measurement process, a cluster of points are generated in the space and they are distributed according to some N-dimensional probability density function (pdf) which characterizes the variance in the speaker's voice. So, the effectiveness of such measurements depends on how well the individual speaker's pdf differs from the others with respect to these features.

Before deciding on the merits of the features, the classifier used in distinguishing the speaker pdf's has to be established. The linear classifier is by far the most common one and it employs a specified distance metric to determine the identity of the unknown speaker by selecting the speaker with the shortest distance between the reference and the test points. Moreover, if the underlying pdf's of the speakers are Gaussian distributed with identical covariance matrices, a not unreasonable assumption [5], the distance metric is simplified to a quadratic distance measure [11].

After establishing the decision logic for the classifier, the features can be evaluated. For identifying unknown speakers, a meaningful effectiveness criterion is the error performance of the features over some test data. The set of features which commits more errors in identifying a group of talkers is said to be less effective. Such an effectiveness measure can be determined experimentally by employing the attributes in the identification experiment and tallying up the mistakes made. However, implementation of the criterion requires a tremendous amount of computation, especially for large data sets. An alternative is to exploit the statistical properties of the features and derive the probability of error from the talker's pdf. The scheme involves the estimation of the multidimensional distribution from a set of labeled training samples and if the distribution is Gaussian, the probability of error is obtained by integrating over the error range. As a result, the calculation of the probability of error is difficult and even in the case of a discrete distribution, the computation of the error criterion is tedious for large feature dimensions [13].

In light of the computational limitations of the probability of error, various statistical measures are utilized which are relatively simple if the pdf is Gaussian. One example is the

divergence [8], [20] criterion which for the 2-speaker identification is defined as

$$D_{ij} = E\{\log(L(X))|i\} - E\{\log(L(X))|j\} \tag{9}$$

where $E\{\cdot\}$ denotes the expected value, X is the feature vector, and $L(X)$ is the likelihood function given by

$$L(X) = \frac{p_i(X)}{p_j(X)} \tag{10}$$

where $p_i(X)$, $p_j(X)$ are the pdf's of speakers i and j. If $p_i(X)$, $p_j(X)$ are Gaussian, with mean values $\boldsymbol{\mu}_i$, $\boldsymbol{\mu}_j$ and the same within-speaker covariance matrix W, D_{ij} is shown as [8]

$$D_{ij} = (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T W^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \tag{11}$$

where $(\cdot)^T$ denotes the transpose of the matrix and D_{ij} can be interpreted as a distance measure between p_i and p_j. The divergence present in (11) can be shown to increase monotonically with the probability of correct recognition [8]. Applying the above criterion to feature selection, the idea is to choose the subset of features that has the maximum divergence.

For the m-speaker identification case, the average value of the divergence defined as follows can be utilized [14]:

$$D = \langle D_{ij} \rangle_{ij} \quad \begin{matrix} i = 1, 2, \cdots, m \\ j = 1, 2, \cdots, m \end{matrix} \tag{12}$$

where $\langle\cdot\rangle_{ij}$ denotes averaging over indices i and j. Assuming the within-speaker covariance matrices W for all the speakers are the same, substituting (11) into (12) results in:

$$\begin{aligned} D &= \langle (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T W^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \rangle_{ij} \quad \begin{matrix} i = 1, 2, \cdots, m \\ j = 1, 2, \cdots, m \end{matrix} \\ &= \operatorname{tr} \langle W^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \cdot (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \rangle_{ij} \\ &= \operatorname{tr} (W^{-1} \cdot \langle (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) \cdot (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \rangle_{ij}) \\ &= \operatorname{tr} (W^{-1} B) \end{aligned} \tag{13}$$

where tr represents the trace of a matrix and B, defined as

$$B = \langle (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^T \rangle_{ij}, \tag{14}$$

is known as the between-speaker covariance matrix [14].

In practice, the covariance matrices W and B, for a given feature subset, are estimated from labeled samples in the training data set. If W_n and B_n represent the within-speaker and between-speaker covariance matrices for the n-feature subset $\boldsymbol{P}_n^j = (p_1^j, p_2^j, \cdots, p_n^j)$ where $p_i^j \in X$, $i = 1, \cdots, n$, the divergence D_n given by

$$D_n(\boldsymbol{P}_n^j) = \operatorname{tr} (W_n^{-1} \cdot B_n) \tag{15}$$

does not decrease with increasing feature number n [20], thus satisfying the monotonicity condition as stated in (5). For uncorrelated features, D_n also fulfills the separability condition since

$$D_n(\boldsymbol{P}_n^j) = D_{n-1}(\boldsymbol{P}_{n-1}^j) + \left\langle \frac{(m_{in} - m_{jn})^2}{\sigma_n^2} \right\rangle_{ij} \quad \begin{matrix} i = 1, 2, \cdots, m \\ j = 1, 2, \cdots, m \end{matrix} \tag{16}$$

where $D_{n-1}(\boldsymbol{P}_{n-1}^j)$ denotes the divergence for the feature subset $\boldsymbol{P}_{n-1}^j = (p_1^j, p_2^j, \cdots, p_{n-1}^j)$, m_{in} represents the mean value of the nth feature, p_n^j, for the ith speaker, and σ_n^2 is the variance of p_n^j. Unfortunately, for correlated features, D_n does not satisfy the separability condition as given in (6). Therefore, in selecting statistically dependent attributes using dynamic programming, the divergence measure does not guarantee the decomposition of the multistage process and consequently, the chosen subset may not necessarily be optimum.

IV. The Speaker Identification Experiment

The recordings of ten male speakers reading the first four lists of the Harvard PB sentences [15] were used as the data base for the text-independent speaker identification experiment. These sentences, other than being phonetically balanced, bear no linguistic relationship to each other. Lists 1 and 2, which contained ten sentences each, were used as a training set while lists 3 and 4 were used as a test set. The recordings after low-pass filtering were digitized at 6400 Hz and then fed into the feature extractor as shown in Fig. 3.

A set of 32 acoustic attributes was determined from the input speech, namely, the pitch value, log energy, ten PARCOR coefficients, ten cepstral coefficients, normalized absolute prediction error energy, and nine normalized autocorrelation coefficients. The pitch value M was determined using the average magnitude difference function (AMDF) [16]. The log energy in decibels was computed as follows:

$$E = 10 \cdot \log_{10} \frac{1}{L} \left(\sum_{i=1}^{L} S_i^2 \right) \tag{17}$$

where S_i denoted the input speech and L, the frame length, was 128 samples. The linear prediction parameters were obtained every 20 ms as a result of a tenth-order linear predictive analysis on the Hamming-windowed input waveform using the autocorrelation approach [18]. The normalized autocorrelation coefficients R_i were calculated as the following:

$$R_i = \frac{1}{R_0} \cdot \sum_{t=0}^{L-i-1} S_t S_{t+i} \qquad i = 2, 3, \cdots, 10 \tag{18}$$

where R_0 is given by

$$R_0 = \sum_{t=0}^{L-1} S_t^2. \tag{19}$$

The PARCOR coefficients, K_i, and the normalized absolute prediction error energy, $|e|$, were determined using the Levinson recursion [18]. The cepstral coefficients, C_i, for the all-pole model were derived from the predictive coefficients [17].

In order for the above feature set to be applicable to text-independent speaker identification, each feature was averaged over some input text as shown

$$\langle x_i \rangle_j = \frac{1}{L_v} \cdot \sum_{j=1}^{L_v} x_{ij} \tag{20}$$

where x_{ij} was the ith feature obtained from the jth speech

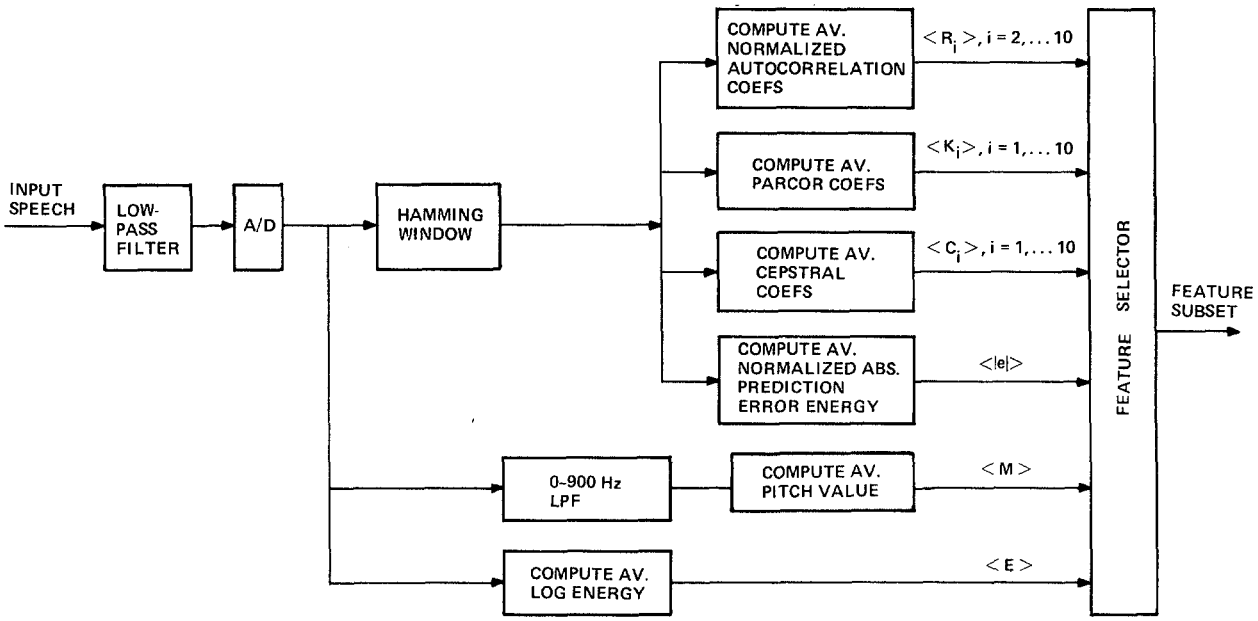

Fig. 3. Extraction of features from speech in text-independent speaker identification.

TABLE I
FEATURES SELECTED AT EACH STAGE FOR DYNAMIC PROGRAMMING AND THE KNOCK-OUT STRATEGY ($L_v = 10$)

	STAGE NUMBER									
	0	1	2	3	4	5	6	7	8	9
DYNAMIC PROGRAMMING		$\langle \lvert e \rvert \rangle \langle K_7 \rangle$	$\langle K_7 \rangle \langle M \rangle$ $\langle C_1 \rangle$	$\langle K_7 \rangle \langle M \rangle$ $\langle C_1 \rangle \langle K_9 \rangle$	$\langle K_7 \rangle \langle C_1 \rangle$ $\langle M \rangle \langle R_{10} \rangle$ $\langle K_9 \rangle$	$\langle K_7 \rangle \langle M \rangle$ $\langle C_1 \rangle \langle R_{10} \rangle$ $\langle K_9 \rangle \langle C_3 \rangle$	$\langle K_7 \rangle \langle M \rangle$ $\langle C_1 \rangle \langle K_9 \rangle$ $\langle C_5 \rangle \langle R_3 \rangle$ $\langle E \rangle$	$\langle K_7 \rangle \langle M \rangle$ $\langle C_1 \rangle \langle K_9 \rangle$ $\langle R_{10} \rangle \langle E \rangle$ $\langle R_4 \rangle \langle C_2 \rangle$	$\langle K_7 \rangle \langle M \rangle$ $\langle C_1 \rangle \langle K_9 \rangle$ $\langle R_{10} \rangle \langle E \rangle$ $\langle C_2 \rangle \langle C_4 \rangle$ $\langle C_6 \rangle$	$\langle K_7 \rangle \langle M \rangle$ $\langle C_1 \rangle \langle K_9 \rangle$ $\langle R_{10} \rangle \langle E \rangle$ $\langle C_2 \rangle \langle C_4 \rangle$ $\langle C_6 \rangle \langle C_3 \rangle$
KNOCK-OUT STRATEGY	$\langle K_7 \rangle$	$\langle K_7 \rangle \langle \lvert e \rvert \rangle$	$\langle K_7 \rangle \langle \lvert e \rvert \rangle$ $\langle M \rangle$	$\langle K_7 \rangle \langle \lvert e \rvert \rangle$ $\langle M \rangle \langle K_9 \rangle$	$\langle K_7 \rangle \langle \lvert e \rvert \rangle$ $\langle M \rangle \langle K_9 \rangle$ $\langle C_1 \rangle$	$\langle K_7 \rangle \langle \lvert e \rvert \rangle$ $\langle M \rangle \langle K_9 \rangle$ $\langle C_1 \rangle \langle C_3 \rangle$	$\langle K_7 \rangle \langle \lvert e \rvert \rangle$ $\langle M \rangle \langle K_9 \rangle$ $\langle C_1 \rangle \langle C_3 \rangle$ $\langle R_{10} \rangle$	$\langle K_7 \rangle \langle \lvert e \rvert \rangle$ $\langle M \rangle \langle K_9 \rangle$ $\langle C_1 \rangle \langle C_3 \rangle$ $\langle R_{10} \rangle \langle E \rangle$	$\langle K_7 \rangle \langle \lvert e \rvert \rangle$ $\langle M \rangle \langle K_9 \rangle$ $\langle C_1 \rangle \langle C_3 \rangle$ $\langle R_{10} \rangle \langle E \rangle$ $\langle R_3 \rangle$	$\langle K_7 \rangle \langle \lvert e \rvert \rangle$ $\langle M \rangle \langle K_9 \rangle$ $\langle C_1 \rangle \langle C_3 \rangle$ $\langle R_{10} \rangle \langle E \rangle$ $\langle R_3 \rangle \langle R_7 \rangle$

$\langle K_i \rangle$ = i^{TH} AV. PARCOR COEF.
$\langle R_i \rangle$ = i^{TH} AV. NORMALIZED AUTOCORRELATION COEF.
$\langle C_i \rangle$ = i^{TH} AV. CEPSTRAL COEF.
$\langle M \rangle$ = AV. PITCH VALUE
$\langle E \rangle$ = AV. LOG ENERGY
$\langle \lvert e \rvert \rangle$ = AV. NORMALIZED ABSOLUTE PREDICTION ERROR ENERGY

frame and L_v was the number of frames used in the averaging. Furthermore, since silence, voiced, and unvoiced speech are assumed to be sample functions of different random processes, only voiced frame features were utilized in (20).

From the training data set, both the dynamic programming procedure and the knock-out strategy were implemented to select the subset of ten out of the 32 features that had the maximum divergence. Table I summarized the feature selection process at each stage for the two schemes for $L_v = 10$.

These selected feature subsets were utilized in the linear classifier for identifying talkers text-independently. Fig. 4 shows a plot of the average identification error in the test data versus the number of frames used in the averaging, L_v, for features selected by dynamic programming, the knock-out strategy, the PARCOR, and the cepstral coefficients. From the graph, the features obtained from the two selection schemes yield much lower average identification error than that of the PARCOR and cepstral coefficients. The result is not surprising since the first two sets of attributes contain pitch and energy in addition to spectral components, whereas

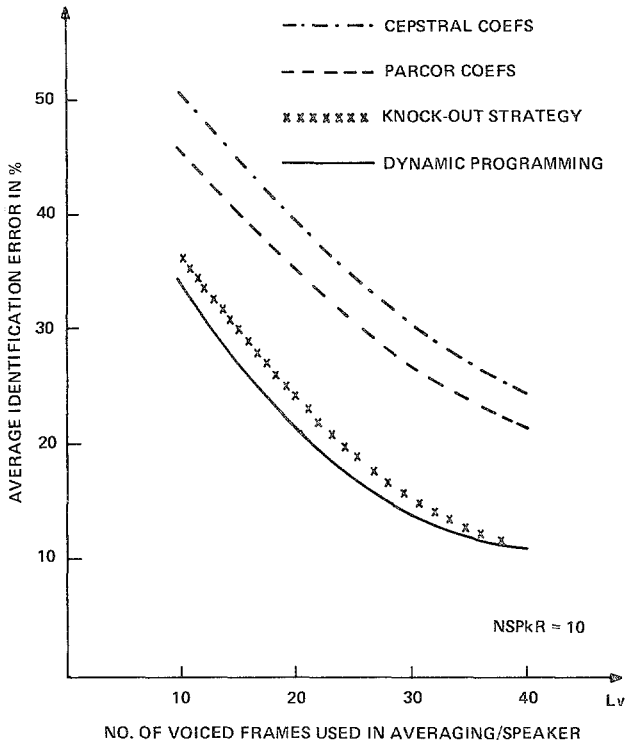

Fig. 4. Comparisons of average speaker identification errors for four feature sets in text-independent speaker identification.

the PARCOR and cepstral coefficients only have spectral information. Though the cepstral coefficients were reported to be the most effective linear prediction characteristics in text-dependent speaker identification [17], Fig. 4 shows that the PARCOR coefficients yield a lower identification error than the cepstral coefficients in the text-independent case. The graph in Fig. 4 also indicates that the features obtained from the dynamic programming performed slightly better than those derived from the knock-out strategy.

The effect of the number of features k in the selected subset on the identification error is shown in Fig. 5. Over 50 percent error is obtained when only two features are used. The average identification error gradually decreases for increasing k but the derivative of error starts to taper off at $k = 6$. As a practical matter, not much improvement in identification error is gained for $k > 7$.

V. Summary

This paper has presented the results of an investigation of the selection of feature subsets using dynamic programming in text-independent speaker identification. The procedure shown is a multistage decision process which selects the feature subset with maximum divergence using a functional recursive equation. No optimality is claimed in the scheme because for statistically dependent features, the divergence does not satisfy the separability condition, which in conjunction with monotonicity is sufficient for decomposing the k-stage problem into k subproblems. Nevertheless, the procedure allows the selection of a feature subset whose performance in text-independent speaker identification compares favorably to those obtained from the knock-out strategy, the PARCOR coefficients, and the cepstral coefficients.

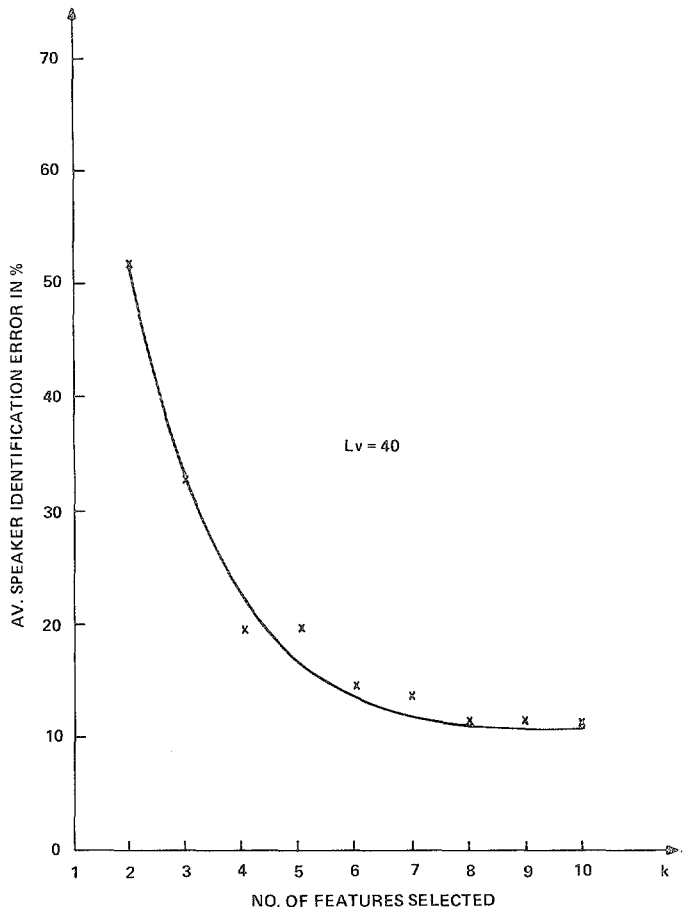

Fig. 5. Plot of average speaker identification error versus the number of features selected in the subset using dynamic programming.

Appendix A

To select a subset of k best features out of the available N, the number of searches required by the knock-out strategy is given by

$$N_{ko} = \sum_{i=1}^{k} (N - i + 1) = k(N + 1) - \frac{k(k+1)}{2}$$

$$= k\left(N - \frac{k-1}{2}\right). \tag{A-1}$$

Appendix B

The improvement ratio of the dynamic programming procedure over the exhaustive search is defined as

$$R = \frac{N_{ES}}{N_{DP}} = \frac{N!}{k!(N-k)!\,N(k-1)\left(N - \frac{k}{2}\right)} \tag{A-2}$$

where N_{ES} is the number of searches required by the exhaustive search and N_{DP} is that required by the dynamic programming method. Since $N > k$, R is simplified to

$$R = \frac{N(N-1)\cdots(N-k+1)}{k!\,N(k-1)\left(N - \frac{k}{2}\right)}, \tag{A-3}$$

and its value is bounded by

$$R < \frac{N^{k-2}}{k!(k-1)}. \quad \text{(A-4)}$$

ACKNOWLEDGMENT

The authors wish to thank their colleagues at GTE Sylvania Inc., Needham Heights, MA for helping to record the speech data, and in particular, they are grateful to Dr. A. J. Goldberg for his helpful suggestions.

REFERENCES

[1] R. C. Lummis, "Speaker verification by computer using speech intensity for temporal registration," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 80-89, Apr. 1973.
[2] G. R. Doddington, "A new method of speaker verification," *J. Acoust. Soc. Amer.*, vol. 49, p. 139(A), 1971.
[3] J. D. Markel, B. T. Oshika, and A. H. Gray, Jr., "Long-term feature averaging," presented at the 92nd Meeting Acoust. Soc. Amer., San Diego, CA, Nov. 1976.
[4] A. N. Mucciardi and E. E. Gose, "A comparison of seven techniques for choosing subsets of pattern recognition properties," *IEEE Trans. Comput.*, vol. C-20, pp. 1023-1031, Sept. 1971.
[5] M. R. Sambur, "Selection of acoustic features for speaker identification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 176-182, Apr. 1975.
[6] G. D. Nelson and D. M. Levy, "A dynamic programming approach to the selection of pattern features," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, pp. 145-151, July 1968.
[7] C. Y. Chang, "Dynamic programming as applied to feature subset selection in a pattern recognition system," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, pp. 166-171, Mar. 1973.
[8] T. Marill and D. M. Green, "On the effectiveness of receptors in recognition systems," *IEEE Trans. Inform. Theory*, vol. IT-9, pp. 11-17, Jan. 1963.
[9] F. Itakura and S. Saito, "On the optimum quantization of feature parameters in the PARCOR speech synthesizer," in *Conf. Rec., IEEE Conf. Speech Commun. and Process.*, New York, NY, 1972.
[10] R. Bellman and R. Kalaba, *Dynamic Programming and Modern Control Theory*. New York: Academic, 1965.
[11] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1972.
[12] P. D. Bricker *et al.*, "Statistical techniques for talker identification," *Bell Syst. Tech. J.*, vol. 50, pp. 1427-1454, Apr. 1971.
[13] G. T. Toussaint, "Recent progress in statistical methods applied to pattern recognition," in *Conf. Rec., 2nd Int. J. Conf. on Pattern Recognition*, pp. 479-488, Copenhagen, Denmark, 1974.
[14] B. S. Atal, "Automatic speaker recognition based on pitch contours," *J. Acoust. Soc. Amer.*, vol. 52, pp. 1687-1697, Dec. 1972.
[15] IEEE, "IEEE recommended practice for speech quality measurements," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 239-246, Sept. 1969.
[16] M. J. Ross, H. L. Shaffer, A. Cohen, R. Freudberg, and H. Manley, "Average magnitude difference function pitch extractor," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 353-362, Oct. 1974.
[17] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech waves for automatic speaker identification and verification," *J. Acoust. Soc. Amer.*, vol. 55, pp. 1304-1312, 1974.
[18] J. D. Markel and A. H. Gray, Jr., *Linear Prediction of Speech*. New York: Springer-Verlag, 1976.
[19] G. L. Nemhauser, *Introduction to Dynamic Programming*. New York: Wiley, 1966.
[20] S. Kullback, *Information Theory and Statistics*. New York: Wiley, 1959.

Intelligibility and Ratings of Digitally Coded Speech

DAVID J. GOODMAN, MEMBER, IEEE, JANET S. GOODMAN, AND MUN CHEN, MEMBER, IEEE

Abstract—**An experiment has been performed to investigate perceptual effects of digital encoding of speech. The aim was to gain information about the influence on intelligibility and subjective quality of three distortions: bandwidth reduction, peak clipping, and amplitude quantization. Allowing two levels of each distortion, we produced eight encoders encompassing all possible combinations of the three impairments. By means of a consonant recognition test (CRT) and a category-rating test, twelve subjects provided intelligibility and rating data about each encoder.**

The results show that the effect of multiple distortions is not, in general, the sum of effects of individual distortions and that the distortions influence intelligibility and subjective quality differently. For example, with respect to consonant recognition, quantization and clipping reinforce one another; occurring together they cause more recognition errors than the sum of the errors caused by the two distortions occurring individually. On the other hand, the effects of quantization and clipping on subjective quality are essentially additive.

Manuscript received November 29, 1977; revised April 18, 1978. This experiment was performed while D. J. Goodman and M. Chen were at Imperial College, London, England, and J. S. Goodman was at the University of Essex, Colchester, England.

D. J. Goodman is with Bell Laboratories, Holmdel, NJ 07733.
J. S. Goodman is with the British Post Office, London, England.
M. Chen is with the University of Singapore, Singapore.

I. BACKGROUND

ALTHOUGH a large body of information exists on the perceptual effects of analog distortions of speech signals [1], relatively little is known about impairments produced in digital communication. Some perceptual studies of digital systems have been oriented toward determining the levels of analog impairments that are perceptually equivalent to digital impairments [2] and other studies have had the purpose of describing the quality of specific systems. A more fundamental approach was taken in a recent paper [3] that identified three impairments associated with the pulse-code modulation (PCM) representation of speech: peak clipping, amplitude quantization, and band limiting. In that study a large number of transmission conditions were produced by independently adjusting

0096-3518/78/1000-0403$00.75