Considerations in applying clustering techniques to speaker-independent word recognition

L. R. Rabiner and J. G. Wilpon

Acoustics Research Department, Bell Laboratories, Murray Hill, New Jersey 07974
(Received 2 January 1979; accepted for publication 23 April 1979)

Recent work at Bell Laboratories has demonstrated the utility of applying sophisticated pattern recognition techniques to obtain a set of speaker-independent word templates for an isolated word recognition system [Levinson *et al.*, IEEE Trans. Acoust. Speech Signal Process. ASSP-27 (2), 134–141 (1979); Rabiner *et al.*, IEEE Trans. Acoust. Speech Signal Process.(in press)]. In these studies, it was shown that a careful experimenter could guide the clustering algorithms to choose a small set of templates that were representative of a large number of replications for each word in the vocabulary. Subsequent word recognition tests verified that the templates chosen were indeed representative of a fairly large population of talkers. Given the success of this approach, the next important step is to investigate *fully automatic* techniques for clustering multiple versions of a single word into a set of speaker-independent word templates. Two such techniques are described in this paper. The first method uses distance data (between replications of a word) to segment the population into stable clusters. The word template is obtained as either the cluster minimax, or as an averaged version of all the elements in the cluster. The second method is a variation of the one described by Rabiner [IEEE Trans. Acoust. Speech Signal Process. ASSP-26 (3), 34–42 (1978)] in which averaging techniques are directly combined with the nearest neighbor rule to simultaneously define both the word template (i.e., the cluster center) and the elements in the cluster. Experimental data show the first method to be superior to the second method when three or more clusters per word are used in the recognition task.

PACS numbers: 43.70.Sc

INTRODUCTION

Recent studies of isolated word recognition systems have shown that a set of carefully chosen templates can be used to bring the performance of speaker-independent systems up to that of systems trained to the individual speaker (Levinson *et al.*, 1979; Rabiner *et al.*, 1979). A key aspect of that work was that a very sophisticated set of pattern recognition algorithms was used, along with a fairly large amount of human intervention (i.e., decisions on merging, splitting, branching, etc.), to create the set of templates (multiple) for each word in the vocabulary. Not only is this procedure time consuming (e.g., it took about 30–45 min to cluster 100 repetitions of a single word) but it is impossible to reproduce exactly, and it is highly dependent on decisions made by the experimenter. As such this procedure is inappropriate for a general word recognition system. It is the purpose of this paper to investigate and discuss several fully automatic alternatives to the clustering approaches used by Levinson *et al.*, 1979.

Prior to discussing the automatic approaches which we have studied for clustering word data, it is worthwhile reviewing the structure of the entire word recognition system. Figure 1 shows a block diagram of the recognition system. There are three modes in which the system can run as determined by the position of the MODE switch. Mode 1 is a training mode in which the talker speaks a given word list (i.e., the word vocabulary) into a standard telephone, an autocorrelation analysis is made of the digitized speech (as determined by an endpoint detector), and the autocorrelation coefficients are put into a store. Mode 2 is a clustering mode in which a pattern recognition algorithm finds all replications of a given word, segments them into clusters, and updates a word reference store with a template representative of each cluster. Finally mode 3 is a testing mode in which the talker can say any word in the vocabulary, an autocorrelation and an LPC analysis is made of the digitized speech, a dynamic time warped distance between the unknown word and each reference template is made, and an appropriate decision rule chooses the recognized word. The recognition system of Fig. 1 has been successfully applied in a variety of word recognition contexts (Rabiner, 1978; Itakura, 1975; Rosenberg and Itakura, 1976; Levinson *et al.*, 1978; Rosenberg and Schmidt, 1977; Gupta *et al.*, 1978).

Earlier work on clustering used highly sophisticated, interactive, pattern recognition algorithms for obtaining a set of stable clusters for each word in the vocabulary. The word templates were chosen as the "minimax center" of the cluster, i.e., the point in the cluster whose maximum distance to all other points in the cluster was minimum. In this paper we consider several alternative procedures for clustering. In particular we investigate:

(1). Two fully unsupervised algorithms for clustering. One algorithm uses only the matrix of distances (similarity) between tokens of each word to be clustered and attempts to place each token uniquely in a cluster with all other tokens which are similar (distance within some threshold). A second algorithm attempts to combine (by averaging) tokens which are similar (small distance) to directly give both the cluster set and the cluster center.

(2). Differences between word templates obtained by the minimax center (i.e., an actual token) and those obtained by averaging techniques (i.e., an artifically created token).

(3). Differences between averaging different feature sets to give word templates from clustered data.

The organization of this paper is as follows. In Sec.

 0001-4966/79/090663-11$00.80

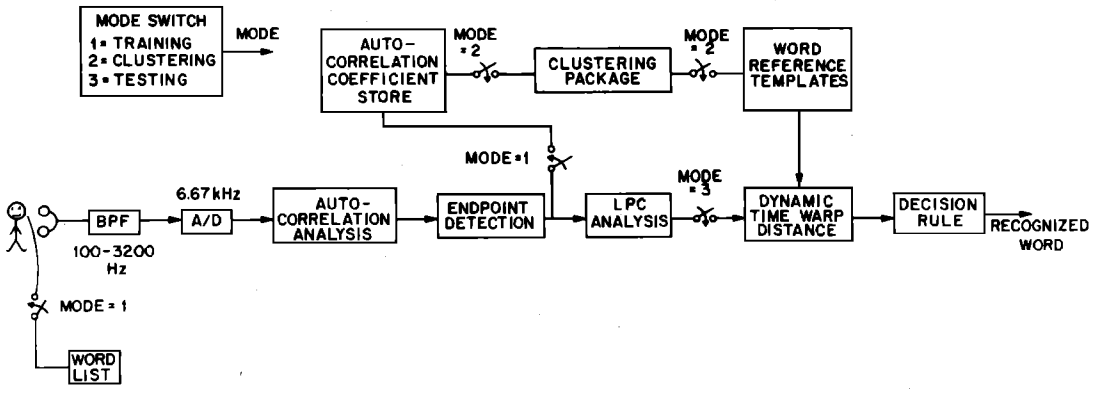


FIG. 1. Block diagram of the isolated word recognition system.

I we discuss the two unsupervised clustering procedures that were used. In Sec. II we discuss the applicability of averaging procedures to obtain word templates from clustered data. In Secs. III and IV we present results obtained from applying the techniques of Secs. I and II to the 39-word vocabulary of Levinson *et al.* (1979) and Rabiner *et al.* (1979). Finally in Sec. V we discuss the implication of the results for practical implementations of word recognition systems.

I. UNSUPERVISED ALGORITHMS FOR CLUSTERING WORD DATA

Following the development in Levinson *et al.*, 1979, we assume that we are given a finite set, Ω, of N observations

$$\Omega = \{x_1, x_2, \ldots, x_N\}, \tag{1}$$

where each observation x_i is a token representing a replication of a spoken word. Each token has an inherent duration (e.g., x_i is n_i frames long), and each frame of the token is some measured set of features. In the recognition system of Fig. 1, the feature set is the set of $(p+1)$ autocorrelation coefficients ($p=8$). Equivalently the set of p LPC coefficients or any transformation of them (Markel and Gray, 1975) could be used as the feature set.

Since it is intended that the clustering of the N observations be based entirely on distance (similarity) data (as is done in the actual recognition system), a distance d_{ij} between tokens x_i and x_j is defined as

$$d_{ij} = \delta(x_i, x_j) = \frac{1}{N_1} \sum_{k=1}^{N_1} d(k, w(k), i, j), \tag{2}$$

where the local frame distance $d(k, w(k), i, j)$ is the log likelihood distance proposed by Itakura (1975) between the kth frame of x_i and the $w(k)$th frame of x_j, i.e.,

$$d(k, w(k), i, j) = \log\left[\frac{(\mathbf{a}^j_{w(k)})' R^i_k (\mathbf{a}^j_{w(k)})}{(\mathbf{a}^i_k)' R^i_k (\mathbf{a}^i_k)}\right], \tag{3}$$

where $\mathbf{a}$ is the vector of LPC coefficients of the lth frame of token i, R^i_k is the matrix of autocorrelation coefficients of the kth frame of token i, and $'$ denotes vector transpose. The function $w(k)$ is the warping function obtained from a dynamic time warp match of token j to token i which minimizes d_{ij} over a constrained set of possible $w(k)$ (Levinson *et al.*, 1979; Sakoe and Chiba, 1971 and 1978; Rabiner *et al.*, 1978).

From the initial set of N tokens, an $N \times N$ distance matrix $\hat{D}$ can be defined with entry $\hat{d}_{ij}$ defined as

$$\hat{d}_{ij} = \frac{d_{ij} + d_{ji}}{2} = \frac{\delta(x_i, x_j) + \delta(x_j, x_i)}{2}. \tag{4}$$

Equation (4) yields a symmetric distance matrix ($d_{ij} = d_{ji}$) requiring storage for only $N(N-1)/2$ terms (since $d_{ii} = 0$ all i). The purpose of the clustering is to represent the set Ω as the union of M disjoint clusters, $\{\omega_i, i = 1, 2, \ldots, M\}$ such that

$$\Omega = \bigcup_{i=1}^{M} \omega_i. \tag{5}$$

The total number of clusters, M, need not be known or specified *a priori*. We denote the center of prototype of cluster ω_i as $\hat{x}_i$ and we note that $\hat{x}_i$ need *not* be a member of ω_i.

In the earlier supervised approach a sequence of four procedures was used interactively to determine both the number of clusters (M) and the tokens belonging to each cluster. The four procedures were the chainmap (which identified large prominent clusters), the shared nearest neighbor method (which identified overlap between clusters), the k-means procedure (which ascertained the detailed structure of the data), and the ISODATA procedure which merged and split the clusters until an optimal configuration was found (Levinson *et al.*, 1979). Although this approach was quite successful, the necessity for having a fully automatic clustering procedure led to the following methods.

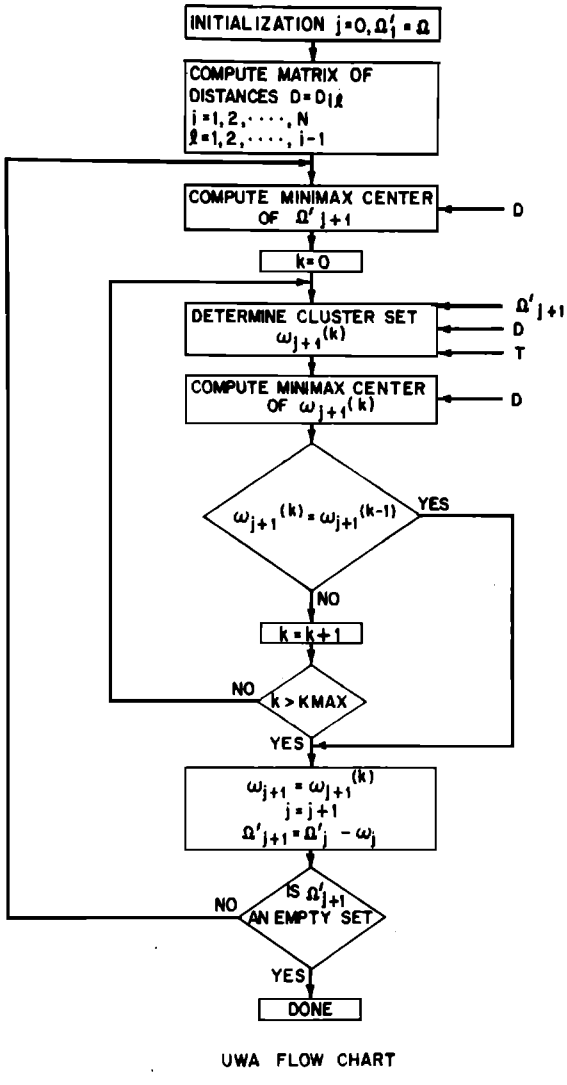
A. Unsupervised clustering without averaging (UWA)

A block diagram of the first procedure (called the UWA method) is given in Fig. 2. For notational purposes we define the partial observation set Ω'_{j+1} as the ordered observation set without the tokens that were included in clusters $\omega_1, \omega_2, \ldots, \omega_j$, i.e.,

$$\Omega'_{j+1} = \Omega - \bigcup_{i=1}^{j} \omega_i = \Omega'_j - \omega_j \tag{6}$$

$$= \{x'_1, x'_2, \ldots, x'_{q(j)}\}, \tag{7}$$

where x'_i is an element of set Ω, and $q(j)$ is the number of tokens that remain to be clustered after the first j clusters have been formed. [By definition $q(0) = N$.]



UWA FLOW CHART

FIG. 2. Flow diagram of the UWA clustering procedure.

The UWA clustering algorithm uses the following steps:

1. Initialization – $j = 0$.

2. Determination of the minimax center of the observation set Ω'_{j+1}. (Initially $j = 0$ and $\Omega'_1 = \Omega$). We denote the minimax center as $\hat{x}_{j+1}$ which is obtained as

$$\hat{x}_{j+1} = x'_{i*} \ni \max_j \delta(x'_{i*}, x'_j) \leq \min_i \max_j \delta(x'_i, x'_j), \tag{8}$$

i.e., the minimax center is the token x'_{i*} such that the maximum distance to any other token in Ω'_{j+1} is minimum. Since all distances of any token in Ω to any other token in Ω are precomputed and stored in D, minimax computations of the type given in Eq. (8) are especially simple to implement.

3. Initial choice ($k = 0$) of the cluster ω_{j+1} as

$$\omega_{j+1}^{(k)} = \bigcup_{i \in \Omega'_{j+1}} x'_i \ni \delta(\hat{x}_{j+1}, x'_i) \leq T, \tag{9}$$

where T is a user-defined distance threshold. Thus the initial choice of the $(j+1)$st cluster is the set of all tokens in Ω'_{j+1} that are within a given distance of the cluster center $\hat{x}_{j+1}$.

4. Determination of the minimax center of $\omega_{j+1}^{(k)}$ using Eq. (8) on only the tokens in $\omega_{j+1}^{(k)}$.

5. Increment k and determine $\omega_{j+1}^{(k)}$ using Eq. (9). Check if $\omega_{j+1}^{(k)} = \omega_{j+1}^{(k-1)}$ of if $k > K\text{MAX}$, a user-supplied iteration check. If either is true the jth cluster is obtained as $\omega_{j+1}^{(k)}$, j is incremented, and the observation set Ω'_{j+1} is obtained from Eq. (6). The algorithm proceeds to step 2 as long as Ω'_{j+1} is not an empty set. If neither check above is true the algorithm proceeds to step 4 and continues.

It can be seen that prominent, distinct clusters will be readily found by this procedure since the cluster sets at consecutive iterations will be identical. However, for highly overlapping data, as the cluster center changes so does the cluster composition, causing the need for several iterations. These iterations are reminiscent of the merge and split phases of ISODATA (Ball and Hall, 1965).

The only user-supplied inputs to the UWA method are the lower half-matrix of distances, the number of observations N, the distance threshold T, and the maximum iteration count $K\text{MAX}$. Initial values of T are chosen based on theoretical estimates of LPC distances (Itakara, 1975; Tribolet *et al.*, 1979); however the algorithm itself can modify T to increase (lower T) or decrease (raise T) the total number of clusters required to represent the observation set. Typically a maximum iteration count of 10 is used. For most clusters only 1 or 2 iterations are required to obtain a stable result.

The UWA algorithm is fully automatic and can cluster a set of 100 observations into from 10 to 25 clusters in about 1 min. Each cluster is represented by a cluster center $\hat{x}$. We have considered two distinct methods of obtaining the cluster center. Consider cluster ω_i, with J tokens, i.e.,

$$\omega_i = \{x'_1, x'_2, \ldots, x'_J\}. \tag{10}$$

We define the L_p norm of ω_i as

$$x_{LP} = \frac{1}{J}\left[\sum_{j=1}^{J} (x'_j)^p\right]^{1/p}, \tag{11}$$

where we define the averaging in Eq. (11) as proceeding on a frame-by-frame basis. For time normalization we define the minimax center of ω_i to be the standard and we warp each of the J tokens to the minimax center, and then average the warped tokens to give x_{LP}. The two cluster centers we have considered are

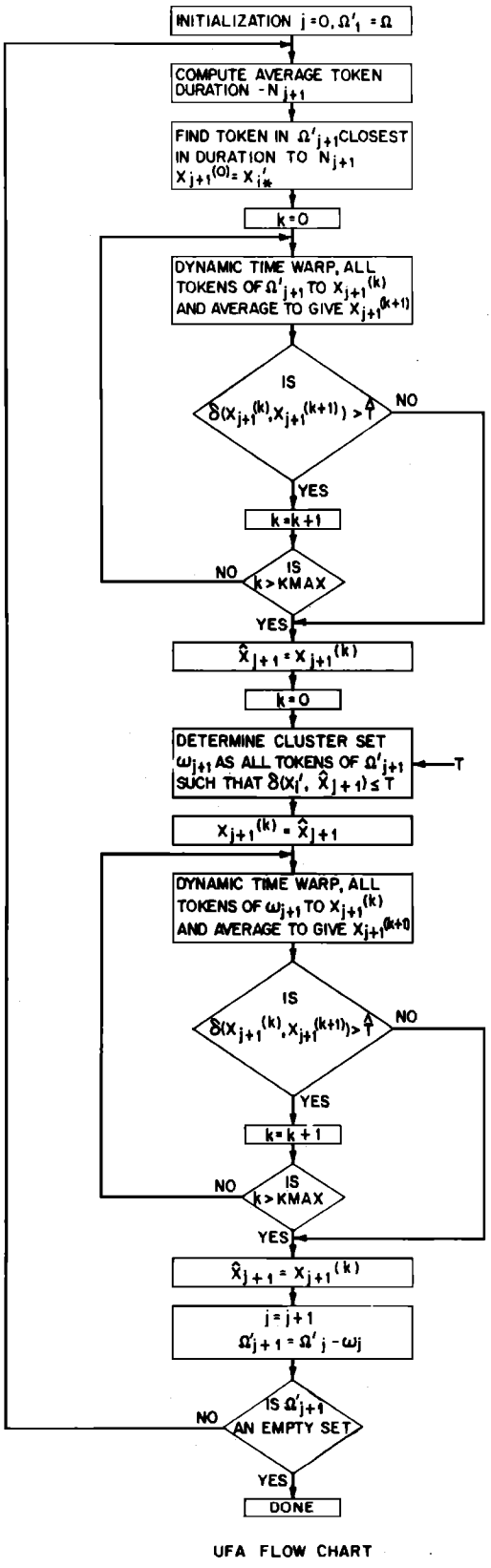
1. The minimax center, as defined previously.

2. The "average" token as obtained from Eq. (11) with $p = 1$.

The details of the averaging procedure will be discussed in Sec. II of this paper. Although *we have not investigated larger values of p* in Eq. (11), they may be of interest in some applications, e.g., the rms token for $p = 2$ or the "largest" token for $p = \infty$.

B. Unsupervised clustering with full averaging (UFA)

The second unsupervised clustering algorithm we have considered is one which attempts to find clusters in the



UFA FLOW CHART

FIG. 3. Flow diagram of the UFA clustering procedure.

vicinity of the averaged center of the current observation set. A flow diagram of the UFA method is given in Fig. 3. This method is similar to the one used by Rosenberg and Sambur (1975) for clustering speaker verification features, and by Rabiner (1978) for word clustering.

There are basically three stages to the UFA algorithm. In the first stage the averaged center of the current observation set is found by recursively warping each token of the observation set to an estimate of the center, and updating the center estimate by averaging the warped tokens. For the second stage the elements of the current cluster are found as those tokens of the observation set whose distances to the estimate of the cluster center (as found in the first stage) is less than some specified threshold. (If the cluster set is empty, the threshold is increased progressively until at least a single token is in the cluster. This situation may occur with outlier points.) The third stage of the procedure is to recursively estimate the center of the cluster set obtained in the second stage using the procedure of the first stage.

Based on the above discussion and the flow chart of Fig. 3, the detailed steps of the UFA method are:

1. Initialization in which the cluster center (j) is set to 0 and the initial observation set Ω'_{j+1} is set to Ω, the entire set of tokens.

2. Determination of the $k=0$th estimate of the cluster center, $\hat{x}^{(0)}_{j+1}$, of the $(j+1)$st cluster by first finding the average duration (in frames) of the tokens in the set Ω'_{j+1}, and then finding the token in Ω'_{j+1} that is closest in duration to the average.

3. Determination of the averaged center of the observation set Ω'_{j+1} by dynamic time warping each token of Ω'_{j+1} to $\hat{x}^{(k)}_{j+1}$ and then averaging (on a frame-by-frame basis) the features of the tokens. Thus if we denote Ω'_{j+1} as

$$\Omega'_{j+1} = \{x'_1, x'_2, \ldots, x'_L\} \tag{12}$$

then

$$\hat{x}^{(k+1)}_{j+1} = \frac{1}{L}\left[\sum_{l=1}^{L} x'_l\right], \tag{13a}$$

where the jth frame of $\hat{x}^{(k+1)}_{j+1}$ is

$$\hat{x}^{(k+1)}_{j+1}(i) = \frac{1}{L}\left[\sum_{l=1}^{L} x'_l(w(i))\right], \tag{13b}$$

where $w(i)$ is the warping path obtained from warping x'_l to $\hat{x}^{(k)}_{j+1}$ (Sakoe and Chiba, 1971 and 1978). The iterations of Eqs. (12) and (13) terminate when either $k > K$ MAX, a user defined iteration count, or when

$$\delta(\hat{x}^{(k)}_{j+1}, \hat{x}^{(k+1)}_{j+1}) < T' \tag{14}$$

where T' is a small distance indicating center point convergence. (The reader should note that the dynamic time warps required in Eq. (13) cannot be precomputed; thus this step is extremely time consuming.)

4. Determination of the cluster set ω_{j+1} as all tokens of Ω'_{j+1} within a specified distance of the cluster center of the preceeding step. This step is also a time consuming one in that distances cannot be precomputed.

5. Determination of the averaged center of the cluster set ω_{j+1}. This procedure is identical to the one of step 3 except the $k=0$th estimate of the cluster center is the final estimate of step 3, and the set of tokens is

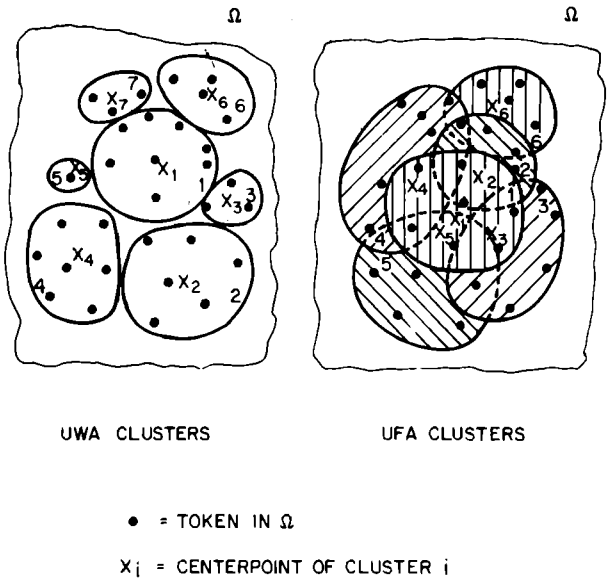


FIG. 4. Example (two-dimensional) showing fundamental differences in UWA and UFA clusters for an observation set Ω.

drawn from ω_{j+1} instead of Ω'_{j+1}.

6. Updating the cluster counter j, the observation set Ω'_{j+1} and checking if Ω'_{j+1} is empty. If so the procedure is done; if not we return to step 2 and iterate the entire procedure.

As noted above, the UFA algorithm is considerably slower than the UWA algorithm since all distances must be computed as needed. Typically it takes about 1 h to automatically cluster 100 tokens of a word into a set of from 10 to 25 clusters for the UFA method. This is about 60 times longer than the UWA method.

Although no formal proof exists, we have found experimentally that the UWA and UFA algorithms tend to cluster data in significantly different ways. Figure 4 shows an abstract plot of a set of data and a series of circles indicating how the data would be clustered using the UWA and UFA methods. The UWA method can localize each cluster anywhere in the observation space and thus clusters tend to be balls distributed in the observation space which are uniform in their coverage of the space. Because of the recursive averaging of the UFA method, the clusters tend to be centered near the center of the observation space and consist of balls with significantly larger overlap than for the UWA method. We will see the effects of such clustering in the statistics of the clusters given in Sec. III, and in the recognition results of Sec. IV.

II. AVERAGING TECHNIQUES USED TO OBTAIN CLUSTER CENTERS

For both algorithms of Sec. I the concept of averaging tokens of the observation set to form a cluster center was discussed. Thus one important consideration in implementing the algorithms for clustering is the manner in which this averaging is carried out. To facilitate this discussion we consider two tokens x and y in the observation set Ω. Token x is assumed to consist of N_x frames of LPC features, and token y consists of N_y frames of features, where each frame is a set of $p+1$ ($p=8$ in our system) unnormalized autocorrelation coefficients. If we denote the ith frame of x (or y) as $x(i)$ (or $y(i)$) then we can represent x and y as the set of vectors

$$x=(x(1),x(2),\ldots,x(i),\ldots,x(N_x)), \tag{15a}$$

$$y=(y(1),y(2),\ldots,y(i),\ldots,y(N_y)), \tag{15b}$$

where

$$x(i)=(x_0(i),x_1(i),\ldots,x_p(i)) \tag{16}$$

and similarly for $y(i)$. Although the features we are using to represent the token are the unnormalized autocorrelation coefficients of each frame, alternative, equally attractive feature sets can be derived by suitable transformations (Markel and Gray, 1975). Included among such feature sets are the LPC coefficients, the log areas, the PARCOR coefficients, the roots of the LPC polynomial, etc.

In order to average tokens x and y we must have a correspondence between frames of x and frames of y. For simplicity, we assume token y is being mapped to token x. As such a dynamic time warping procedure is used to give the mapping

$$x(i)\leftarrow y(k)=y(w(i)) \tag{17a}$$

or

$$k=w(i),\quad i=1,2,\ldots,N_x, \tag{17b}$$

i.e., the ith frame of token x corresponds to the $k=w(i)$th frame of token y. As such when we average tokens x and y we produce token z

$$z=(z(1),z(2),\ldots,z(N_x)), \tag{18}$$

where

$$z(i)=\tfrac{1}{2}[x(i)+y(w(i))] \tag{19a}$$

and the kth component of $z(i)$, i.e., $z_k(i)$, is obtained simply as

$$z_k(i)=\tfrac{1}{2}[x_k(i)+y_k(w(i))] \tag{19b}$$

When we average Q tokens of Ω, we successively warp each of the tokens to the estimated center of the cluster and then average the time registered patterns. (Of course we normalize by $1/Q$ in this case.)

The most important consideration in performing the averaging is the choice of feature sets. The issue here is the stability of the resulting averaged feature set. Although each individual feature set is guaranteed stable (by the LPC method of analysis), when certain LPC feature sets are averaged, stability cannot be guaranteed. To illustrate this point Fig. 5 shows a series of root locus plots obtained by taking the weighted average

$$z_k(i)=\alpha x_k(i)+(1-\alpha)y_k(w(i)), \tag{20}$$

where α went from 0 to 1 in steps of 0.01. For $\alpha=0$ (as indicated by the 0's in the plots) the roots of the averaged LPC polynomial are at the roots of the LPC polynomial for the $w(i)$th frame of token y. For $\alpha=1$ (as indicated by the x's in the plots) the roots are at the positions of the roots of the ith frame of token x. For other values of α the roots move continuously between

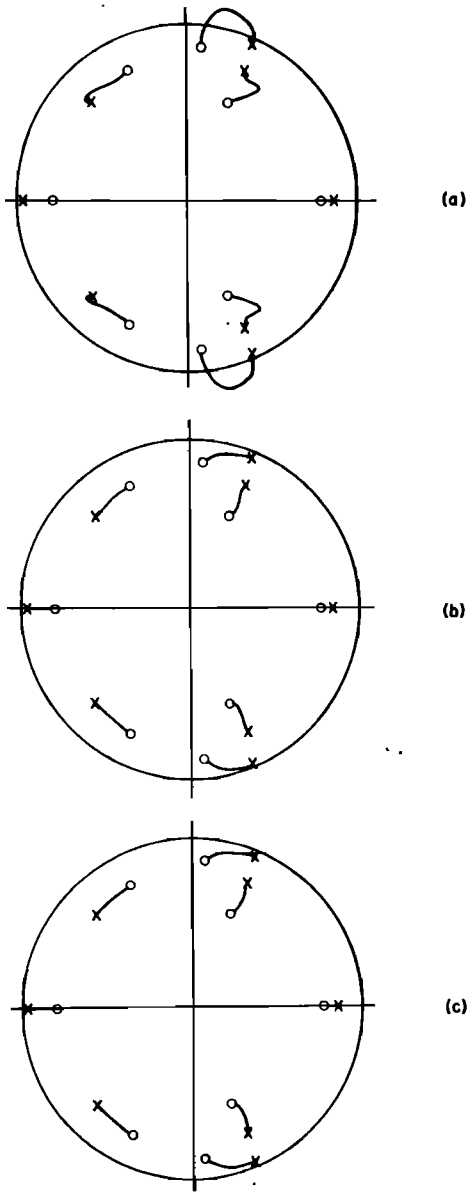


FIG. 5. Variations of the root locations of an LPC polynomial as a function of the averaging coefficient α for averaging of (a) normalized autocorrelation coefficients, (b) log areas, and (c) arcsin PARCOR coefficients.

these positions. Figure 5(a) shows one example of the effect of averaging normalized autocorrelation coefficients. It is readily seen that the root locus is generally an irregular (but continuous) path between the roots (Shadle and Atal, n.d.). It is also seen that, for this example, one pair of roots goes outside the unit circle (unstable) for a large range of values of α. This result is possible since stability is not guaranteed when autocorrelation coefficients are averaged. Figure 5(b) and 5(c) show similar plots when the features that are averaged are the log areas and the arcsin of the PARCOR coefficients, respectively. For these plots we see very different root loci; however more significantly we see that the roots always stay inside the unit circle thereby guaranteeing stability of the averaged system.

Alternative indications of the effects of averaging different feature sets are shown in Fig. 6 which shows two plots of the variation of the (pseudo) area function of the vocal tract corresponding to values of α from 0 to 1 in steps of 0.2. Figure 6(a) shows the result obtained when the normalized autocorrelation coefficients are averaged: Figure 6(b) is for averaging of log areas. Significant differences are seen in the (pseudo) areas for $\alpha = 0.4$ and 0.6 for the two different averaged feature sets.

In light of the above discussion it is seen that averaging different LPC feature sets can produce significant differences in the resulting (pseudo) areas, roots, etc. However when both the feature sets being averaged are from essentially the same speech sounds, and when a sufficient number of them are averaged, we would expect that almost any LPC feature set can be averaged and still give good results. These conditions are met when the tokens to be averaged all lie within a single cluster (as in the UWA method). For the UFA method these conditions are not generally as well met and we might expect some significant differences in the results depending on which feature set is averaged.

Based on the above discussion, and informal experimentation with the various parameter sets, three feature sets were chosen for averaging in the clustering algorithms. These were the normalized autocorrelation coefficients ($x_k = R_k$), the log areas ($x_k = g_k$), and the arcsin ($\sin^{-1}$) of the PARCOR coefficients ($x_k = p_k$). The arcsin transformation of the PARCOR coefficients is a transformation, suggested by Atal, for spreading out the distribution of the PARCOR coefficients around the peak (i.e., when the PARCORS are close to 1). In the next two sections we present results obtained on clustering using these three feature sets for averaging.

III. OBJECTIVE EVALUATION OF CLUSTERING ALGORITHMS

The clustering algorithms and averaging techniques of the preceeding sections were applied to a 39-word speech vocabulary consisting of the letters (A to Z), the digits (0 to 9), and the cueing words STOP, ERROR, and REPEAT (Rabiner *et al.*, 1979). A total of 100 replications of each word of the vocabulary from 50 different male and 50 different female talkers were used as the tokens in the observation set. The 100 tokens of each of the 39 words were clustered by the following procedures:

C1-UWA algorithm, cluster centers obtained as the minimax centers,

C1R-UWA algorithm, cluster centers obtained by averaging autocorrelation coefficients,

C1G-UWA algorithm, cluster centers obtained by averaging log area coefficients,

C1P-UWA algorithm, cluster centers obtained by averaging arcsin PARCOR coefficients,

C2R-UFA algorithm with autocorrelation coefficient averaging,

C2G-UFA algorithm with log area averaging,

C2P-UFA algorithm with arcsin PARCOR averaging,

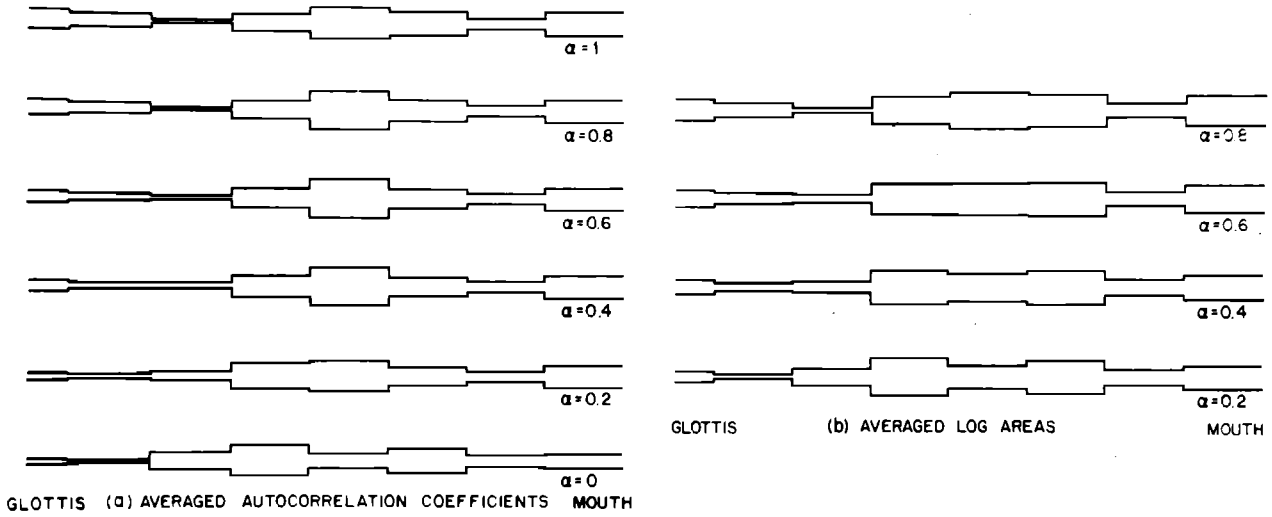


FIG. 6. Variations of the (pseudo) area function as a function of the averaging coefficient α for averaging of (a) normalized autocorrelation coefficients, and (b) log areas.

C3- Supervised algorithm of Levinson *et al.* (1979), cluster centers obtained as the minimax centers,

C3R- Supervised algorithm with cluster centers obtained by averaging autocorrelation coefficients.

The results using the C3 procedure provide a bound on the performance obtained by any of the automatic algorithms in the list above if we assume that a supervised approach is at least as good as any unsupervised pattern recognition procedure. The results using the C3R procedure provide a comparison between averaging and minimax methods for obtaining cluster centers, and provide a bound on the UWA clusters with post averaging to obtain the cluster centers, i.e., the C1R, C1G, and C1P results.

One objective measure of the "quality" of the clustering is the measure

$$\sigma = \frac{\dfrac{1}{(M)(M-1)} \sum_{i=1}^{M} \sum_{j=1}^{M} \delta(\hat{x}_j, \hat{x}_i)}{\dfrac{1}{M} \sum_{i=1}^{M} \dfrac{1}{m_i(m_i-1)} \sum_{j=1}^{m_i} \sum_{k=1}^{m_i} \delta(x_j^{(i)}, x_k^{(i)})}, \tag{21}$$

i.e., the ratio of the average intercluster distance (i.e., the average distance between cluster centers) to the average intra-cluster distance (Levinson *et al.*, 1979). In Eq. (21) we assume the $N = 100$ observations are clustered into M classes, and the number of tokens in the ith cluster is m_i. As noted in Levinson *et al.* (1979), for two spherically symmetric clusters, $\sigma > 2$ implies no overlap.

Table I gives results obtained on the 9 clustering algorithms for the quality ratio (σ), and 3 other clus-

TABLE I. Statistics of the clustering algorithms.

		Number of clusters per word	Number of outliers per word	Size of largest cluster	Quality ratio (σ)	Quality ratio (σ)		
						R	*G*	*P*
C1-	avg	10	13	27	2.95	3.88	3.50	3.45
	min	3	6	18	2.21	2.66	2.39	2.37
	max	13	24	47	4.22	5.26	4.62	4.52
C2R	avg	8	5	60	2.20			
	min	3	0	27	1.52			
	max	16	11	90	2.99			
C2G	avg	9	5	51	1.83			
	min	3	0	16	0.86			
	max	15	11	82	2.52			
C2P	avg	9	6	52	1.93			
	min	3	2	14	1.35			
	max	15	11	85	2.85			
C3-	avg	13	8	24	2.95	3.62		
	min	8	3	12	2.41	2.61		
	max	19	16	36	3.88	4.70		

ter statistics:

1. Number of clusters per word.

2. Number of outliers (i.e. clusters with a single token) per word.

3. Size (in tokens) of the largest cluster.

The average (across all 39 words), and the minimum and maximum of each of the quantities is given in Table I. For the algorithms with post averaging (C1R, C1G, C1P, and C3R) the quality statistics are given following the results using the minimax center.

Several interesting observations can be made from the data in Table I. First we see that post averaging of tokens to obtain cluster centers yielded significantly larger σ ratios than those obtained from minimax centers. (Contrast C1R and C1, or C3R and C3.) It is also seen that the fully automatic results obtained using the UWA algorithm are essentially comparable to results obtained from the fully supervised approach (C3). However results obtained from the UFA algorithm were significantly worse than any other method. As anticipated in the earlier discussion, this is due to the extraordinarily large number of tokens in the biggest cluster (54 on average across C2R, C2G, and C2P). This result indicates that the tokens tended to be distributed in a small ring with respect to the averaged center of the set, and the largest cluster contained a significant slice of the ring. Informal experimentation with the distance threshold procuded small differences in these results. These cluster statistics therefore indicate that the UFA method would yield good results if we were interested in only 1–2 clusters per word.

In summary the cluster statistics indicate that the UWA method appears capable of providing a reasonable set of clusters with separability that is essentially as good as a previously successful supervised approach. Also it was shown that obtaining cluster centers by averaging methods yield significant improvements in cluster separability over using the minimax center.

IV. RECOGNITION RESULTS

An alternative, and more significant, measure of the performance of the clustering procedures is the recognition accuracy obtained in the system for which the templates were designed. As such we have tested the nine procedures (along with a randomly chosen set of templates) on the first three test sets discussed in Rabiner *et al.* (1979). These test sets (denoted TS1, TS2, and TS3) contained, respectively:

TS1-1 repetition of the 39-word vocabularly by each of ten talkers (five male, five female) who were not part of the training set. Recordings from analog tape.

TS2-1 repetition of the 39-word vocabulary by each of eight talkers (four male, four female) who were not part of the training set. Recordings in real time using a high speed floating point array processor.

TS3-10 repetitions of the 39-word vocabulary chosen at random from the 100 talkers who trained the system. (No token here was part of the training set).

Recognition accuracies for TS1 were obtained as a function of p, the number of templates per word used in the reference set, where p varied from 1 to 12, and as a function of the *position*, c, of the actual word in the final candidate list. For TS2 and TS3, recognition accuracies were measured for a fixed number (12) of templates per word and as a function of c for values of c from 1 to 5. In addition recognition accuracies were measured for a set of randomly chosen templates from the original training set. These results hopefully provide an underbound on the accuracy obtainable from the clustering procedures.

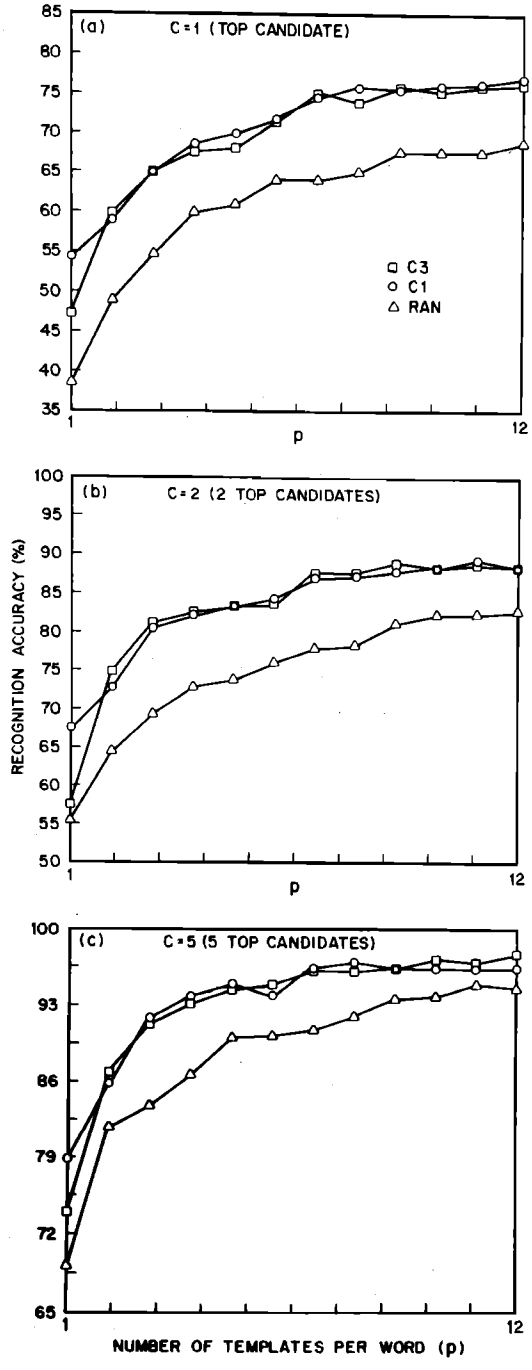


FIG. 7. Recognition accuracy as a function of p for the data of TS1 for $c=1$ (a), $c=2$ (b), and $c=5$ (c) for the C1, C3, and RAN clustering procedures.

Results of the recognition tests are shown in Figures 7–11. Figures 7–9 show plots (for TS1 data) of the recognition accuracy as a function of p for $c = 1$ (a), $c = 2$ [top two candidates— (b)] and $c = 5$ [top five candidates—(c)]. The decision rule for recognition is the KNN rule discussed in Rabiner *et al.* (1979) in which KNN = 1 for small values of p and KNN = 2 or 3 for p larger than about 4. Figure 7 shows the comparisons between the C1, C3, and RAN (random template) algorithms. It can be seen that, except for $p = 1$, the C1 and C3 algorithms provide essentially identical recognition accuracies for all p and c. For $p = 1$ the C1 algorithm provides an improvement in recognition accuracy of from 5% to 10% over the C3 algorithm for different values of c. This result says the single biggest template of the UWA procedure provides a better representation (on average) of each word than the single biggest template of the supervised approach. However once we use two or more templates per word, the recognition accuracies of both procedures are comparable. Figure 7 also shows significantly poorer recognition accuracy from the randomly chosen templates than from either clustering approach.

Figure 8 shows a comparison of the recognition accuracies for the post-averaged algorithms—namely C1R, C1G, C1P, C3R. It can be seen that for $c = 1$, the C3R and C1R provide from 3% to 5% higher accuracy than the C1G and C1P procedures for values of p from 3 to 8. Also, except for $p = 1$, the C3R provides essentially the highest recognition accuracies (by about 1%–2%) of the four procedures. This result is anticipated from the earlier discussions. For $c = 2$ the C3R procedure gives a 2% higher recognition accuracy than the other procedures (except for $p = 1$). For $c = 5$ all the recognition accuracies are comparable (to within ±1%). By comparing Figs. 7 and 8, averaging to give cluster centers provides large improvements in recognition accuracy for small values of p, and small improvements near $p = 12$. However in almost all cases the recognition accuracy is higher with the averaging techniques.

Figure 9 shows a comparison of the recognition accuracies for the full averaging procedures—C2R, C2G, C2P. These results show that the averaging of autocorrelation coefficients provided consistently better

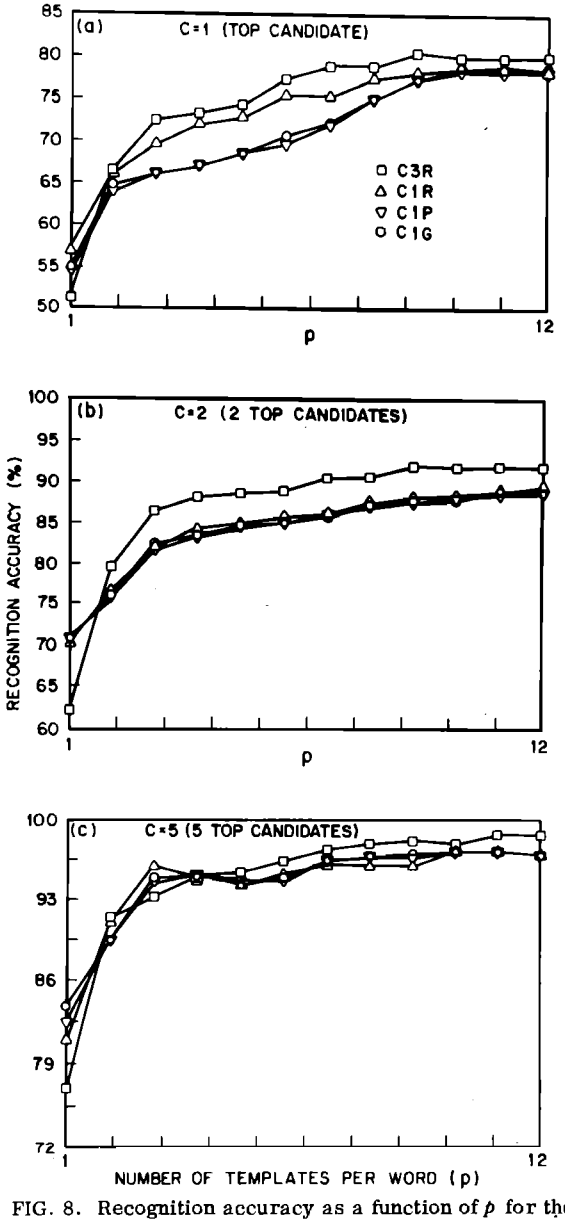


FIG. 8. Recognition accuracy as a function of p for the data of TS1 for $c = 1$ (a), $c = 2$ (b), and $c = 5$ (c) for the C1R, C1G, C1P, and C3R clustering procedures.

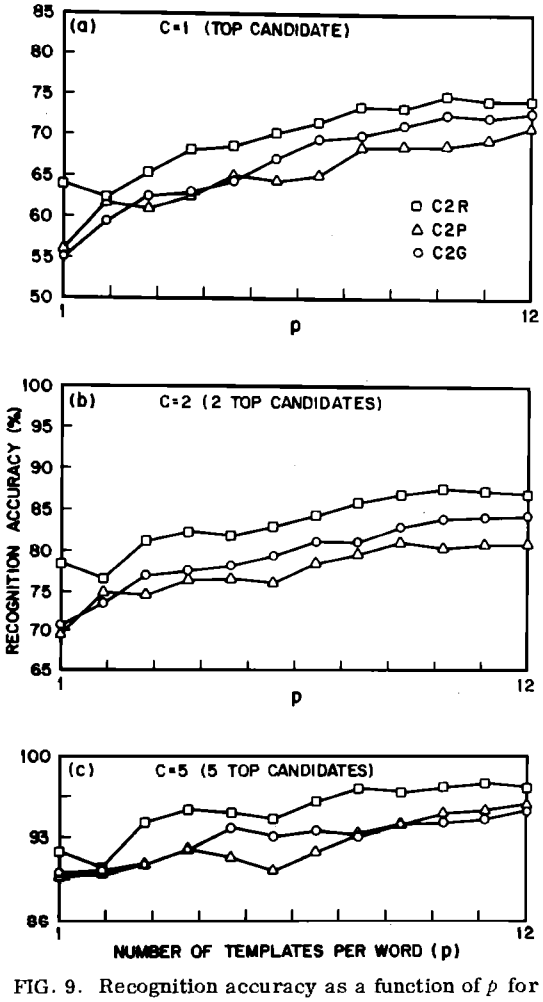


FIG. 9. Recognition accuracy as a function of p for the data of TS1 for $c = 1$ (a), $c = 2$ (b), and $c = 5$ (c) for the C2R, C2G, and C2P clustering procedures.

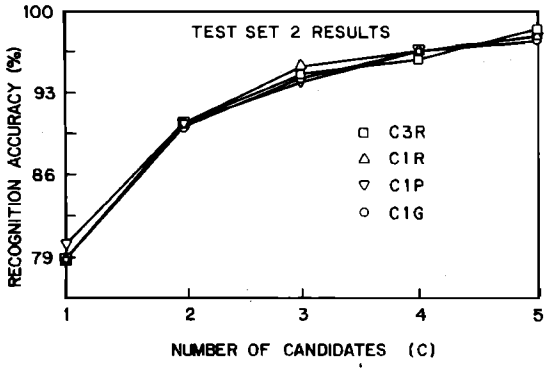


FIG. 10. Recognition accuracy as a function of c for the data of TS2 for the C1R, C1G, C1P, and C3R clustering procedures.

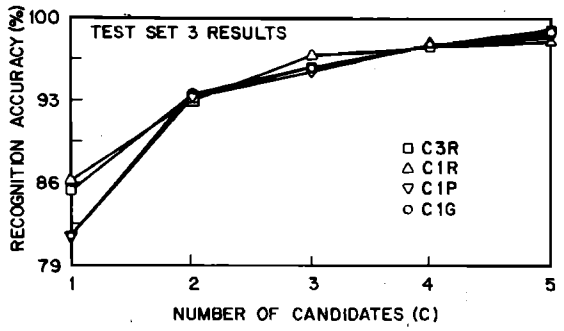


FIG. 11. Recognition accuracy as a function of c for the data of TS3 for the C1R, C1G, C1P, and C3R clustering procedures.

results than the averaging of log areas or arcsin PARCOR's. However, except for $p = 1$, it is seen that the recognition accuracies of the best C2 algorithm (C2R) were not as high as those of the C1R algorithm. For $p = 1$ the C2R procedure always gave significantly higher recognition accuracies (by about 7%) than any other procedure. Thus if we were truly interested in the best, single universal template to represent each word in the vocabularly, the fully averaging clustering procedure would yield the best results.

Figures 10 and 11 show recognition results from TS2 and TS3, respectively. For each of these figures, recognition accuracy is plotted as a function of c, the number of candidates considered, for $p = 12$ templates per word. Results are plotted for the four post-averaging procedures (C1R, C1G, C1P, and C3R) since these yielded uniformly the highest accuracies. The results given in these figures show that only small differences occur in the performance of these different procedures. In general the recognition accuracy is about 80% for the top candidate, and increases to about 98% for the top five candidates.

V. DISCUSSION AND SUMMARY

The purpose of this investigation was to determine if a fully automatic word template clustering procedure could obtain the performance of a previously investigated, supervised approach to clustering. To this end two procedures were described—one in which the clusters were obtained from a matrix of distances between pairs of tokens, and one in which averaging techniques were heavily relied on to provide estimates of cluster centers from which individual clusters could be defined. In addition we were interested in finding out if the method in which the cluster center was obtained would strongly affect either the quality of the clusters or the recognition accuracy of the system.

Based on the results presented in the previous section, the following statements can be made:

1. The UWA algorithm is capable of clustering word data as well as the supervised approach, and significantly better than random selection of templates.

2. Obtaining cluster centers by averaging is always as good as or better than obtaining cluster centers by minimax techniques. The performance of the UWA method with post-averaging is slightly worse than the supervised method with post-averaging.

3. The UWA method with averaging of autocorrelation coefficients to give cluster centers provides performance which is as good as, or better than that obtained when other LPC feature sets are averaged.

4. The UFA method provides the best, *single template*, representation of each word. However when multiple templates per word are used, the incorporation of averaging into the clustering procedures appears to lump together too many tokens in the largest cluster, thereby making the following clusters hard to find in a reasonable manner. As such this procedure should not be used when multiple clusters are desired.

The results of this investigation indicate that a reasonably simple, fully automatic clustering procedure can be used in a speaker-independent, isolated word recognition system and still provide good performance.

Ball, G. H., and Hall, D. J. (1965). "Isodata—an Iterative Method of Multivariate Analysis and Pattern Classification," Proc. IFIPS Congress.

Gupta, V. N., Bryan, J. K., and Gowdy, J. N. (1978). "A Speaker Independent Speech Recognition System Based on Linear Prediction," IEEE Trans. Acoust. Speech Signal Process. ASSP-26(3), 27–33.

Itakura, F. (1975). "Minimum Prediction Residual Applied to Speech Recognition," IEEE Trans. Acoust. Speech Signal Process. ASSP-23(1), 67–72.

Levinson, S. E., Rabiner, L. R., Rosenberg, A. E., and Wilpon, J. G. (1979). "Interactive Clustering Techniques for Selecting Speaker Independent Reference Templates for Isolated Word Recognition," IEEE Trans. Acoust. Speech Signal Process. ASSP-27(2), 134–141.

Levinson, S. E., Rosenberg, A. E., and Flanagan, J. L. (1978). "Evaluation of a Word Recognition System Using Syntax Analysis," Bell System Tech. J., 57(5), 1619–1626.

Markel, J. D., and Gray, Jr., A. H. (1975). *Linear Prediction of Speech* (Springer-Verlag, Berlin).

Rabiner, L. R., Rosenberg, A. E., and Levinson, S. E. (1978). "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition," IEEE Trans. Acoustics, Speech, Signal Proc. ASSP-26(5), 575–582.

Rabiner, L. R. (1978). "On Creating Reference Templates for Speaker Independent Recognition of Isolated Words," IEEE Trans. Acoust. Speech Signal Process. ASSP-26(3), 34–42.

Rabiner, L. R., Levinson, S. E., Rosenberg, A. E., and Wilpon, J. G. (1979). "Speaker Independent Recognition of

Isolated Words Using Clustering Techniques," IEEE Trans. Acoust. Speech Signal Process. (in press).
Rosenberg, A. E., and Schmidt, C. E. (1972). "Recognition of Spoken Spelled Names Applied to Directory Assistance," J. Acoust. Soc. Am. Suppl. 1 62, S563(A).
Rosenberg, A. E., and Itakura, F. (1976). "Evaluation of an Automatic Word Recognitions System Over Dialed-Up Telephone Lines," J. Acoust. Soc. Am. Suppl. 1 60, S12(A).
Rosenberg, A. E., and Sambur, M. R. (1975). "New Techniques for Automatic Speaker Verification," IEEE Trans. Acoust. Speech Signal Progess. ASSP-23(2), 169-176.
Sakoe, H., and Chiba, S. (1978). "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," IEEE Trans. Acoust. Speech Signal Process. ASSP-26(1), 43-49.
Sakoe, H., and Chiba, S. (1971). "A Dynamic Programming Approach to Continuous Speech Recognition," Proc. Int. Congress on Acoustics, Budapest, Hungary, paper 20 C-13.
Shadle, C. H., and Atal, B. S. (n.d.). "On the Use of Pseudo-Area Parameters for Speech Synthesis by Rule " (unpublished work).
Tribolet, J. M., Rabiner, L. R., and Sondhi, M. M. (1979). "Statistical Properties of an LPC Distance Measure," IEEE Trans. Acoust. Speech Signal Process. (in press).