Interactive Clustering Techniques for Selecting Speaker-Independent Reference Templates for Isolated Word Recognition

STEPHEN E. LEVINSON, MEMBER, IEEE, LAWRENCE R. RABINER, FELLOW, IEEE,
AARON E. ROSENBERG, MEMBER, IEEE, AND JAY G. WILPON

Abstract—It is demonstrated that clustering can be a powerful tool for selecting reference templates for speaker-independent word recognition. We describe a set of clustering techniques specifically designed for this purpose. These interactive procedures identify coarse structure, fine structure, overlap of, and outliers from clusters.

The techniques have been applied to a large speech data base consisting of four repetitions of a 39 word vocabulary (the letters of the alphabet, the digits, and three auxiliary commands) spoken by 50 male and 50 female speakers. The results of the cluster analysis show that the data are highly structured containing large prominent clusters. Some statistics of the analysis and their significance are presented.

I. INTRODUCTION

RECENTLY Rabiner [11] has described a procedure for constructing composite reference templates for speaker-independent word recognition from utterances of many different speakers. An important conclusion of this study is that a few carefully constructed templates can represent a large speaker population adequately for the purpose of speaker-independent word recognition. This appears to be true because utterances of the same word by different speakers form tight clusters (in an appropriate space) representing variations in pronunciation and voice characteristics. It seems natural, then, to try to identify these clusters using sophisticated pattern recognition techniques and judiciously select tokens from them to be used as reference templates for speaker-independent word recognition.

It was decided at the outset that any clustering method which we used would have to possess certain properties. The method would have to be able to detect overlap among clusters, to identify outliers, to operate on similarity data only, and to be interactive in nature.

Overlap among clusters is an important structural feature of data. It is significant because the clusters are to be used subsequently for pattern recognition. We are, therefore, interested in knowing how many prototypes are required to represent large clusters which are actually made up of several overlapping ones. Overlap also gives a clue as to the nature of the boundaries between clusters.

The existence of outliers is also important in recognition tasks because an outlier is a potential cause of misclassification. If a token of one word is very dissimilar from all other tokens of that word, it may well be similar to some tokens of another word. It is necessary to identify such "bad data points" so that they can be eliminated or otherwise treated.

It is essential that the clustering procedures work on similarity data alone since the structure of the word recognition system (for which the clusters are to provide reference templates) is fixed. In particular, we have used the "canonical" recognition system described by Flanagan *et al.* [2] based on the cumulative, nonlinearly, time-registered distance metric proposed by Itakura [3] as a measure of word similarity. A great deal of experience has been obtained with this sytem, and it has proven robust for a variety of word recognition tasks. Unfortunately, in this system, the notion of a Cartesian feature space is lost and many of the classical clustering techniques are rendered inapplicable. This required that extensive modifications of classical procedures be made.

Finally, it was felt that it was important that the clustering procedure be interactive in order that information gleaned from one phase of clustering could be used in another. We also wanted the user to be able to bring to bear any *a priori* information about the data. We were quite certain that both situations would arise since we were intending to use independent techniques to examine the data for different structural properties.

To satisfy these diverse criteria, we selected four well-known clustering techniques and modified them where necessary to meet our requirements. The four techniques are the chainmap [9], the shared nearest neighbor (SNN) procedure, [15] the k-means iteration [8], and the ISODATA (Iterative Self Organizing Data Analysis Technique A) method of Ball and Hall [1]. Although we have greatly modified the last three of these procedures to suit our needs, we shall refer to them by their classical names since they are, in spirit, the same. In subsequent sections of this paper, we shall give the mathematical details of our versions of the procedures. Readers wishing to compare these to the original versions should consult the references.

The chainmap is a very simple procedure (which requires no user intervention) for discovering the coarse structure of the data, such as large prominent clusters. It also provides a concise picture of the data which is useful for supplying information to other more complicated procedures. An additional feature of the chainmap is that it is based on a sorted array of distances (token similarities) and is thus suitable without modification for use on our data.

Manuscript received August 23, 1978; revised October 23, 1978.
The authors are with Bell Laboratories, Murray Hill, NJ 07974.

0096-3518/79/0400-0134$00.75

The shared nearest neighbor technique is used to find any overlap among clusters. The procedure is based on the intuitively appealing notion that, if two tokens have several nearest neighbors in common, then they belong to the same cluster. A token may have common nearest neighbors with more than one token which may mean that the token belongs to distinct but overlapping clusters. Information obtained from this procedure will aid in subsequent analysis.

The k-means iteration is an automatic procedure which will find the detailed structure of the data, given information from the other procedures. The basic iteration scheme is designed to give increasingly accurate estimates of cluster center location. It was originally formulated for coordinate data; we have revised it to work on similarity data at the expense of some stability properties.

The ISODATA procedure is highly interactive and can be used to isolate outliers. Its novelty lies in its ability to merge and split clusters in order to find the "best" configuration. It uses the k-means algorithm as a subroutine and, like the k-means algorithm, was originally formulated for coordinate data. We have, therefore, made some important changes.

The four procedures are designed to be used together in the following way. First, a chainmap is computed and the number of large prominent clusters and their separation noted. Then, the SNN method is used to find which, if any, clusters overlap significantly. This enables the user to refine his estimate of the number of clusters. Next, the k-means iteration is performed based on the estimated number of clusters. These operations are all necessary preliminaries to the successful use of the ISODATA procedure, which is the most sophisticated and interactive of the four routines. Using the coarse classification results from the preliminary computations, the parameters for ISODATA can be set and a final configuration found.

The procedures have been implemented in a clustering package called VMCLUSTER, which is described in [14]. The remainder of this paper is devoted to the theory and application of the procedures. In the second section, we present the mathematical preliminaries which permit us to give the details of the procedures in the third section. Then, in Section IV, we give the results obtained in applying cluster analysis to both synthetic examples and to a large speech data base.

II. Mathematical Preliminaries

In what follows, we shall assume we are given a finite set, Ω, of N observations.

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

Bear in mind that these observations are not vectors but, rather, tokens representing spoken words. We do have, however, a distance matrix D whose ijth entry d_{ij} is a measure of the dissimilarity of the observations x_i and x_j. In our case, we have

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

where K is the number of frames in the reference utterance x_i, and $d(k, w(k))$ is the LPC distance proposed by Itakura [3] between the kth 15 ms frame of the utterance x_i and the $w(k)$th frame of the utterance x_j; that is,

$$d(k, w(k)) = \log \left[\frac{\vec{a}_{w(k)} V \vec{a}_{w(k)}^T}{\vec{a}_k V \vec{a}_k^T} \right] \tag{3}$$

where $\vec{a}_k$ is the vector of LPC coefficients associated with the kth frame of the test or unknown utterance x_i; $\vec{a}_{w(k)}$ is the vector of LPC coefficients derived from the $w(k)$th frame of the reference utterance x_j; and V is the matrix of autocorrelation coefficients computed from the kth frame of the test utterance. The function $w(k)$ is the so-called warping function and is chosen to minimize d_{ij} in (2). For a detailed discussion of the $w(k)$ function, see [12].

We will stipulate that the set of observations Ω is composed of samples of only one word and that Ω contains M (not necessarily disjoint) clusters $\{\omega_i\}_{i=1}^{M}$. The value of M is to be determined, but formally we have

$$\Omega = \bigcup_{i=1}^{M} \omega_i. \tag{4}$$

The cardinality of ω_i is denoted by m_i and its center or prototype designated $x_p^{(i)}$. Note that $x_p^{(i)} \in \omega_i$. A superscript (i) on an observation will be used to mean that the observation belongs to the ith cluster, and a subscript in square brackets denotes a nearest neighbor; thus, $x_{[k]}$ is the kth nearest neighbor to x. That is,

$$\begin{aligned} \delta(x, x_{[1]}) \leqslant \delta(x, x_{[2]}) \leqslant \cdots \leqslant \delta(x, x_{[k]}) \\ \leqslant \cdots \delta(x, x_{[N]}). \end{aligned} \tag{5}$$

Finally, for a given assignment of the N observations into a fixed number M of classes, we compute the quality measure σ according to

$$\sigma = \frac{\dfrac{1}{M(M-1)} \displaystyle\sum_{i=1}^{M} \sum_{j=1}^{M} \delta(x_p^{(i)}, x_p^{(j)})}{\dfrac{1}{M} \displaystyle\sum_{i=1}^{M} \frac{1}{m_i(m_i-1)} \sum_{j=1}^{m_i} \sum_{k=1}^{m_i} \delta(x_j^{(i)}, x_k^{(i)})}. \tag{6}$$

Thus, σ is the ratio of the average-intercluster to average-intracluster distance. For two spherically symmetric clusters, $\sigma > 2$ implies no overlap.

The most general approach to the clustering problem is to treat the observations as being produced by a single random process of density function $F(z)$ which is the finite mixture

$$F(z) = \sum_{i=1}^{M} f(z|\omega_i, \vec{b}_i) P(\vec{b}_i|\omega_i). \tag{7}$$

In (7), the $f(z|\omega_i, \vec{b}_i)$ are the ith cluster-conditional density functions which are characterized by the parameter vectors $\vec{b}_i$. $P(\vec{b}_i|\omega_i)$ is the probability of the vector $\vec{b}_i$, given that it characterizes the ith cluster. This term is simply a weight which tells how much the ith conditional density function contributes to the overall density in a small neighborhood of z. In the case that $f(z|\omega_i, \vec{b}_i)$ contributes most strongly to $F(z)$, then the points in a neighborhood about z belong, with high probability, to ω_i.

For example, assume $F(z)$ to be a weighted sum of Gaussian density functions of different means and covariance matrices.

The vectors $\vec{b}_i$ would be these very parameters. Such an $F(z)$ corresponds to elliptical clusters centered about the means and with half axes of length equal to the eigenvalues of the respective covariance matrices.

If, as in the above example, it is known *a priori* that the conditional density functions are all members of the same family differing only in the parameter vector $\vec{b}_i$, then one can solve for the $\vec{b}_i$ in the mixture problem of (7) directly and the solution is unique up to a renaming of the clusters. In the absence of such information, however, one must use some technique for estimating $F(z)$.

The most obvious method for estimating $F(z)$ is to find the optimal assignment of tokens to classes with respect to some quality measure. In our notation, we would like to maximize σ in (6) over all assignments of the x_j to an ω_i for $1 \leqslant j \leqslant N$ and $1 \leqslant i \leqslant M$. For even moderate values of M and N, however, the optimization is computationally intractable since the number of distinct ways that N objects can be assigned to M nonempty classes is given by the Stirling numbers of the second kind [5], $S_2(M, N)$, where

$$S_2(M,N) = \frac{1}{M!}\sum_{j=0}^{M}(-1)^{M-j}\binom{M}{j}j^N. \tag{8}$$

Clearly, $S_2(M, N)$ grows exponentially with both M and N. Branch and bound techniques, such as those of Koontz *et al.* [4], lessen the computational burden only slightly.

Finally, then, we are left to estimate $F(z)$ by some robust but necessarily suboptimal technique. Many such procedures are discussed in the literature. They most often require that the observations be vectors in some feature space. Since our data are not in that form, we have selected procedures which are amenable to similarity data or can be modified to be such. The next section gives the details of the clustering procedures we have used.

III. Procedures

In this section, we give the mathematical details of four clustering procedures: the chainmap, the shared nearest neighbor method, the k-means iteration, and ISODATA.

A. The Chainmap

One of the major difficulties in performing a cluster analysis on a large data base is just getting started, getting some insight into the gross structure of the data. The chainmap is a very simple analysis technique which gives its output in the form of a two-dimensional plot from which a surprising amount of information can be obtained.

The first step in creating the chainmap is to reorder the data. We designate an arbitrary token as the start of the chain; let us call it x_s. The next token in the sequence will be the nearest neighbor to x_s which is $x_{s[1]}$. In general, the k + 1st element of the ordered list will be $x_{k[1]}$ where it is understood that the nearest neighbor of x_k is selected from amongst the, as yet, unordered tokens. The process is continued until we have ordered all observations, at which point we have constructed the sequence

$$x_{i_0} \leqslant x_{i_1} \leqslant x_{i_2} \leqslant \cdots \leqslant x_{i_{N-1}}; \quad i_0 = s. \tag{9}$$

We then associate with the kth member of this sequence the distance d_k where

$$d_k = \delta(x_{i_{k-1}}, x_{i_k}) \qquad 1 \leqslant k \leqslant N-1. \tag{10}$$

The chainmap is simply a plot of d_k against k.

The salient features of the plot are the large spikes, each corresponding to a cluster boundary. The prominence of these peaks is indicative of the distinctiveness of the clusters as is the relative smoothness of the rest of the plot. Typically, the chainmap is quite noisy, but well separated clusters usually show up. The procedure is somewhat sensitive to the choice of starting point x_s but is computationally simple so that several starting points can be tried for little additional cost.

B. A Method Based on Shared Nearest Neighbors

The SNN procedure is based on the simple notion that two tokens, which have at least some number k_s of common nearest neighbors, belong in the same cluster. We have formalized this idea in the following way. Let L be the nearest neighbor list

$$L = \begin{bmatrix} x_1 & x_{1[1]} & x_{1[2]} & \cdots & x_{1[k]} \\ x_2 & x_{2[1]} & x_{2[2]} & \cdots & x_{2[k]} \\ x_3 & & & & \\ \vdots & & & & \\ x_N & x_{N[1]} & x_{N[2]} & \cdots & x_{N[k]} \end{bmatrix} \tag{11}$$

in which there are N rows corresponding to the N tokens and the ith row R_i is an ordered list of the tokens which are the k nearest neighbors to x_i.

Now suppose that $x_i \in R_j$ and $x_j \in R_i$ and suppose further that

$$|R_i \cap R_j| \geqslant k_s \tag{12}$$

for some fixed threshold k_s. Then x_i and x_j share at least k_s neighbors, including the tokens themselves and thus are assigned to the same class.

It is, of course, possible that x shares a set of k_i nearest neighbors with x_i, and k_j nearest neighbors with x_j, with $k_i + k_j \leqslant k$ and $k_i > k_s$ and $k_j > k_s$. Then x belongs in both the ith and jth clusters or ω_i has a nonempty intersection with ω_j. Thus, this procedure can be used to identify overlapping clusters. Even if several clusters overlap, this method will discover it.

In practice, we place some upper limit $l_{\max}$ on the number of clusters to which a token may belong. Of course the overall results will depend upon the selection of the values of k, k_s, and $l_{\max}$. However, the computation is simple and requires no interaction other than choosing the parameters so that one can easily experiment with several choices.

C. The k-Means Iteration

The k-means procedure is an automatic iteration scheme which will quite reliably find any specified number of clusters. The iteration consists of three basic steps: classification, computation of cluster centers, and convergence testing.

Assuming that we wish to find M clusters, we choose M arbitrary tokens to serve as initial cluster centers. For simplicity, we set

$$x_p^{(i)} = x_i \quad \text{for} \quad 1 \leqslant i \leqslant M. \tag{13}$$

Classification then proceeds on the basis of the nearest neighbor rule, namely,

$$x_j \in \omega_i \text{iff } \delta(x_j, x_p^{(i)}) \leqslant \delta(x_j, x_p^{(k)}) \quad 1 \leqslant k \leqslant M. \tag{14}$$

After (14) has been applied for $1 \leqslant j \leqslant N$, we recompute the cluster centers using a minimax criterion. That is, we let

$$x_p^{(i)} = x_j^{(i)} \quad \text{such that} \quad \max_k \{\delta(x_j^{(i)}, x_k^{(i)})\} \tag{15}$$

is minimized for $1 \leqslant i \leqslant M$.

The convergence test consists of checking whether or not the same tokens are designated as cluster centers as in the previous iteration. If not, another iteration is performed.

Even when the k-means procedure is used on coordinate data, convergence is not guaranteed. When the cluster centers are computed according to (15), oscillation between two configurations may occur. This is particularly true if the value chosen for M does not agree with the structure of the data. However, we have not observed this problem often and generally convergence does occur. As we shall see in the next section, however, the k-means procedure is best used in conjunction with an interactive scheme.

D. ISODATA

In order to bring the human into the process, to identify outliers, and to determine the actual number M of clusters, we have implemented an ISODATA procedure. The novelty of this technique lies in its ability to split and merge existing clusters in order to change M and increase σ. Furthermore, this permits the isolation of outliers which will occur when a single point is split from a cluster.

The principal part of ISODATA is the k-means procedure, but the number of clusters is adjusted at each iteration according to criteria based on a number of fixed and variable thresholds.

Clusters are merged if one or more of the following conditions occurs: 1) the present number of clusters M exceeds some threshold value $M_{\max}$. 2) The size of the ith cluster $|\omega_i|$ becomes less than a threshold value $m_{\min}$. 3) The distance between the ith and jth cluster centers $\delta(x_p^{(i)}, x_p^{(j)})$ is less than some threshold θ_m.

If $M > M_{\max}$, then the two closest clusters are merged. If $|\omega_i| < m_{\min}$, then ω_i is merged with the cluster nearest to it. If one wishes to isolate outliers, $m_{\min}$ must be set to unity and a merge will never result from criterion 2). The most general merging condition results when

$$\delta(x_p^{(i_1)}, x_p^{(j_1)}) \leqslant \delta(x_p^{(i_2)}, x_p^{(j_2)}) \leqslant \cdots \leqslant \delta(x_p^{(i_L)}, x_p^{(j_L)}) < \theta_m. \tag{16}$$

In which case the L pairwise merges will be performed according to

$$\omega_k^* = \omega_{i_k} \cup \omega_{j_k} \quad 1 \leqslant k \leqslant L \tag{17}$$

so that the actual merging operation is just the common set theoretic union. If more than one merge takes place in any iteration, then criterion 3) must be rechecked to see whether more than two clusters should have been combined.

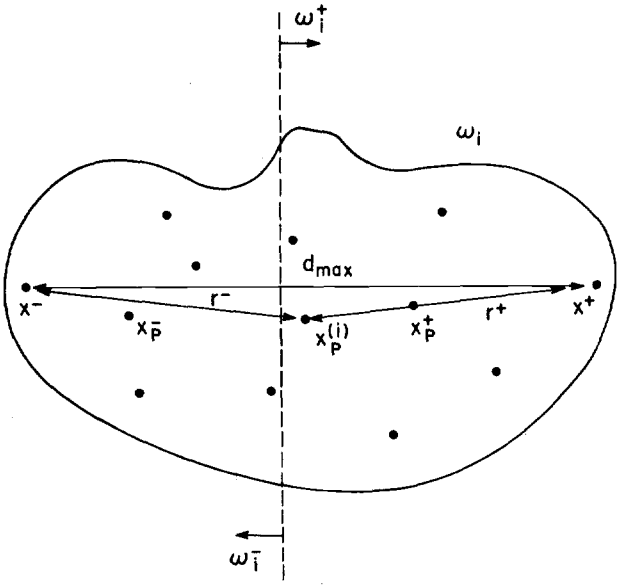

Fig. 1. Two splitting procedures for ISODATA.

There are also three conditions under which clusters are split: 1) The present number of clusters M becomes less than some preassigned value $M_{\min}$. 2) The size of the ith cluster $|\omega_i|$ exceeds some threshold $m_{\max}$. 3) The ith cluster becomes too sparse relative to the other clusters.

The first two criteria are similar to the corresponding ones for merging. The third is slightly more intricate. To test a configuration against the third criterion, we first compute an intracluster distance D_i for each cluster from

$$D_i = \frac{1}{m_i - 1} \sum_{x \in \omega_i} \delta(x, x_p^{(i)}) \quad 1 \leqslant i \leqslant M. \tag{18}$$

We then form an average-intracluster distance $\bar{D}$ according to

$$\bar{D} = \frac{1}{M} \sum_{i=1}^{M} m_i D_i. \tag{19}$$

Then, ω_i will be split if

$$D_i > \max \{\bar{D}, \theta_s\} \tag{20}$$

for some splitting threshold θ_s.

When the ith cluster is to be split, it is divided into two parts so that

$$\omega_i = \omega_i^+ \cup \omega_i^-. \tag{21}$$

The actual procedure by which (21) is implemented is somewhat untidy because we have only the distances between tokens available to us. We use two different methods which are both shown in Fig. 1. In the simpler case, we find the two points x^+ and x^- such that $\delta(x^+, x^-)$ is maximum. Then, each point in ω_i is assigned to either ω_i^+ or ω_i^- if its distance to x^+ or x^-, respectively, is smaller. In this case, x^+ and x^- are not very good center points for the new clusters.

An alternative which gives better estimates of the new center points is the following. As before we locate x^+ and x^-, we then set

$$r^+ = \delta(x^+, x_p^{(i)}) \tag{22}$$

and

$$r^- = \delta(x^-, x_p^{(i)}) \tag{23}$$

and find x_p^+ and x_p^- such that

$$\epsilon^+ = \delta(x^+, x_p^+) + \delta(x_p^+, x_p^{(i)}) - r^+ \tag{24}$$

and

$$\epsilon^- = \delta(x^-, x_p^-) + \delta(x_p^-, x_p^{(i)}) - r^- \tag{25}$$

are minimized. As before, the members of ω_i are assigned to ω_i^+ or ω_i^- on the basis of their proximity to x_p^+ and x_p^-.

In this case, x_p^+ and x_p^- are much better centers for ω^+ and ω^-. We have used both splitting procedures in practice and there seems to be little difference in the final configuration. It is clear, however, that one could construct cases for which one or the other would be markedly better. We have, therefore, included both procedures in our version of ISODATA.

In light of the foregoing discussion, the ISODATA procedure is easily described. First, cluster centers are found according to (15) and the remaining points are assigned to clusters according to (14). These steps are actually part of the k-means iteration. Following classification clusters are merged and split according to (16), (17), and (18)-(25), respectively. After merging and splitting are complete, the clusters are renamed to account for new clusters generated by splitting and old ones lost in merging. Finally, we check for convergence in two ways. We either use the convergence criterion of the k-means iteration or we may stop whenever the current configuration has a σ ratio which is greater than some threshold η.

It may appear at first glance that ISODATA is very difficult to use because of the seven parameters, $M_{\max}$, $M_{\min}$, $m_{\max}$, $m_{\min}$, θ_m, θ_s, and η, which must be set. Actually these parameters afford the procedure great power and flexibility. With the information obtained from preliminary applications of chainmap and SNN, and from previous iterations of ISODATA, one gets a good grasp of the data which usually results in a good cluster configuration. Reference [14] contains a section on "helpful hints" for actually using ISODATA.

IV. Experimental Results

In this section, we give some results of application of the clustering techniques described above. We first give some illustrative examples obtained by clustering synthetic data. From these, the reader can get an intuitive idea of how the procedures perform. We then move on to the major results which were obtained in clustering a large speech data base.

A. Results on Synthetic Data

Figs. 2 and 3 show the results from the chainmap procedure. Fig. 2 is a plot of two-dimensional synthetic data consisting of points drawn from a mixture of four normal distributions. The chainmap for this data is shown in Fig. 3.

Fig. 4 shows the results of a similar experiment with the SNN routine. The data, which were analyzed in this case, were drawn in equal proportions from two bivariate normal distributions whose means and variances were chosen so that there would be substantial overlap. In Fig. 4, those points that were judged by the SNN procedure to be in the overlap regions are plotted with a separate symbol. The actual means and contours of equal variance are superimposed on the plot. The cluster configuration was obtained with $k = 15$ and $k_s = 6$.

Table I shows the results of clustering the data of Fig. 2 with the k-means iteration in which M has been set to 4. The actual means and variances of the cluster distributions are compared with those estimated from the final cluster configuration. It is

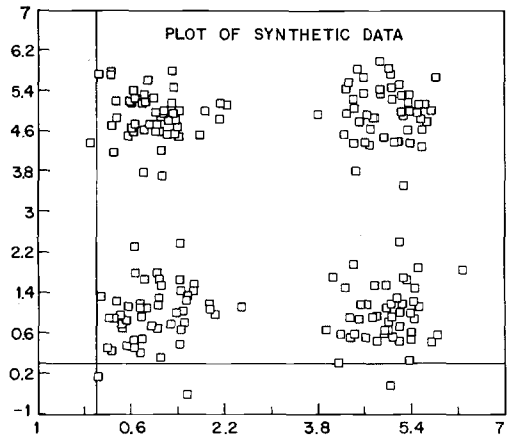

Fig. 2. Synthetic test data.

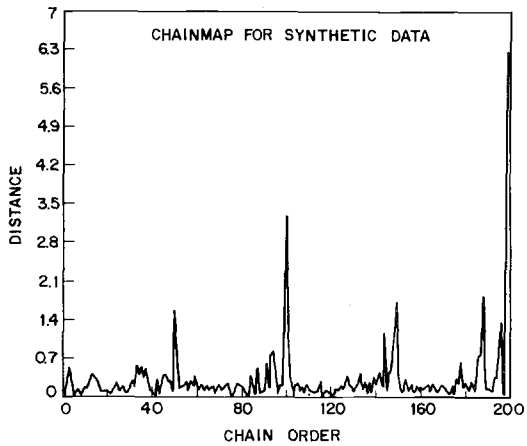

Fig. 3. Chainmap of synthetic test data.

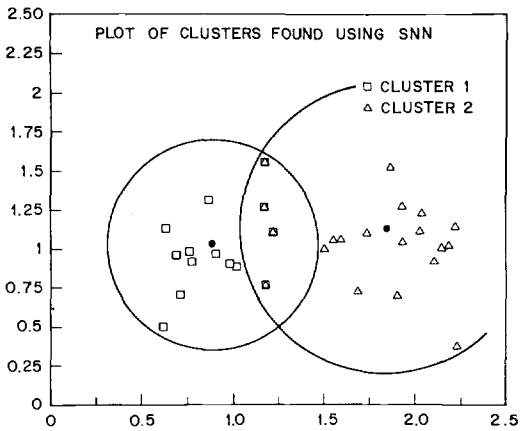

Fig. 4. Clusters in synthetic data found by the SNN method.

TABLE I
OUTPUT OF *k*-MEANS ITERATION FOR SYNTHETIC GAUSSIAN DATA

CLUSTER	MEANS		VARIANCES	
	ACT.	EST.	ACT.	EST.
1	1,1	1.0, .99	.25, .25	.30, .33
2	1.5	.97, 4.9	.25, .25	.26, .21
3	5,5	4.96, 4.98	.25, .25	.22, .27
4	5,1	4.99, .98	.25, .25	.23, .28

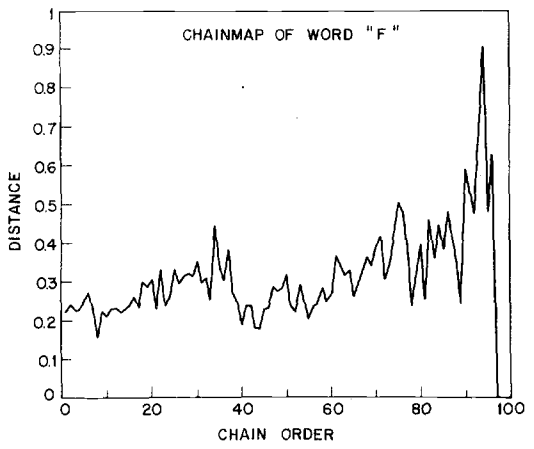

Fig. 5. Chainmap of speech data.

clear from this example in what sense clustering techniques are heuristic and suboptimal, unsupervised density estimators.

B. Results for Speech Data

A good feeling for the quality of the speech data can be seen from the chainmap of Fig. 5. This is a plot of the chainmap for the word "*F*." Note that there are three prominent clusters and relatively good separation. This is typical of the kind of information about a set of data which is obtained from chainmap.

The ISODATA procedure was tested on a small subset of the large speech data base which it was our object to analyze. Before giving the results of that test and the complete study, we will briefly describe the data and the procedure used to collect it.

Our data consists of four replications of a 39 word vocabulary composed of the alphabet, the digits, and three "command" words (stop, error and repeat) spoken by 50 male and 50 female native speakers of English. The vocabulary is difficult yet very useful for word recognition (see Rosenberg and Schmidt [13]).

Since the data was ultimately to be used as training data for a recognition system, every effort was made to insure that artifacts were not present. Speech from a standard telephone handset located in a sound booth was recorded on analog tape. The tape recordings were bandpass filtered (200-3200 Hz) and digitized at 6.67 kHz. After semiautomatic endpoint detection in which the operator had audio playback and an energy envelope display upon which to base his/her judgments, the set of eighth-order autocorrelation coefficients [3] for each 45 ms frame was stored on disk. In the final stage, a distance matrix for each vocabulary word was computed according to

TABLE II
OUTPUT OF THE ISODATA FOR 18 UTTERANCES OF THE WORD "*A*"

CLUSTER	SIZE	TOKENS	M/F	CENTER	INTRACLUSTER DISTANCE
1	4	1,6,10,15	4/0	15	.202
2	3	3,4,13	3/0	4	.173
3	2	5,14	2/0	14	.100
4	2	7,16	0/2	16	.183
5	4	8,9,17,18	0/4	17	.285
6	3	2,11,12	3/0	2	.129

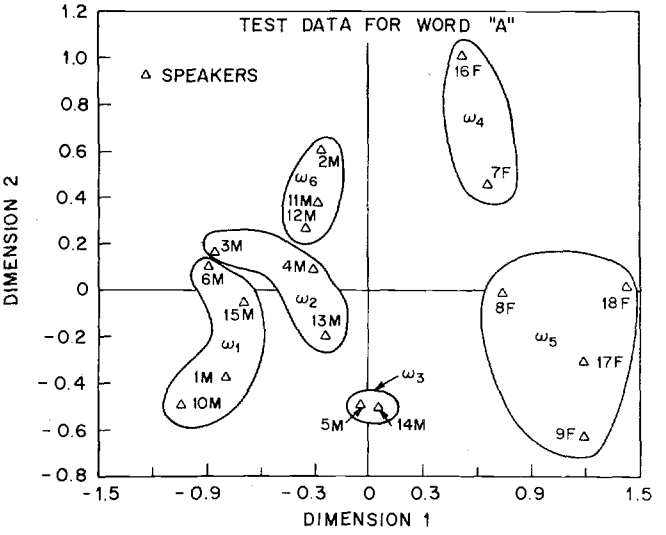

Fig. 6. Pilot study with ISODATA.

(2) and (3) using the 100 utterances from the first replication. These 39 matrices were the input to the clustering procedures. Only one matrix corresponding to a single vocabulary word was analyzed at a time.

Before analyzing the entire data base, several pilot studies were done. We shall describe one of the simpler ones in which we analyzed 18 utterances of the word "*A*" spoken by twelve males and six females. The 18 X 18 distance matrix was input to the ISODATA procedure. The results, which are shown in Table II, indicate very strong structure in the data in the form of six clusters. Four of the clusters contain only male utterances and two more, only female. The sigma ratio of 3.2 shows strong separation.

In order to corroborate these results, we input the same distance matrix to the multidimensional scaling program KYST-II [6]. This program finds a configuration of points in an *n*-dimensional Euclidean space such that the pairwise distances between points match the corresponding entries in the input distance matrix optimally with respect to a goodness-of-fit criterion. This procedure finds structure in the data of a different kind than is identified by clustering. After scaling, each dimension quantifies the presence of some abstract feature. Ideally, these abstract features can be correlated to physically meaningful measurements. The reader interested in the relationship of multidimensional scaling to clustering should consult Kruskal [7]. Part of the output is shown in Fig. 6 which

TABLE III
CLUSTERING RESULTS ON SPEECH DATA

Word	No. of Clusters	No. of Outliers	σ Ratio	Size of Largest Cluster
A	12	4	3.1	35
B	10	5	3.1	25
C	11	3	2.6	21
D	10	6	3.5	20
E	10	6	3.9	19
F	6	4	2.5	32
G	13	3	3.3	22
H	8	4	2.5	32
I	12	7	2.9	26
J	10	6	3.2	36
K	11	5	3.1	33
L	9	9	2.6	28
M	13	7	2.6	28
N	13	8	2.6	20
O	12	11	2.9	28
P	12	7	3.4	19
Q	15	8	3.2	17
R	13	11	3.6	22
S	10	7	2.6	29
T	10	12	3.7	30
U	17	7	3.3	12
V	18	3	3.1	24
W	15	9	2.5	23
X	10	11	2.5	23
Y	12	12	3.0	20
Z	15	10	3.4	18
STOP	12	14	2.6	31
ERROR	15	15	3.1	31
REPEAT	19	8	3.0	16
0	19	11	2.4	17
1	17	5	2.4	17
2	16	6	3.0	33
3	12	14	3.5	19
4	18	7	3.0	28
5	13	10	2.9	20
6	11	16	2.9	22
7	16	15	2.9	17
8	14	8	3.3	20
9	14	14	3.1	19

is a plot of dimension 1 versus dimension 2, with the axes rotated to principal components. The plot is one plane of a six-dimensional configuration having a stress of 0.012. This means that six features account for more than 98 percent of the variation in the data. This also shows that the data are highly structured. A further verification of the clustering results is that dimension 1, which is the strongest contributor to the variation in the data, is the male/female axis.

The other pilot studies include an 18 speaker set of the word "*B*" and 40 speaker sets of the words "*W*" and "Error." All were clustered and scaled as described above. The results were consistent with those for the word "*A*."

In addition, we clustered a 100 token set consisting of 10 utterances (males and females together) of each of 10 words. The clusters showed very strong separation of words compared to the male/female dichotomy of the intraword separation.

Finally, we attempted to learn more about the significance of the six-dimensional space obtained from KYST-II. This was done using the individual differences scaling technique SINDSCAL [10]. This cursory study showed that only a single dimension was common amongst words and that was the male/female axis.

The pilot studies gave us some experience in using our clustering package, and their outcome convinced us that the data was highly structured. We, therefore, conducted an analysis of each of the 39 words using the entire 100 speaker set. The results of these runs are summarized in Table III. In brief, we found clusters ranging in number from 6 for the word *F* to 19 for the word "repeat." The largest clusters contained a minimum of 12 and a maximum of 36 utterances. There were as few as 3 and as many as 16 outliers for a single word. The value of the quality ratio was lowest at 2.4 for the word "one" and was highest at 3.9 for the word "*E*."

V. SUMMARY

We have discussed the theory underlying a set of clustering procedures for the purpose of selecting reference templates for speaker-independent word recognition. We have given examples of the performance of these procedures on synthetic and speech data and we have given the results of applying the procedures to a large speech data base. In analyzing the speech data base, we found that tokens for each word are arranged in well-defined clusters with a few outliers. Work on the application of this structure to a speaker-independent word recognition system will be described in subsequent papers.

REFERENCES

[1] G. H. Ball and D. J. Hall, "Isodata—An iterative method of multivariate analysis and pattern classification," in *Proc. IFIPS Congr.*, 1965.

[2] J. L. Flanagan, S. E. Levinson, L. R. Rabiner, and A. E. Rosenberg, "Techniques for expanding the capabilities of practical speech recognizers," in *Trends in Speech Recognition*, W. Lea, Ed. Englewood Cliffs, NJ: Prentice-Hall, to be published.

[3] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 67-72, Feb. 1975.

[4] W. L. G. Koontz, P. M. Narendra, and K. Fukunaga, "A branch and bound clustering algorithm," *IEEE Trans. Comput.*, vol. C-24, pp. 908-915, Sept. 1975.
[5] D. E. Knuth, *The Art of Computer Programming, Vol. I: Fundamental Algorithms.* Reading, MA: Addison-Wesley, 1968.
[6] J. B. Kruskal, F. W. Young, and J. W. Seery, "How to use KYST-II; A very flexible program to do multidimensional scaling and unfolding," Computing Information Service, Bell Labs.
[7] J. B. Kruskal, "The relationship between multidimensional scaling and clustering," in *Classification and Clustering*, J. Van Ryzin, Ed. New York: Academic, 1977.
[8] J. MacQueen, "Some methods for classification and analysis of multivariate data," in *Proc. 5th Berkeley Symp. Probability and Statistics*, Berkeley, CA, 1967.
[9] E. A. Patrick, *Fundamentals of Pattern Recognition.* Englewood Cliffs, NJ: Prentice-Hall, 1972.
[10] S. Pruzansky, "How to use SINDSCAL—A computer program for individual differences in multidimensional scaling," Computing Information Service, Bell Labs.
[11] L. R. Rabiner, "On creating reference templates for speaker independent recognition of isolated words," *IEEE Trans. Acoust. Speech, Signal Processing*, vol. ASSP-26, pp. 34-42, Feb. 1978.
[12] L. R. Rabiner, A. E. Rosenberg, and S. E. Levinson, "Considerations in dynamic time warping algorithms for discrete word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 575-583, Dec. 1978.
[13] A. E. Rosenberg and C. E. Schmidt, "Directory assistance by means of automatic recognition of spoken spelled names," in *Proc. IEEE ICASSP-78*, Tulsa, OK, Apr. 1978.
[14] J. G. Wilpon, and S. E. Levinson, "How to use the clustering package VMCLUSTER," Bell Labs. Tech. Memo, Aug. 1978.
[15] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *IEEE Trans. Comput.*, vol. C-22, pp. 1025-1034, Nov. 1973.

A Two-Sided Rational Approximation Method for Recursive Digital Filtering

CHARLES K. CHUI AND ANDREW K. CHAN, MEMBER, IEEE

Abstract—A two-sided rational approximation procedure for recursive digital filter design is presented in this paper. More specifically, an explicit expression for the designed filter transfer function can be obtained once a rational approximation for the analytic part of the Fourier series expansion of the desired filter characteristic is determined. To demonstrate the efficiency of this technique, we derive a two-sided Padé approximation method. Several examples are given to illustrate this design procedure.

Introduction

In the area of recursive digital filter design, there are many techniques available in the literature. See, for example, [10] and [11]. However, only a few of these methods give explicit expressions of the realizable transfer functions $H(z^{-1})$. In particular, the Padé approximant method [1], [2], [8] was introduced to approximate the truncated delayed Fourier expansions of the ideal filter characteristics $H(\omega)$. The advantages and disadvantages of the Padé approximant method and the extension of it were discussed in [2] and [8]. All these techniques are for approximation of a (formal) power series. We call this a one-sided rational approximation. To obtain a power series, one usually truncates a Fourier expansion and then introduces a constant delay. The main disadvantage of this transformation is that the lower order terms (which are the most important ones) of the new power series no longer represent the low-frequency terms of the original Fourier expansion. This might create a serious problem, especially in designing maximally flat characteristic filters. This phenomenon will be discussed further in the next section. The main contribution of this paper is to introduce a *two-sided rational approximation* of $H(\omega)$, that is, we will derive a method to approximate (the Fourier expansion of) $H(\omega)$ directly by a rational function $H_a(z^{-1})$ in $z = e^{j\omega}$.

Manuscript received April 11, 1978; revised August 24, 1978.

C. K. Chui is with the Department of Mathematics, Texas A&M University, College Station, TX 77843.
A. K. Chan is with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843.

A Two-Sided Rational Approximation Method

In the Fourier expansion of a given ideal filter characteristic

$$H(\omega) = \sum_{n=-\infty}^{\infty} c_n e^{-jn\omega}$$

the most representative terms in the series are the terms $c_n e^{-jn\omega}$ for small $|n|$. Hence, in any approximation method, it is intuitively clear that the weight of approximation should be given to $c_0, c_{\pm 1}, c_{\pm 2}, \cdots$, in this order. This is particularly so in Padé approximation.

A rational function $r(z^{-1}) = P_m(z^{-1})/Q_n(z^{-1})$, where P_m and Q_n are polynomials of degrees m and n, respectively, is called a Padé approximant of a (formal) power series $f(z^{-1}) = a_0 + a_1 z^{-1} + \cdots$ if its Maclaurin expansion agrees with as

0096-3518/79/0400-0141$00.75