Considerations in Applying Clustering Techniques to Speaker Independent Word Recognition

L. R. Rabiner

J. G. Wilpon

Acoustics Research Department
Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT

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 [1,2]. 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 [3] 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 shows the first method to be superior to the second method when 3 or more clusters per word are used in the recognition task.

I. 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 [1,2]. 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 minutes 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. 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 artificially created token).
3. Differences between averaging different feature sets to give word templates from clustered data.

II. Unsupervised Algorithms for Clustering Word Data

Following the development in Ref. 1, we assume that we are given a finite set, Ω, of N observations

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

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.

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

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

where the local frame distance $d(k, w(k), i, j)$ is the log likelihood distance proposed by Itakura [4] between the k^{th} frame of x_i and the $w(k)^{th}$ frame of x_j, i.e.

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

where $\underline{a}^i_l$ is the vector of LPC coefficients of the l^{th} frame of token i, R^i_k is the matrix of autocorrelation coefficients of the k^{th} 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 $w(k)$.

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

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

Eq. (4) yields a symmetric distance matrix $(d_{ij} = d_{ji})$ requiring storage for only $N(N-1)/2$ terms (since $d_{jj} = 0$ 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 or prototype of cluster ω_i as $\hat{x}_i$ and we note that $\hat{x}_i$ need *not* be a member of ω_i.

2.1 Unsupervised Clustering Without Averaging (UWA)

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.

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

CH1379-7/79/0000-0578$00.75 © 1979 IEEE

$$= \{x_1', x_2', \ldots x_{q(j)}'\} \qquad (7)$$

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 $q(0) = N$.)

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

   $$\hat{x}_{j+1} = x_{i*}' \ni \max_j \delta(x_{i*}', x_j') \leqslant \min_i \max_j \delta(x_i', x_j') \qquad (8)$$

   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

   $$\omega_{j+1}^{(k)} = \bigcup_{i \in \Omega_{j+1}'} x_i' \ni \delta(\hat{x}_{j+1}, x_i') \leqslant T \qquad (9)$$

   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 $\omega_{j+1}^{(k)} = \omega_{j+1}^{(k-1)}$ or if $k > KMAX$, a user supplied iteration check. If either is true the j^{th} 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.

The UWA algorithm is fully automatic and can cluster a set of 100 observations into from 10 to 25 clusters in about 1 minute. 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.

$$\omega_i = \{x_1', x_2', \ldots, x_J'\} \qquad (10)$$

We define the L_p norm of ω_i as

$$x_{LP} = \frac{1}{J}\left[\sum_{j=1}^{J} (x_j')^p\right]^{1/p} \qquad (11)$$

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 $p = 1$.

2.2 Unsupervised Clustering with Full Averaging (UFA)

The second unsupervised clustering algorithm we have considered is one which attempts to find clusters in the vicinity of the averaged center of the current observation set.

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. The UFA algorithm is considerably slower than the UWA algorithm since all distances must be computed as needed. Typically it takes about 1 hour 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.

III. 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 [2]. 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

C3- Supervised algorithm of Reference [1], 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.

The 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 9 procedures (along with a randomly chosen set of templates) on the first 3 test sets discussed in Reference 2. Recognition accuracies 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.

Results of the recognition tests are shown in Figures 1-4. Figures 1-2 show plots (for TS1 data) of the recognition accuracy as a function of p for $c = 1$ (part a), $c = 2$ (top two candidates - part b) and $c = 5$ (top five candidates - part c). The decision rule for recognition is the KNN rule discussed in Reference 2 in which KNN = 1 for small values of p and KNN = 2 or 3 for p larger than about 4. Figure 1 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 2 or more templates per word, the recognition accuracies of both procedures are comparable. Figure 1 also shows significantly poorer recognition accuracy from the randomly chosen templates than from either clustering approach.

Figure 2 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 4 procedures. 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 Figures 1 and 2, 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.

Comparisons of the recognition accuracies for the full averaging procedures - C2R, C2G, C2P show that the averaging of autocorrelation coefficients provided consistently better results than the averaging of log areas or arcsin PARCOR's. However, except for $p = 1$, it was found 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 vocabulary, the fully averaging clustering procedure would yield the best results.

Figures 3 and 4 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 4 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 5 candidates.

IV. 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.

References

1. S. E. Levinson, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "Interactive Clustering Techniques for Selecting Speaker Independent Reference Templates for Isolated Word Recognition," IEEE Trans. on Acoustics, Speech and Signal Proc., 1979 (to appear).
2. L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker Independent Recognition of Isolated Words Using Clustering Techniques," submitted for publication.
3. F. Itakura, "Minimum Prediction Residual Applied to Speech Recognition, "IEEE Trans. Acoustics, Speech, and Signal Proc., Vol. ASSP-23, pp. 67-72, Feb. 1975.

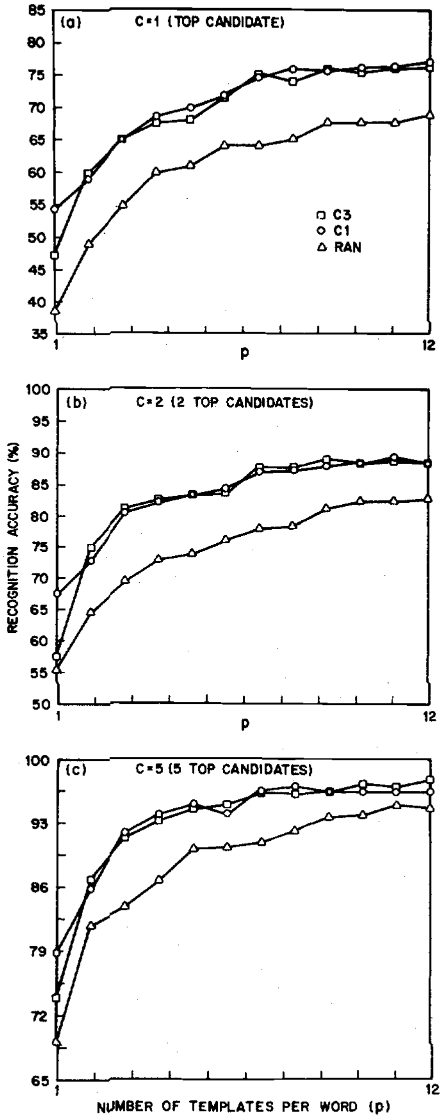

Fig. 1. Recognition accuracy as a function of p for the data of TS1 for $c = 1$ (part a), $c = 2$ (part b), and $c = 5$ (part c) for the C1, C3, and RAN clustering procedures.

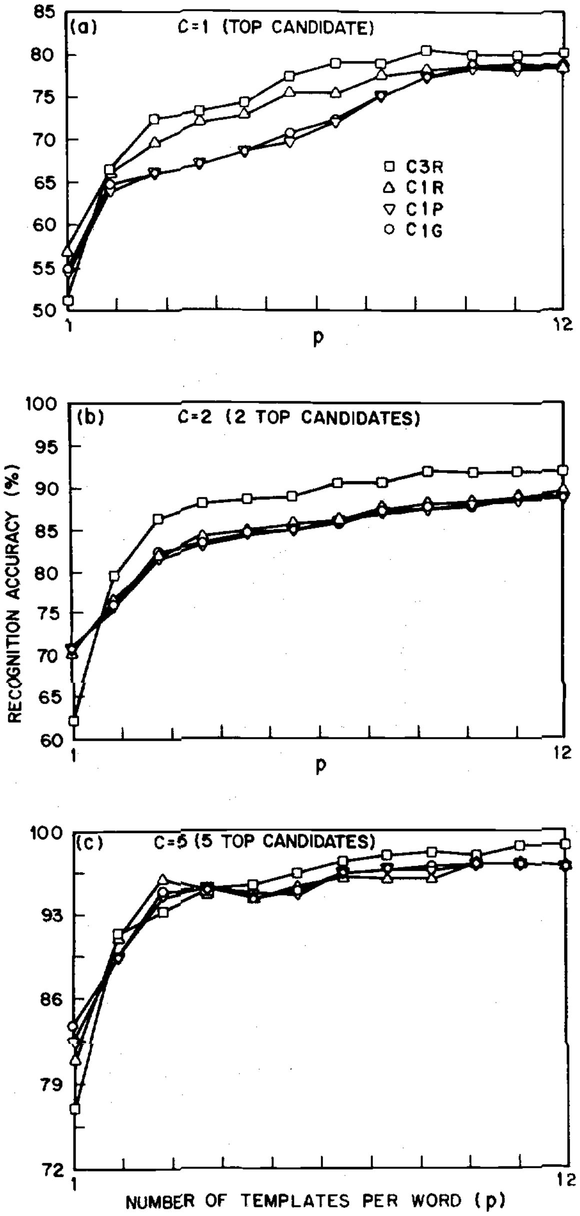

Fig. 2. Recognition accuracy as a function of p for the data of TS1 for $c = 1$ (part a), $c = 2$ (part b), and $c = 5$ (part c) for the C1R, C1G,C1P and C3R clustering procedures.

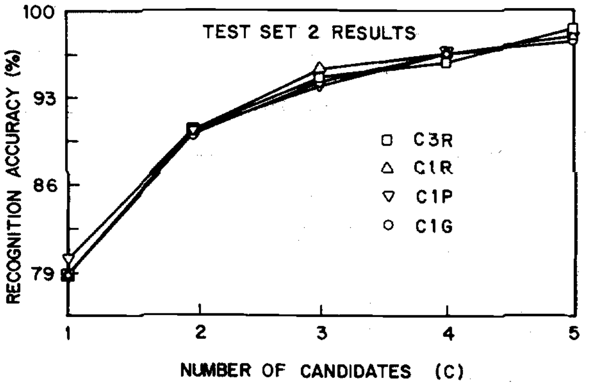

Fig. 3. Recognition accuracy as a function of c for the data of TS2 for the C1R, C1G, C1P and C3R clustering procedures.

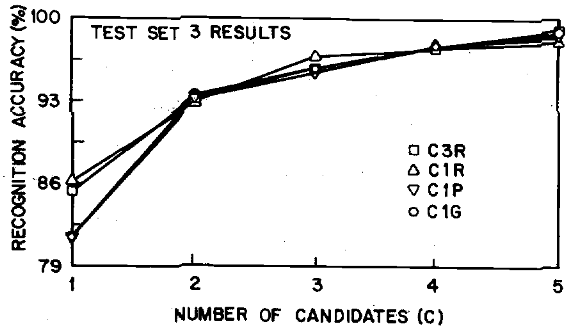

Fig. 4. Recognition accuracy as a function of c for the data of TS3 for the C1R, C1G, C1P and C3R clustering procedures.