NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

A NOVEL PATTERN LEARNING AND RECOGNITION PROCEDURE APPLIED TO THE LEARNING OF VOWELS

John Burge and Frederick Hayes-Roth
Computer Science Department[†]
Carnegie-Mellon University
Pittsburgh, Pa. 15213

January 16, 1976

ABSTRACT

The ability of a set of simple predicates to capture characteristic patterns in a parametric representation of vowels in continuous speech was investigated with the aid of an efficient conjunctive pattern recognition and classification system. The results compare favourably with those produced by a cluster-based minimal Euclidean distance technique, run over the identical training and test samples. The predicates used are similar to auditory receptive fields.

1 General Introduction

One of the most challenging problems in the construction of Speech Understanding Systems[1] is that of finding an inexpensive but accurate characterization of phones in terms of the initial, parametric, representation of the speech wave. A good labeller would reduce the amount of effort expended by the complex heuristic knowledge sources (such as those for syntax and semantics) when they are required to reduce the uncertainty due to poorly performing bottom-up phone and word recognizers. Regrettably, variability in the realization of phones in continuous speech militates against any simple-minded approach to labelling and no method has yet been forthcoming which is both accurate and inexpensive at run-time. It is our belief,

[†] This research was supported in part by the Defense Advanced Research Projects Agency under contract no. F44620-73-C-0074 and monitored by the Air Force Office of Scientific Research.

HUNT LIBRARY
CARNEGIE-MELLON UNIVERSITY

however, that the general but simple conjunctive pattern learning and classification technique and feature-description methodology described below has the ability to deal effectively with such difficult task domains. The phone recognition task is a hard problem which provides a good opportunity to develop the method in a domain subject to a high degree of variability. Moreover, the solution will be of immediate practical use.

The approach to be described is one of finding pattern templates which are conjunctive abstractions of simple feature descriptions of training examples from each class. A conjunctive abstraction is the intersection of the sets of features (or predicates) used to describe several training examples from the same class. Because the abstraction represents properties which are true of all the exemplars from which is derived, it may be used as a template for pattern classification. Any test item exhibiting all the features in the templates is assigned to the same class as the examples from which it was abstracted. The abstractions are found by a procedure called SLIM (for Space Limited Interference Matching)[2]. This procedure finds maximal conjunctive abstractions in feature-spaces, and has been implemented as an interactive system in such a way as to facilitate the exploration of different feature representations. The technique of feature encoding developed in the course of this study appears to have a general utility in conjunctive pattern learning. Furthermore, there are parallels with physiological feature encoding and it may provide a bridge between natural and artificial speech processing methodologies.

The remainder of this paper is organized as follows. First we describe SLIM and the feature-encoding method. We then describe the data and their feature representation and the classification experiment. The results obtained are then compared with those from a labelling system in current use which employs a Euclidean distance technique applied to the same training and test examples.

We are able to conclude that the method works better than the one in current use for the vowels upon which the experiments were conducted. Moreover, an examination of the form of the abstractions suggests that formant peaks may not be as good indicators of vowel class as their "shoulders".

2 SLIM

In this section we first give a summary of SLIM's operation and then describe it in sufficient detail to understand the remainder of this paper.

The general nature of SLIM's application is as follows. A number of distinct training examples is provided for each of several mutually exclusive pattern classes. SLIM operates with the examples of each class to produce abstractions. Only those which are expected to perform well as classificatory templates are retained. Subsequently, novel test items are compared with the abstracted templates and are classified as belonging to the class associated with the matched template which performs best.

We shall now be more explicit. SLIM first prepares a description of each exemplar in terms of user-defined boolean features. It then attempts to find characteristic conjunctive abstractions which distinguish specified classes from each other. The basic operation by which this is done is interference matching. The effect of interference matching applied to the descriptions of two exemplars is to produce a schema (or abstraction) which comprises all and only the features common to both. Because a schema is a set of features, as each encoded exemplar, exactly the same matching process may be applied to a schema and an exemplar.

The procedure of abstracting new diagnostic pattern templates for a class is called decomposition. The class for which characteristic patterns are to be induced is called the positive class. A negative class is also defined, usually as the universe of exemplars minus those in the positive class. Each exemplar from the positive class is taken in turn and interfered with each schema in the dynamic decomposition list. (See Figure 1.) At the outset this list is empty and a copy of the first exemplar is placed in the list. For each subsequent positive exemplar, one schema is derived from interference matching applied to it and each schema in the dynamic decomposition list. In addition, the exemplar is itself entered as a schema. The schemata so formed are then evaluated for placement in the list, and then the next exemplar is processed. The decomposition is complete when all the positive exemplars have been processed.

The evaluation mentioned above is a calculation of the schema's performance as a diagnostic indicator of the positive against the negative class. The performance measure in current use computes a weighted difference between the *a posteriori* expected hit rate (*i.e.* the frequency of matches within the positive class) and the *a posteriori* expected false alarm rate (*i.e.* the frequency of matches by that schema within the negative class). The hit rate and false alarm rate are weighted by a gain and a loss factor, respectively[2].

The number of schemata which may be generated in this fashion rises exponentionally with the number of training exemplars, and so some techniques are used to limit the size of the decomposition list. The power of SLIM derives from its heuristic methods for preventing a combinatorial explosion without seriously compromising the discriminative power of the templates induced. The basis of SLIM's approach lies in the performance measure. Performance limits may be set which act as thresholds for a new schema's acceptance into the dynamic decomposition list. The schemata are ordered in the dynamic decomposition list according to their performance. In addition, a limit on the length of the dynamic decomposition list may be set. This will have the effect of constantly pruning off the more poorly performing schemata. Other heuristic constraints may be applied to the speed the process with minimal discriminative loss[2]. The ones mentioned here are those referred to in the following sections.

This process is usually repeated for each of a number of classes. At the completion of the decomposition for each class, its dynamic decomposition list is merged into the final decomposition list. Once a final decomposition list has been formed, it is possible to classify test exemplars. Each exemplar to be classified is encoded into the feature representation and then matched against each schema in the decomposition list in turn. If a match occurs, *i.e.* if the exemplar contains all the features in one of the schemata, it is assigned to the class from which the schema was derived. The process is self-terminating, so that it is the highest-performing match which determines the classification.

3 The Overlapping Receptive-Field Feature Representation

Learning is considered here to be a process of inducing pattern templates which are as discriminating (*i.e.* precise) as possible and at the same time as general (inclusive) as necessary to characterize each class. These two goals are in conflict in that precision is necessary to take advantage of fine differences, but it is equally necessary to infer beyond the specific training examples in order to encompass novel examples within any domain which is subject to variability. One of the goals of our work is to find features which can do both these tasks well. SLIM is suited to such an effort, because what generalization is not accomplished with the individual schemata themselves will be reflected in the decomposition list as a whole; the list may be viewed as the disjunction of the schemata which comprise it. Because the dynamic decomposition list is itself the basis for incremental learning, any extra capacity for generalization provided by a particular set of features will be reflected multiplicatively in succeeding generations of schemata.

The solution presented here is to describe each value in terms of features which are series of of overlapping intervals. For the purposes of exposition, let us consider a value of 38 on some dimension. The features could be the ranges [5:34], [10:39], [15:44], etc. Let us give these features names:

F1: [5,34]
F2: [10,39]
F3: [15,44]
F4: [20,49]
F5: [25,54]
F6: [30,59]
F7: [35,64]
F8: [40,69]
F9: [45,74]
F10: [50,79]

With this set of features, the value 38 on a dimension whose values range from 5 to 79 would be represented as the conjunctive product

$$F2 \wedge F3 \wedge F4 \wedge F5 \wedge F6 \wedge F7$$

because these are all the predicates which are true for this value. Interference matching with the feature representation for 48, which becomes

$$F4 \wedge F5 \wedge F6 \wedge F7 \wedge F8 \wedge F9,$$

gives the conjunction

$$F4 \wedge F5 \wedge F6 \wedge F7$$

of common features. It should be noted that this schema defines an interval from 35 to 49 which, within the framework adopted, is at once the most precise and the most general observation which can be made from the two events. This method of overlapping intervals thus provides a solution to the problem of simultaneously discriminating and generalizing within a conjunctive abstraction framework.

This descriptive methodology can be considered as a uniform coding technique with four parameters. They are:

(1) The maximum generalization interval, G, which is the distance between the upper and lower bound of each feature interval.

(2) The maximum discrimination interval, D, which is the distance by which adjacent features are shifted with respect to each other and is equivalent to the JND of a learning procedure based on these features.

(3) The lower limit, L, which is the lowest value codable in the series of features.

(4) The upper limit, U, which is the highest value codable in the series of features.

We shall now estimate the efficiency of this form of interval encoding. Given the parameters G, D, L and U ($G \geq D$), the total number of features required to encode corresponding values is always $\leq 2*(U-L)/D \ -1$, the value when $G=(U-L)$, and is always $\geq (U-L)/(2*D)$, the value when $G=(U-L)/2$. Thus there is a trade-off between big G and small number of features. Let us consider an alternative method of interval encoding. The total number of intervals which can be represented by conjunctions of successive overlapping intervals (i.e. the number of features needed if each interval were independently coded by a feature) is $(U-L)/D$ when $G=D$ and $((U-L)/D)*(((U-L)/D)+1)/2$ when $G=(U-L)$, and when $G=(U-L)/2$ is $(3/8)*((U-L)/D)^2$. Thus, as an example, when $G=100$, $D=5$, $U=100$ and $L=0$, the preferred method requires 39 features while the alternative method needs 210, giving an economy of 3 to 1. If $G=50$ instead of 100, the corresponding numbers are 20 and 150, giving an economy of 7.5 to 1. Clearly, as D decreases or as (U-L) increases, for any G, the relative efficiency of the preferred method, which is quadratically related to these parameters, becomes increasingly significant. Thus, if learning is to be based on interval discrimination and generalization, the proposed code if a highly efficient one.

We are encouraged in our use of this approach by several physiological observations. Firstly, the receptive fields of auditory perceptual system neurones are apparently distributed in an overlapping manner[3]. Secondly, the shapes of auditory tuning curves, which define the frequency characteristics of auditory receptive fields, are often wide[4], which suggests that the square window nature of our features may be appropriate. Thirdly, the proposed method is a very general method (not at all language-specific, as a formant-extraction approach might be) and may help to explain how animals can be trained to discriminate between speech sounds[5]. Lastly, the proposed code is a redundant one which would produce well-controlled generalizations if features were to be lost for some reason; although we are not able to go into the possibility or significance of discarding features from schemata here, we wish to point out the method's potential for graceful degradation under such loss.

The method, which we may call the *overlapping receptive field* representation, is applicable to any ordinal scale and may be used for more than one dimension at once, as is the case in the current work.

4 The Parametric Representation and its Preparation

The parametric representation employed here provides, for each centisecond, an amplitude for each of the 128 frequencies which may be sampled at 39.625 Hz intervals between 39.625 and 5000 Hz. They are derived from an original 10 kHz digitization *via* the discrete fast Fourier transform of a 14-pole Linear Predictive Coefficient filter. This provides a smoothed, amplitude-normalized spectrum. Although the signal energy for each centisecond is available along with the spectral data, only the latter were used in the current experiment. Figure 2 shows the spectrum of the central centisecond from an example of /i/.

The data were drawn from two corpi of utterances spoken by the same General American male speaker. Each corpus, one of 27 sentences and one of 40, was segmented and labelled by a phonetician according to aural and waveform criteria. All occurrences of the four phones /i/, /ɪ/, /ə/ and /æ/ were considered. Phones less than 3 centiseconds in duration were rejected. Of those remaining, the ten examples nearest the centre of each corpus were set aside to be used for testing the discrimination produced by the system. The remaining exemplars were those used in training. There were 64 /i/, 99 /ɪ/, 111 /ə/ and 55 /æ/. They were used for training by both SLIM and INTRAC, a Euclidean distance, cluster-based pattern recognition program[6].

Each sample was encoded by the overlapping receptive field technique described above, simultaneously in both of the amplitude and the frequency dimensions. There were six such features for the encoding of each amplitude. The normalization process in the LPC encoding ensured that most of the spectral amplitude points fall within 20 units of 256. In terms of the description above, the features for the amplitudes had a G of 30, a D of 5, an L of 230, and a U of 285, thus providing a maximum discrimination of 5 units and a maximum generalization of 30 over the interval [230,285].

The raw amplitudes at each frequency were not used, but instead the maxima and minima in each frequency interval of 485.5 Hz, considered at successive intervals of 158.5 Hz, were encoded. This sampling started at 198.125 Hz. There were 29 such frequency samples for each centisecond's spectrum, and hence there were a total of 29x2x6=348 features.

The features are gross in that they cover an interval of 30 units, while most of the spectral amplitude points in the data fall within a 40 unit interval.

5 The Production of the Decompositions

Each exemplar having been read into SLIM and encoded, the next step before decomposition was to set the values of the space-limiting heuristics. The maximum size of the dynamic decomposition list was set to 25. The lower admissible limit on performance was set to -.1. (A negative value is useful because a poorly performing early schema may ultimately spawn a better performing one.) The values of loss and gain in the performance metric were 5 and 1, respectively.

In this way, a final decomposition list was produced, consisting of the dynamic decomposition list resulting from decomposition of each of the four phone classes against all the others. On classification of the test items, it was found in 23 of the 80 cases (*i.e.* 28.75%) that the data were so variable that none of the general schemata produced could classify an exemplar. In that case an alternative classification method within SLIM was employed. Here the decomposition process is repeated, but with the additional constraint that when each stored exemplar is converted into a schema for possible addition to the decomposition list only those features which are also present in the example to be classified are retained. Hence only those features which are relevant to the item to be classified will enter into the decomposition. This procedure is called filtered decomposition to distinguish it from the unfiltered abstraction method described in Section 2 above. Those test exemplars remaining unclassified by the unfiltered technique were classified according to which filtered decomposition list contained the highest performing schema.

6 Results

Confusion matrices for the classification results for SLIM and INTRAC are presented as Tables 1 and 2, respectively. The forms of the matrices are similar, suggesting that the two methods respond to intra-class variability similarly. However, SLIM was as good as, or better than, INTRAC in 13 of the 16 cells of the confusion matrices. The case where SLIM is markedly worse is the phone /ɪ/, which also the worst for INTRAC. A measure of overall success is the proportion of elements on the right diagonal of the matrices. SLIM's success is 0.64 on this basis, while INTRAC's success is 0.54.

Figures 2 and 3 show the centre of the largest cluster for /i/ found by INTRAC and the highest performing schema for /i/ found by SLIM, respectively. Formant peaks of Figure 2 are largely absent from Figure 3.

7 Conclusions

Firstly, we can conclude that SLIM may be used effectively to find good characterizations in a very difficult area, at the acoustic level of description of continuous speech. Not only are the recognition rates generally good, but in most cases they show the SLIM outperforms those of another method in current use, when applied to precisely the same data. The phone /I/ gave SLIM more trouble than it did to INTRAC. We feel that this is a sign that our features are still insufficiently general, and we are continuing to refine our feature representation.

Secondly, this is accomplished without recourse to sophisticated techniques of description, such as formant extraction. Indeed, formants are somewhat less in evidence in the schemata, as Figures 2 and 3 exemplify, even though vowels were the training data.

Thirdly, we have given an example of how SLIM may be used to explore the ability of theories about relevant features of speech by testing their ability to discriminate between phones. Our work, to date, has investigated only some of the simplest forms of description which might be used. We are continuing this study by working with other phones, additional speakers and by trying different simple feature representations.

REFERENCES

[1] Newell, A., *et al.* Speech Understanding Systems. Final Report of a Study Group. 1971. Computer Science Department, Carnegie -Mellon University.

[2] Hayes-Roth, F. Schematic Classification Problems and their Solution. Pattern Recognition 1971, vol 6, pp105-113.

[3] Merzenich, M. and Brugge, J. Representation of the Cochlear Partition on the Superior Temporal Plane of the Macaque Monkey Brain Research 1973, vol 50, pp275-296

[4] Katsuki, Y., *et al.* Single Unit Activity in the Auditory Cortex of the Unanaesthetized Monkey. Proceedings of the Imperial Japanese Academy 1960, vol 36, pp435-437

[5] Burdick, C. and Miller, J. Speech Perception by the Chinchilla: Discrimination of Sustained /a/ and /i/. Journal of the Acoustical Society of America 1975, vol 78, pp415-427

[6] Goldberg, H. A Comparative Evaluation of Parametric Segmentation and Labelling Strategies Ph. D. Thesis, Carnegie -Mellon University, 1975

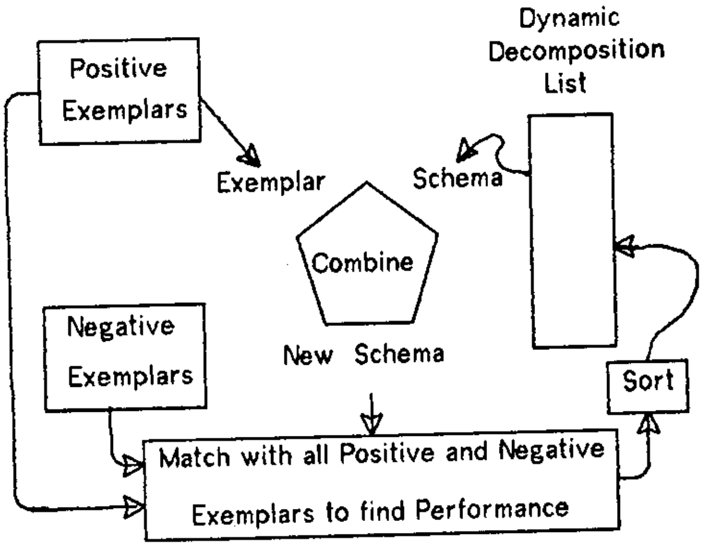

Figure 1: A Diagram of SLIM

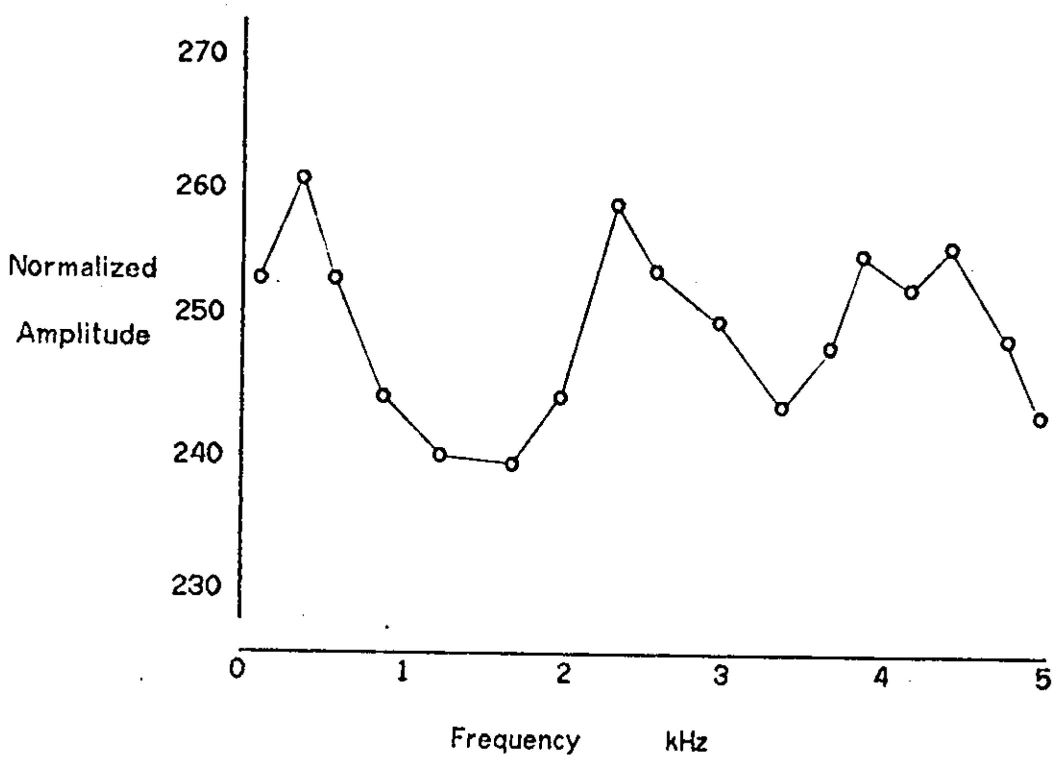

Figure 2: A representative /i/ found by INTRAC

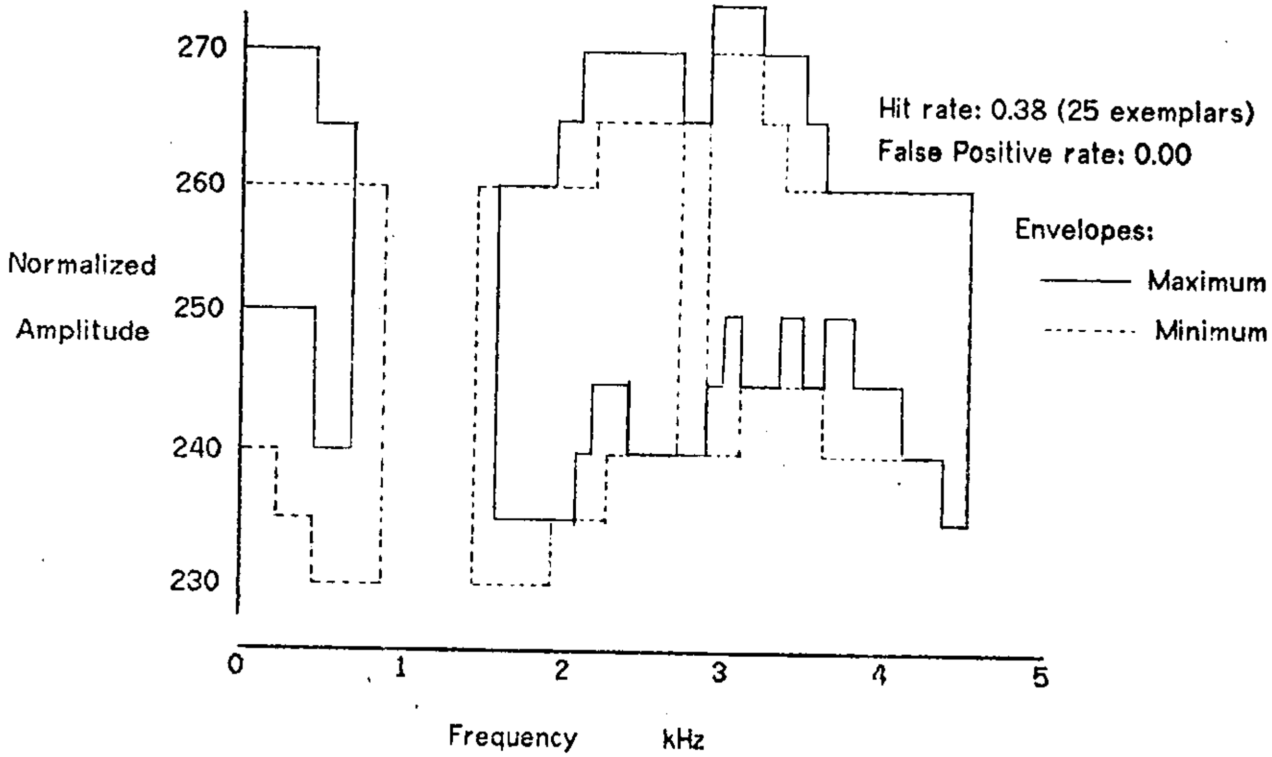

Figure 3: The highest-performing schema for /i/ found by SLIM

		Hand Labels				
		/i/	/ɪ/	/ə/	/æ/	total
	/i/	19	8	0	0	27
SLIM	/ɪ/	1	4	2	2	8
Labels	/ə/	0	3	13	3	19
	/æ/	0	5	5	15	25
	total	20	20	20	20	80

Table 1: Confusion Matrix for SLIM

		Hand Labels				
		/i/	/ɪ/	/ə/	/æ/	total
	/i/	13	0	0	1	14
INTRAC	/ɪ/	5	8	3	6	22
Labels	/ə/	1	6	13	4	24
	/æ/	1	6	4	9	20
	total	20	20	20	20	80

Table 2: Confusion Matrix for INTRAC