Word Recognition in a Reduced Linear Prediction Space

F. Mavaddat
S.K.S. Cheng

Department of Computer Science
University of Waterloo
WATERLOO, Ontario, Canada
N2L 3G1

Research Report CS-84-29
September 1984

Word Recognition in a Reduced Linear Prediction Space

F. Mavaddat
S.K.S. Cheng

Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada

ABSTRACT

A label space is defined as a space to which the reference points of a feature space can be mapped. The measurement of similarity in the space of linear prediction features can benefit from this mapping, and a new two-phase algorithm for word similarity studies is proposed. Two experiments for finding an optimum set of parameters and determining system performance are reported.

Keywords: linear predictive coding, feature space, similarity measurements, vector quantization, isolated word recognition.

Word Recognition in a Reduced Linear Prediction Space

F. Mavaddat
S.K.S. Cheng

Department of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada

Introduction

We first demonstrate a coding technique for the mapping of reference points from a feature space into codewords. Codewords are representable as points in the label space. We will show that the similarity measurements in the feature space are also measurable by the codeword distances in the label space. Such codewords sometimes possess redundancy. Removal of this redundancy may lead to a reduction in the cost of computations.

Redundancy can be reduced by removing some reference points from the feature space. Reduced feature spaces are studied in relation to the linear predictive coding of speech signals. It is shown that in addition to a reduced linear prediction feature space, computationally less expensive similarity measurements can also be employed.

An acoustic processor which assigns codewords to short intervals of speech is proposed. By concatenating these codewords, label matrices are formed. Utterances are compared by measuring the similarity of the corresponding label matrices that result.

An experimental study is made, and it is shown that the proposed techniques are comparable to other known results.

1. The label space

Let us consider n reference patterns by their respective points; $R(1), R(2),..., R(n)$ in the feature space. We also consider the distance measure $d(i,j)$ to be a similarity measure between the ith and the jth reference points, where $d(i,i) = 0$ for $1 \leq i \leq n$, and $d(i,j) > 0$ for $1 \leq i \leq n$, $1 \leq j \leq n$, and $i \neq j$.

Corresponding to each $R(i)$ we will define a codeword, $L(i)$, as an ordered vector of all reference point labels, such that $R(j)$'s label will precede that of $R(k)$ in $L(i)$, if $d(i,j) < d(i,k)$. $L(i)$ will be called the codeword of the ith pattern. Codewords corresponding to individual patterns are uniquely represented by having their own label as their first element. Figure 1 is an example of a feature space with eight codewords.

	(a) a b c d e f g h	(b)
a	0 2 3 1 4 6 7 5	L(1) = adbcehfg
b	2 0 1 3 6 4 5 7	L(2) = bcadfgeh
c	3 1 0 2 7 5 4 6	L(3) = cbdagfhe
d	1 3 2 0 5 7 6 4	L(4) = dacbhegf
e	4 6 7 5 0 2 3 1	L(5) = ehfgadbc
f	6 4 5 7 2 0 1 3	L(6) = fgehbcad
g	7 5 4 6 3 1 0 2	L(7) = gfhecbda
h	5 7 6 4 1 3 2 0	L(8) = hegfdacb

Figure 1. Distance measures of a hypothetical system and corresponding codewords.

We define the distance $D(i,j)$ between $L(i)$ and $L(j)$ in the label space by

$$D\ (i,j) = \sum_{k=1}^{n} |\,(p(i,k) - p(j,k))\,| \tag{1}$$

where $p(i,k)$ $(p(j,k))$ represents the index position of the kth label in $L(i)$ $(L(j))$ and n is the number of label points in the feature space.

There are now two phases of similarity measurements. During the first phase, the distance $d(x,i)$ between the feature vector of the unknown input, $R(x)$, and all reference points, $R(i)$ for $1 \leq i \leq n$ is measured. Based on these measurements $L(x)$ is formed and recognition is based on measuring the distance $D(x,i)$ between $L(x)$ and all other $L(i)$s for $1 \leq i \leq n$. Traditional decision algorithms can then be applied to the association of x with one of the reference patterns.

The nearest neighbour algorithm is a special case of the two phase algorithm which, in a sense, expects an exact match between one of the known and the unknown codewords. Because of the unique representability of the codewords by their first element, this exact match can be reduced to that of comparing the first elements. This eliminates the need for the formation of the codewords and the two are considered matching if their first codeword elements (the nearest neighbour) correspond.

There is little to be said for such two-phase measurement which requires additional computation. In the next section we will demonstrate that, under certain conditions, the additional measurements in the label space can be more than compensated for by the potential computation savings in the feature space.

1.1. Computational advantages

Let us consider n codewords $L(1)$, $L(2)$,...,$L(n)$ each selected from some permutation of n distinct labels. These words are all distinct if

$$D(i,j) > 0 \quad 1 \leq i \leq n, \quad 1 \leq j \leq n, \; i \neq j \tag{2}$$

where $D(i,j)$ is defined by (1). A reference point, in the feature space, is said to be "removable" if all codewords remain distinct after its label is removed.

Intuitively one expects that m such reference points are "removable" if $n \ll (n-m)!$. This need is satisfied by the typical values of n and m in some applications. Codewords corresponding to the eight reference points of Figure 1 are distinct after the removal of the label "a", labels "a" and "b", or even labels "a", "b", and "c".

Removal of the redundant labels may lead to two kinds of computational advantages. The first kind is that of reducing the amount of overall computation. This may be possible if similarity measurements in a reduced feature space more than compensate for the additional costs of label space measurements. This condition is satisfied if:

$$\frac{n}{m} < \frac{c(f)}{c(l)} \tag{3}$$

where $c(f)$ and $c(l)$ are the computational costs of the similarity studies in the feature and the label space respectively, n is the total number of reference points in the feature space and m $(m < n)$ is the number of removable reference points. The remaining $n-m$ reference points are *essential* to the success of the two-phase algorithm.

The second kind is that of replacing the feature space similarity measures by a simpler kind which would not have been possible if the similarity measure was performed totally within the space of the features. This may be possible because pattern similarity questions can be asked in two different ways, leading to distinct formulations with computational differences. The first way measures similarities between the two patterns in absolute terms, and the answers to such questions are in terms of distance measures signifying great, little, or no similarity. The second way measures the similarity of a given pattern with two or more others in relative terms. These questions, in fact, ask for the ordering of many patterns according to their similarity with one. It is not difficult to see that the questions of the first type are more demanding and can be used in answering the second type, while the second type is less powerful and may not be usable as a basis

for answering questions of the first type. Measuring pattern similarities completely within the feature space often requires answers to questions of the first type. When performed as the first phase of a two-phase algorithm as was suggested, needs the second type only. In the next section we will discuss such time savings when applied to measurements in the space of LP features.

2. The label space of LP features.

Over the last few years there has been considerable interest in the study of a suitable distance measure based on features derived by LP techniques (e.g. Itakura (1975), Coker (1976), de Souza (1977), Gupta (1978)). It turns out that all successful formulations, in one way or the other, are based on some form of the power of the residual signal obtained by the filtering of one pattern by the inverse model of another.

Following Makhoul's (1975) formulation, the residue vector $\bar{e}^{\,T} = (e(0), e(i), e(2),...e(p))$ can be defined as

$$\bar{e} = R\bar{a}^{T} \tag{4}$$

where R is the pth order autocorrelation matrix of an arbitrary signal $X1(n)$, and $\bar{a} = (1, a(1), a(2),...a(p))$ is the model of another signal, $X2(n)$. Different functions of $\bar{e}$ have useful properties in measuring the similarities of the two signals $X1(n)$ and $X2(n)$. Gupta (1978) proposed:

$$d_1(X1,X2) = F_1(\bar{e}) = \log\left(\sum_{i=1}^{p} |e(i)|\right) \tag{5}$$

or

$$d_2(X1,X2) = F_2(\bar{e}) = \log\left(\sum_{i=1}^{p} (e(i))^2\right) \tag{6}$$

where $\bar{e}$ is the residual vector of filtering $X1$ by the model of $X2$, as measures of distance.

The direct use of $F(\bar{e})$ towards answering the first type of questions is not satisfactory. Even though model $\bar{a}$ of signal $X1(n)$ results in a minimum prediction error for $X1$, there is no guarantee that the same model will not result in a smaller absolute residual value while filtering some other signals.

To overcome this difficulty, one has to consider any $F(\bar{e})$ relative to the residual vector, say $F(e)$ which can be obtained through the filtering of $X(n)$ by its own model. $F(e)$ is the self-referencing component of the measurement. Should $\bar{a}$ be a model of the similar signal, then $F(\bar{e})$ and $F(e)$ are close and their ratio nears one. Under all other conditions $F(\bar{e}) > F(e)$, and this results in ratios greater than one.

The need for self-referencing has been considered by a number of researchers. Coker and Boll (1976) use $F(\bar{e}) = \bar{a}^T \bar{e}$ as the basis of their studies and propose

$$d(X,\bar{a}) = \frac{\bar{a}^T R \bar{a}}{a^T R a} \tag{7}$$

as a measure of similarity between X and $\bar{a}$, where a is the model of X, and R is the pth order autocorrelation matrix of X.

Itakura (1975), using the log likelihood ratio, derives a similar distance measure:

$$d(X,\bar{a}) = \log \frac{\bar{a}^T R \bar{a}}{a^T R a} \tag{8}$$

Derivation of the input signal model, i.e. the self-referencing component, as required by (7) and (8), contributes to the additional computational expense of answering questions of the first type.

In answering the questions of the second type, we measure the similarity of the input pattern with each of the reference patterns. Because the denominators of the distance ratios are identical in all these measurements, and one is interested only in the relative values of these distances, it is possible to eliminate the self-referencing component from all calculations (as well as the log extraction in Itakura's measure) and base the ordering on the value of the numerators only.

Codewords, representing short and therefore stationary segments of speech, are representations of clusters in the feature space and therefore candidates for quantization of LPC vectors. A codeword-based quantization differs from the standard vector quantization (e.g. Rabiner (1982)) by mapping the LPC vectors into another space. The computational advantages of codebook lookup in the label space are similar to the advantages discussed for the two-phase distance measure.

3. Isolated Word Recognition (IWR) Applications

Every word is partitioned into an equal number of sections, each short enough to be considered stationary. The similarity of every section with a predefined set of signals is measured, and the section is replaced by its codeword. This replaces every uttersnace by its matrix of labels. The overall utterance similarities are measured by measuring the similarities of the label matrices.

3.1 IWR System Organization

For every *essential* reference sound, its model is derived and stored as one of the reference points of the feature space. Once this space is formed, the vocabulary is introduced to the system. Every input is first partitioned into an equal number of sections, each short enough to represent an allophone. The optimum number of such partitions is the subject of one of the following experiments. Once the utterance is partitioned, the relative distances of every one of these intervals from the feature space reference points are calculated and the codewords are formed. Each word's reference template (a matrix of labels) is formed by concatenating the codewords representing the partitions. For every word of the vocabulary several of its utterances are "averaged" (for averaging method used see Rabiner (1978)).

Unknown utterances are processed like the known ones. To recognize the unknown word its label matrix is compared with that of all the known words. In all experiments we have used the sum of the distances between individual codewords, using the time-warping algorithm proposed by Sakoe (1978), as the measure of matrix similarities.

4. Experiments.

To gain a better insight into the power and accuracy of the proposed algorithm, two sets of experiments were designed and conducted. The first experiment was aimed at finding a suitable set of parameters for acceptable system performance. The second experiment tested the system performance for a limited vocabulary of ten words.

4.1 The first experiment

The vocabulary was taken from the first thirty words of a flight reservation system. Ninety words were used for the formation of reference matrices (three for each word in the vocabulary). In the same way, ninety words were used as the test set. The utterances were spoken in a random order and at different times of day.

The parameters under consideration were the order of prediction, the order of partitioning, and a suitable set of reference points in the feature space. While the first two parameters are suited to systematic search, the third evades efficient examination. We limited search to only five subsets of nine predominant sounds in the reference vocabularies. No attempt was made at finding an optimum or *essential* set of reference points. This guarantees that the full implementation will be better than, or at worst similar to the results reported here. Table 1 shows these five sets. The closest phonetic sound to each selected pattern is used for its symbol.

With the order of prediction and reference phonemes fixed at ten and CV1 respectively, the order of partition was varied from five to forty. Figure 2 shows the number of errors under the different orders of partition.

With the order of partition and the reference phonemes fixed at twenty and CV1 respectively, the order of prediction was varied from six to eighteen. Figure 3 shows the number of errors under the different orders of prediction.

With the order of partition and the order of prediction fixed at twenty and ten respectively, the sound sets were varied over all sets of Table 1. The use of C resulted in the worst performance while CV3 gave the best result both confirming the intuitive expectations.

Set Label	Composition
C	*n*, *f*, *s*, *t*
V	*u*, *i*, *ei*, *aI*, *a*
CV1	*n*, *s*, *f*, *t*, *i*, *ei*, *u*
CV2	*n*, *s*, *f*, *t*, *i*, *ei*, *u*, *aI*
CV3	*n*, *s*, *f*, *t*, *i*, *ei*, *u*, *aI*, *a*

Table 1, Composition of Reference Sound Sets

4.2 The second experiment

>From the first experiment the optimal orders of prediction and partition were found to be about sixteen and twenty respectively. It was also found that around nine reference sounds are sufficient for a reasonable recognition rate.

The experimental procedure was the same as in the first experiment. This time the speaker was a different male with English as his second language. The vocabulary consisted of the ten digits. The test set was increased to 380 (38 for each word) to give a more significant estimation of the recognition rate.

In this experiment four test tokens were classified incorrectly- once *one*, twice *three*, and once *nine*.

5. Acknowledgements

Thanks are due to IBM World Trade Corp., the University of Waterloo, and the National Science and Engineering Research Council of Canada who have supported this project at its different stages of development.

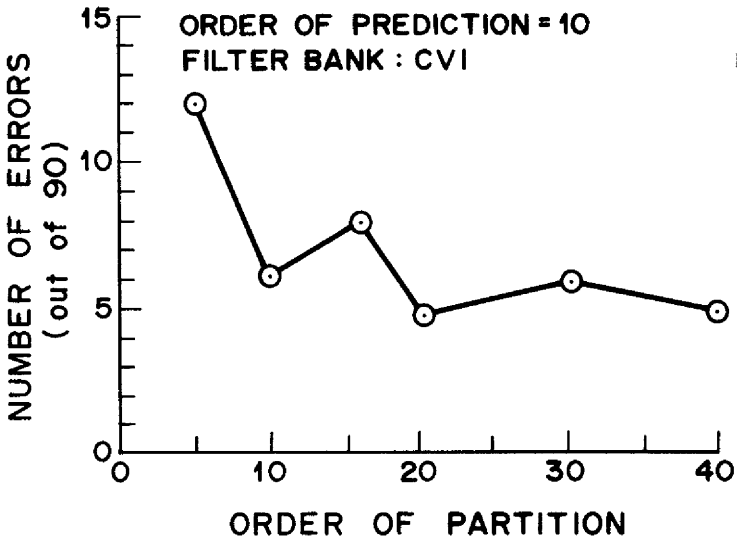

Figure 2. Variations of the Order of Partition:

Error Rate vs. Order of Partition

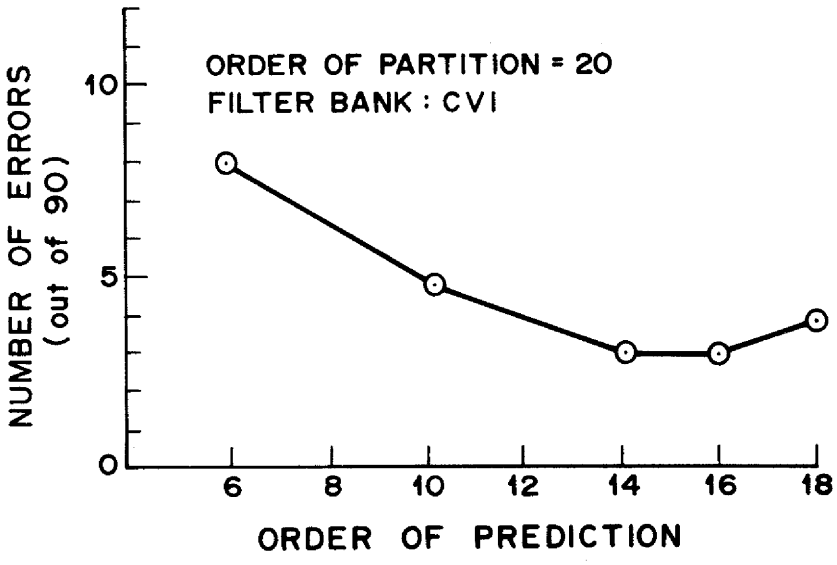

Figure 3. Variation of the Order of Prediction:

Error Rate vs. Order of Prediction.

References

Coker, M.J. and S.F. Boll (1976). An improved isolation word recognition system based upon the linear prediction residual. *ICASSP*, 206-209.

de Souza, P.V. (1977). Statistical tests and distance measures for LPC coefficients. *IEEE Trans. on ASSP* 25, No. 6, 554-559.

Gupta, V.N., J.K. Bryan and J.N. Gowdy (1978). A speaker-independent speech recognition system based on linear prediction. *IEEE Trans. on ASSP* 26, No. 1, 27-31.

Itakura, F. (1975). Minimum prediction residual principle applied to speech recognition. *IEEE Trans. on ASSP* 23, 67-72.

Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE* 63, No. 4, 561-580.

Sakoe, H. and S. Chiba (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. on ASSP* 26, No. 1, 43-49.

Rabiner, L.R. (1978). On creating reference templates for speaker independent recognition of isolated words. *IEEE Trans. on ASSP* 26, No. 1, 34-43.

Rabiner, L.R., Levinson, S. E. and M. M. Sondhi (1982). On the application of vector quantization and hidden markov model to speaker-independent, isolated word recognition. *The Bell System Technical Journal* 62, No. 4, 1075-1105.