V. COMPARISON OF ALGORITHMS

A. Estimation Error

The variance of the estimation error \( \hat{a} - a \) for the solutions (4) and (7) is found by substituting (1) in (4). Defining

\[
\sigma^2 = E \{ \xi^2 \}
\]

and

\[
\sigma^2 = E \{ \xi^2 \}
\]

we find for large \( N \) an estimation error variance of \( \sigma^2/(N\sigma^2) \).

For the sign decorrelator (15), we similarly substitute (1) in (14) and find

\[
\hat{a} - a = \frac{1}{n} \sum_{i=1}^{n} \xi_i \text{sign}(s_{i-1})
\]

(17)

For large \( n \), the denominator approaches \( \sqrt{2/\pi} \sigma_\xi \) [3, p. 258]. The numerator variance is \( \sigma^2/n \). The estimation error variance is, therefore, \( \pi\sigma^2/2n\sigma^2 \). The sign decorrelator thus increases the variance by a factor of \( \sim 12 \).

B. Required Computations

Table I compares the required computations for each of the algorithms described for processing the sequence \( \{s_0, s_1, \ldots, s_N\} \).

References


II. DIMENSIONALITY, SAMPLE SIZE, AND FEATURE SIZE

Kanal [2] points out that the most important recent work in the design of pattern classifiers is that concerned with the relationship between the number of features, the number of design samples, and the achievable error rates. These results are concerned with: 1) quantitative estimation of the bias in the error estimation based on design-set, 2) whether performance is improved by adding additional features, 3) how best to use a fixed size sample in designing and testing a classification scheme, and 4) comparison of density estimation and nonparametric techniques. Foley derives expressions for design-set error rates for a two-class problem with multivariable normal distributions as a function of sample size for class \( (N/N) \) and dimensionality of feature vector \( (L) \) [6]. Fig. 1 shows Foley's results and the importance of \( (N/L) \) ratio on the design- and test-set error rates. Unless \( (N/L) \) is large enough, the design-set error rate has a large optimistic bias. Foley recommends that \( (N/L) \) should at least be greater than three. At this stage, it is instructive to observe the typical \( (N/L) \) ratios considered in speaker recognition literature. Atal [7] uses six utterances per speaker as design-set and a 12-dimensional feature vector giving
an \((N/L)\) ratio of 0.5, which evidently gives significant disparities in design-set, test-set, and Bayes' optimum error rates, as seen from Fig. 1. Foley's results thus are useful in selecting the size of design data set for a given feature size. For example, if pitch contour of 10 dimensions is used as a feature, at least four dimension of feature vector; size of design data set for a given feature size. For example, if Gaussian densities is not unreasonable for typical speech param-

<p>| (a_0 = 0.3) | 1.049 |</p>
<table>
<thead>
<tr>
<th>(x_1 = x_2 = 10)</th>
<th>(E(G_p))</th>
<th>(E(G))</th>
<th>(E(G_p))</th>
<th>(E(G))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2953</td>
<td>0.2947</td>
<td>0.3069</td>
<td>0.2915</td>
</tr>
<tr>
<td>4</td>
<td>0.3468</td>
<td>0.2480</td>
<td>0.3631</td>
<td>0.2236</td>
</tr>
<tr>
<td>8</td>
<td>0.3783</td>
<td>0.2040</td>
<td>0.4499</td>
<td>0.2159</td>
</tr>
<tr>
<td>16</td>
<td>0.3986</td>
<td>0.1648</td>
<td>0.5392</td>
<td>0.1621</td>
</tr>
</tbody>
</table>

\[ \Delta = \frac{1}{2} \text{Mahalanobis distance between the two populations, defined by} \]

\[ \Delta^2 = (\mu_2 - \mu_1)' \Sigma^{-1} (\mu_2 - \mu_1). \]

In practice, the discriminant function is calculated from estimates of parameters of these distributions, which gives rise to a number of error rates. Moron [10] has tabulated the test-set, design-set, and optimal-error rates for a number of cases from which the following figures are extracted and presented in Table I.

It is interesting to interpret Moron's results in the context of speaker verification systems. The two classes are "ACCEPT" and "REJECT" in a speaker verification system. When \(N_1 = N_2 = 10\), we have a two-speaker verification system, with 10 design samples per class. When \(N_1 = 8, N_2 = 32\), we may interpret the corresponding results in Table I to be for a speaker verification system with \(M = 5\) and eight utterances per speaker as design-set. "ACCEPT" class has 8 design samples, while "REJECT" class, in this case, has 32 design samples. When the optimal achievable error rate for this system in Table IA is 0.3 with \(\Delta = 1.049\), the estimated design-set error rate that will be obtained with a 16-dimensional feature vector is seen to be 0.1448 for the first case and 0.1621 for the second case. This clearly explains the optimistic bias in the performance estimates obtained in literature. The test-set error rates are seen to be increasing to a fairly high value and indicate the possible results if an independent test-set is used. Similar conclusions may be drawn from Table IB.

Moron's study [10] also gives a good statistic, namely, the Mahalanobis distance for feature evaluation. A feature vector with larger \(\Delta\) is better, as it gives smaller error of classification. Sambur [5] discusses the disadvantages of \(F\)-ratio as a statistic for feature evaluation and suggests that the relative merit of a group of features should be based upon its performance in a classifier. In practice, the estimated Mahalanobis distance between two populations gives such a useful statistic, and Moron's study indicates that there is no need to build a system to assess the error performance if the Gaussian assumption for feature distribution is satisfied and a linear classifier is assumed.

### III. ERROR RATES FOR A SPEAKER VERIFICATION SYSTEM

In this section, we present the computed error rates for a two-class pattern recognition problem such as a speaker verification problem. Suppose we have a system designed for \(M\) speakers, the number of design samples being \(N_1\) and \(N_2\) for each of the two classes, and a linear discriminant function is used for classification. It is assumed that the feature vector is of dimension \(p\) and the distributions for the two classes are \(N(\mu_1, \Sigma)\) and \(N(\mu_2, \Sigma)\). The assumption of multidimensional Gaussian densities is not unreasonable for typical speech parameters [5]. The probability of misallocation is a function of \(\Delta\), the Mahalanobis distance between the two populations, defined by

\[ \Delta^2 = (\mu_2 - \mu_1)' \Sigma^{-1} (\mu_2 - \mu_1). \]

### IV. CONCLUSIONS

This study brings out the important factors of the size of the data-set and the dimension of feature vector on the estimates of performance of an automatic speaker recognition system and explains the optimistic bias in the performance assessments available in the literature. The ultimate factor for efficient discrimination turns out to be the distance between the populations, thus confirming the need for a continuing search for better features for speaker discrimination.

### ACKNOWLEDGMENT

The authors wish to thank Prof. B. S. Ramakrishna and Dr. B. Yegnanarayana for their useful discussions.

### REFERENCES

Comments on "A Simplified Computational Algorithm for Implementing FIR Digital Filters"

ULRICH HEUTE

In the above paper, a software realization of finite-duration impulse response (FIR) filters with a length \( N \) impulse response and general or linear phase was regarded. It makes use of a "moving pointer," indicating the address of the latest sample \( x(n) \) within a dynamic storage array for the state variables. This, of course, is exactly the software version of the well-known technique for hardware filters, simulating shift registers by random-access memories (RAM's) and a counter.

The main point of the correspondence dealt with, however, is the only difference between the software and hardware version—at the highest address, a jump to the lowest one has to occur. In hardware this means that the counter is simply reset by some logical gate, whereas in a software realization, the gate is replaced by a programmed check of the addresses in each calculation step.

In order to save the time needed for these checks, the author suggests doubling the length of the state-variable memory to keep all variables \( x(n) \) in two storage cells separated by \((N-1)\) addresses and, thus, to prevent the pointer from "falling outside the range" without checking its actual position in every step [see Fig. 1(a)].

This idea saves time as intended, but it is not the only possible solution of the problem—and it is not the best one, especially in the linear phase case:

1) Obviously, the same "trick" may be applied to the coefficient memory as well. Doing so avoids the "storing \( x(n) \) twice" operation occurring in each calculation cycle.

2) In both solutions, there is a "dummy memory cell"—in the version of the paper considered, it contains the latest sample and moves through the upper part of the doubled memory, so it is needed to keep the program working in the intended manner. If the method is applied to the coefficient register, it has always the same address and contains the first element of the impulse response; thus, this cell may as well be omitted [see Fig. 1(b)].

3) In the linear phase case, doubling the coefficient memory means doubling a (roughly) \( N/2 \) storage array instead of an array of \( N \) state variables.

So, a simple extension of the programming method proposed in the above paper yields an equivalent solution with even a