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 $\sigma^2 = E\{\epsilon_i^2\}$ and $\sigma_s^2 = E\{s_i^2\}$, we find for large N an estimation error variance of $\sigma^2/(N\sigma_s^2)$.

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

$$\hat{a} - a = \frac{\frac{1}{n}\sum_{i=1}^{n} \epsilon_i \operatorname{sign}(s_{i-1})}{\frac{1}{n}\sum_{i=1}^{n} |s_{i-1}|}. \tag{17}$$

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

Computer simulation results verify that the two iterative algorithms give close results for $\hat{a}_n$ and for the prediction error power.

B. Required Computations

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

References

[1] J. Makhoul, "Linear prediction: A tutorial review," *Proc. IEEE*, vol. 63, pp. 561-580, Apr. 1975.

[2] B. S. Atal and S. L. Hanauer, "Speech analysis and synthesis by linear prediction of the speech wave," *J. Acoust. Soc. Amer.*, vol. 50, no. 2, pp. 637-655, 1971.

[3] H. Cramer, *Mathematical Methods of Statistics*. Princeton, NJ: Princeton Univ. Press, 1957.

[4] R. L. Kashyap, C. C. Blaydon, and K. S. Fu, "Stochastic approximation," in *Adaptive, Learning and Pattern Recognition Systems*, J. M. Mendel and K. S. Fu, Ed. New York and London: Academic, 1970.

[5] N. S. Jayant, "Digital coding of speech waveforms: PCM, DPCM, and DM quantizers," *Proc. IEEE*, vol. 62, pp. 611-632, May 1974.

[6] M. D. Srinath and M. M. Viswanathan, "Sequential algorithm for identification of an autoregressive process," *IEEE Trans. Automat. Contr.*, vol. AC-20, pp. 542-546, Aug. 1975.

[7] M. R. Sambur, "An efficient linear prediction vocoder," *Bell Syst. Tech. J.*, vol. 54, pp. 1693-1724, Dec. 1975.

Performance Evaluation of Automatic Speaker Verification Systems

V. V. S. SARMA AND D. VENUGOPAL

Abstract—**Attention is drawn to some recent results of pattern recognition theory in the context of designing an automatic speaker verification system of prescribed performance. Associated problems of data base preparation, the number of customers for which the system is to be designed, and the number of features to be used are discussed.**

Manuscript received June 7, 1976; revised October 25, 1976.

The authors are with the Department of Electrical Communication Engineering, Indian Institute of Science, Bangalore 560012, India.

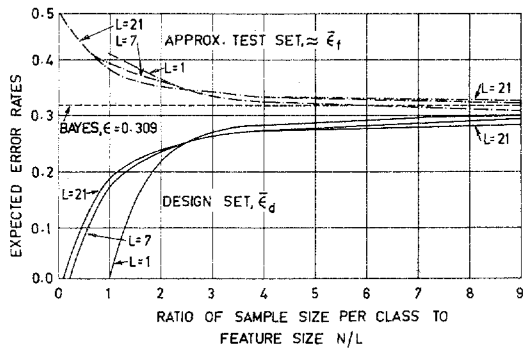

Fig. 1. Expected error rates ($\delta^2 = 1.0$) (after Foley [6]).

I. Introduction

This correspondence considers some aspects of design of automatic speaker verification systems of prescribed accuracy. A recent survey of the field shows that the feasibility of such systems is definitely established, while the performance claims of such systems appear to show an optimistic bias [1]. This problem is inherent in the design and analysis of any pattern recognition scheme and has of late received some attention in the pattern recognition literature [2]. This optimistic bias in performance assessment is due to the problems of dimensionality and sample size, and this aspect has largely passed on unnoticed in speaker recognition literature until recently. Wasson and Donaldson [3] state: "all our results showing percent identification error may be somewhat optimistic, being based on tests in which the same samples were used both for training and testing of the decision algorithm." One of the reasons for this apparent neglect of this problem appears to be that the main emphasis in the area is on the search for new and better features for speaker discrimination. Only recently, there were some efforts for evaluating and selecting features for use in automatic recognition systems [4], [5]. Wolf [4] uses F-ratio for feature evaluation. Sambur suggests two alternative approaches: 1) using an independent test set for performance assessment, or 2) a parametric approach based on estimating unknown multidimensional distributions using the design data set and using well-known bounds on probability of error [5]. This correspondence draws attention to several recent results on this problem in statistical literature and their use in the design of automatic speaker verification systems.

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) 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 errror 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

TABLE I
THE TEST-SET AND DESIGN-SET ERROR RATES FOR A TWO-CLASS PROBLEM (AFTER MORON [10])

	A: $\alpha_0 = 0.3$, $\Delta = 1.049$			
	$N_1 = N_2 = 10$		$N_1 = 8$, $N_2 = 32$	
p	$E(\hat{\alpha}_T)$	$E(\hat{\alpha})$	$E(\hat{\alpha}_T)$	$E(\hat{\alpha})$
1	0.3081	0.2947	0.3059	0.2915
4	0.3468	0.2480	0.3831	0.2536
8	0.3723	0.2040	0.4499	0.2158
16	0.3986	0.1448	0.5382	0.1621
	B: $\alpha_0 = 0.2$, $\Delta = 1.683$			
1	0.2058	0.1939	0.2045	0.1984
4	0.2304	0.1696	0.2482	0.1701
8	0.2544	0.1435	0.2995	0.1485
16	0.2871	0.1050	0.3843	0.1150

Note: α_0 = optimal error rate; Δ = the Mahalanobis distance; p = dimension of feature vector; $E(\hat{\alpha}_T)$ = the expected value of the test-set error rate; $E(\hat{\alpha})$ = the expected value of the design-set error rate.

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 30 utterances per speaker are necessary.

At this point, it may be appropriate to mention the optimum use of a given data set in the design and evaluation of a classification scheme. There are at least four methods [8]: 1) The R-method or the resubstitution method, where the design-set is also used as test-set which gives the maximum optimistic bias. 2) The H-method or the "hold-out" method where the data set is partitioned into design-set and test-set. The former set is used to design the classifier, and the latter to test the performance [7], [9]. This method gives pessimistic estimates of error and makes inefficient use of the data. 3) The U-method or the leave-one-out method, wherein the data set is partitioned into (N-1, 1), and the classifier is designed on the basis of N-1 samples and tested on the basis of the remaining one sample. All possible N combinations are tried and averaged to get performance estimation. 4) The π-method, a compromise between U- and H-methods. All these point out the need for a much bigger data base and suggest the use of improved methods for design of the verification system.

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 Δ, the 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 $N_1 = 8$ and $N_2 = 32$, 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 Δ 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 satisified and a linear classifier is assumed.

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

[1] V. V. S. Sarma and B. Yegnanarayana, "A critical survey of automatic speaker recognition systems," *J. Comput. Soc. India*, vol. 6, pp. 9-19, Dec. 1975.

[2] L. Kanal, "Patterns in pattern-recognition: 1968-1974," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 697-722, Nov. 1974.

[3] D. A. Wasson and R. W. Donaldson, "Speech amplitudes and zero-crossings for automated identification of human speakers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 390-392, Aug. 1975.

[4] J. J. Wolf, "Efficient acoustic parameters for speaker recognition," *J. Acoust. Soc. Amer.*, vol. 51, part 2, pp. 2044-2056, June 1972.

[5] M. R. Sambur, "Selection of acoustic features for speaker identification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 176-182, Apr. 1975.

[6] D. H. Foley, "Considerations of sample and feature size," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 618-626, Sept. 1972.

[7] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech for automatic speaker identification and verification," *J. Acoust. Soc. Amer.*, vol. 55, pp. 1304-1312, June 1974.

[8] G. T. Toussaint, "Bibliography on estimation of misclassification," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 472-479, July 1974.

[9] A. E. Rosenberg and M. R. Sambur, "New techniques for automatic speaker verification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 169-176, Apr. 1975.

[10] M. A. Moron, "On the expectation of errors of allocation associated with a linear discriminant function," *Biometrika*, vol. 62, pp. 141-148, Apr. 1975.

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

ULRICH HEUTE

In the above paper,[1] 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[1] yields an equivalent solution with even a

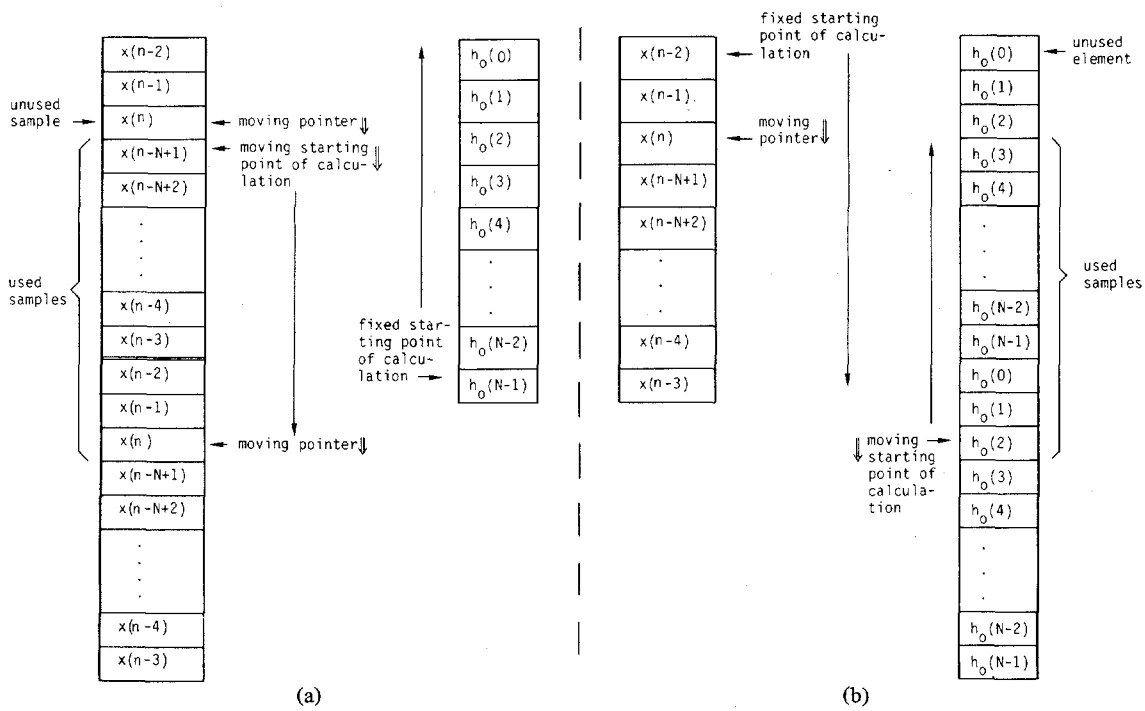

Fig. 1. Memory doubling saving index checking. (a) Doubling the state-variable storage [1]. (b) Doubling the coefficient memory (minus one cell).

Manuscript received October 13, 1976; revised January 12, 1977.

The author is with the Institut für Nachrichtentechnik, Universitat Erlangen-Nurnberg, D-8520, West Germany.

[1] L. R. Rabiner, this issue, pp. 259-261.