Self-Corrective Character Recognition System

G. NAGY AND G. L. SHELTON, JR., SENIOR MEMBER, IEEE

Abstract—The output of a simple statistical categorizer is used to improve recognition performance on a homogeneous data set.

An array of initial weights contains a coarse description of the various classes; as the system cycles through a set of characters from the same source (a typewritten or printed page), the weights are modified to correspond more closely with the observed distributions. The true identities of the characters remain inaccessible throughout the training cycle.

This experimental study of the effect of the various parameters in the algorithm is based on ~30 000 characters from fourteen different font styles. A fivefold average decrease over the initial rates is obtained in both errors and rejects.

Introduction

THE SELF-CORRECTIVE character recognition system about to be described differs from the ordinary garden variety adaptive or statistical categorizer in that the only information available to the system consists of the unknown characters themselves, and of a set of starting parameters containing a coarse description of the classes.

Because the true identity of the characters is not known, erroneous decisions cannot be used to initiate modification of the parameters, nor can "optimal" hyperplane boundaries [1], [7] be calculated to characterize each class. Instead, the initial decisions of the categorizer are considered error-free (rejects are permitted), and a labelled training set is constructed on a "best guess" basis. A new set of decision parameters is now calculated with a single statistical algorithm, and the recognition cycle begins again.

Whether a self-corrective algorithm converges to a low error rate performance depends on the data set, the starting point, and the fine details of the decision procedure. It is easy to see that, even for a very simple two-class, single-layer categorizer, the question of correct convergence rapidly reaches ultra-analytic complexity. It is the purpose of this paper to show that, despite the lack of solid analytical foundations, such a mechanism can significantly improve performance in practical pattern recognition systems.

No claim of originality is advanced for the basic idea. The desirability of such a scheme is broadly hinted at in much of the early speculation on adaptive systems [15], [2], [10]. Experiments in spontaneous organization with unbiased starting points are described in Rosenblatt's *Neurodynamics* under the heading "*R*-controlled reinforcement" [13]. Block et al. invoke only a minimum distance criterion for feature formation in the first layer of a two-layer adaptive pattern recognizer [3]. A somewhat similar problem, that of convergence in the mean on partially misidentified training sets, is discussed by Duda and Singleton [5]. Widrow describes "selective bootstrapping" with minimal supervision in balancing a broomstick [17]. Sears attempts to analyze the interaction of adaptive linear measurements and decision functions [14]. Both analytic and experimental studies on the recognition of samples characterized by continuous density functions with unknown parameters have been recently published by Cooper [4], Fralick [6], and Patrick and Hancock [11].

In view of all this interest, the dearth of published experiments on the application of the self-corrective principle to character recognition is indeed surprising.

The authors are with the IBM Watson Research Center, Yorktown Heights, N. Y.

The Experimental System

The experimental character recognition system at the IBM Thomas J. Watson Research Center provides a convenient framework for testing the idea outlined. The essential features of this system, excluding the self-corrective "positive feedback" loop, are briefly described in this section; additional information may be obtained from [8], [9], and [12].

Documents are scanned by means of a cathode ray tube scanner. The resolution is sufficient to yield a binary matrix of 15 by 22 black and white points for an average sized typewritten capital anywhere on an $8\frac{1}{2}$ inch by 11 inch page. A small special purpose unit, controlled by a general purpose IBM 1401 computer, takes care of document changes, character localization, separation of adjacent characters, stray noise bit suppression, and threshold adjustment. The volume of data processed requires that manual intervention during the scanning phase be held a to minimum.

The video bits from the scanner are shifted through every position of a long shift register in order to achieve registration invariance. The so-called "feature extraction" process takes place here: ninety-six AND-gates are wired, by means of a plugboard, to the shift register outputs. Each AND-gate may have five to nine inputs; these may represent black or white points in the character.

A feature bit is "on" or +1 if the corresponding AND-gate has been satisfied at least once in the course of translating the video through the shift register, i.e., if the geometry represented by that n-tuple occurs anywhere on the character. Although there are facilities for threshold gate type measurements as well as the derivation of positional information, these were not used in the present series of experiments.

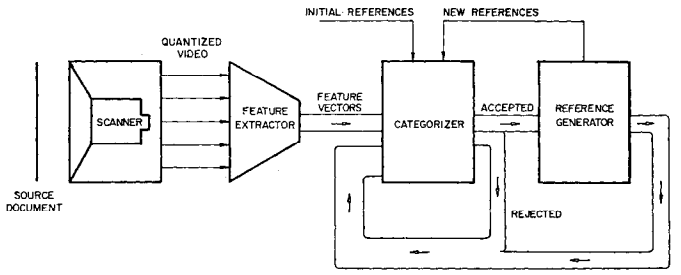

Fig. 1. Schematic diagram of experimental character recognition system. The feature vectors of the characters, labelled by the categorizer, are recirculated to improve the references.

The last stage is a Bayes decision on the ninety-six bit feature vectors. The individual features are assumed to be statistically independent of one another, so that the Bayes decision consists simply of selecting the largest of a set of weighted sums of the feature bits. For ease of implementation, the weights are allowed only three levels; the weight vector corresponding to a character class is then spoken of as a "ternary reference." Q quantization levels of 15 and 85 percent have been experimentally shown to be satisfactory for most applications. If a feature bit is "on" in more than 85 percent of the occurrences of a particular character class, the corresponding reference bit for that class is set to plus one, minus one for less than 15 percent, and zero otherwise.

Thus, "ones" in the reference vectors may be thought of as "on" features, "minus ones" as "off" features, and "zeros" as "don't cares." Then the inverse of the distance between a reference and a feature vector is a reasonable measure of the likelihood that the feature vector originated in the class represented by the reference vector, provided that an additive constant, proportional to the number of "don't cares" in the reference, is included in the calculation. The minimum value of the distance is 0, the maximum is 96.

If none of the references are sufficiently close (within 24 bits) to a feature vector, the character is rejected. A more common occurrence is that of several references almost equally close to the feature vector in question. A reject threshold is therefore defined, specifying the minimum acceptable distance between the first and second ranking candidates. When this threshold is high (say, 5) the error rate is small, but a large fraction of the characters are rejected. When it is low (0 or 1), there are few rejects, but the error rate rises. For references generated at the 15 percent and 85 percent quantization levels, a reject threshold of two normally yields an acceptable 1-to-3 or 1-to-5 errors-to-rejects ratio.

It is, of course, highly desirable to generate references on material representative of the test data. On clean, single-font typewritten material, when the references are derived from identified samples from the same typewriter as the test set, it is possible to achieve a performance rate of the order of 0.2 percent error with 1.0 percent rejects on most fonts [8].

In commercial applications it is, however, unrealistic to expect identified sample alphabets to precede each batch of data. In the United States, there are some thirty major font designs currently in use on typewriters only; in addition, each typewriter contributes its own vagaries in the form of bent typebars, misaligned platens, and ribbon ink variability. When an "average" reference set, generated from a collection of characters including all these variations, was substituted for the custom tailored set, in an attempt to obtain satisfactory overall performance, the percentage of correctly recognized characters seldom exceeded 95 percent (Table 1). While provision of more than one reference for each character class does improve performance, engineering considerations may prevent the inclusion of a sufficient number to approach the quoted single-font performance. Nor does human labeling of font types seem a workable solution for commerical page readers operating on the full range of expected inputs.

The Algorithm

The system described above may be readily modified to take advantage of the recognition results to improve performance. The modified system, including a feedback loop with a "reference generator," is shown in Fig. 1. The mode of operation is as follows.

The processor cycles through a preset number of characters, concluding each recognition phase by "averaging" all the feature vectors which were accepted in a given channel, and generating new references by the thresholding process. When no characters are accepted in a channel, that reference remains unchanged.

At the end of a cycle the reject threshold is decremented by one bit whenever the number of rejects in that cycle failed to decrease by a preset fraction F. Iteration ceases when the reject threshold reaches a given minimum, when the number of rejects falls below a specified level, or when the number of cycles reaches some arbitrary maximum.

The self-corrective algorithm offers the possibility of achieving essentially single-font, single-machine performance on batched data. Conditions for success are, in general, conjectured to be: 1) fairly good initial references to avoid mislabeling the derived references, and 2) moderately large sets of characters of the same origin.

Experiments

Important factors to be studied appear to be the level of quantization of the weights, the rate of decrease of the reject threshold, and the average number of characters required for convergence. While there is little

reason to believe that the various parameters do not interact at all, an exhaustive investigation was not attempted.

Two series of experiments were performed. The object of the first set was to find a workable range of values for the parameters in the algorithm. The second set of experiments was designed to evaluate the recognition performance obtainable on as large and variegated a data set as possible.

About one thousand upper case characters each of two different sans-serif fonts, Model B Artisan and Model B Courier, were selected for "tuning the algorithm." Initial references were generated by averaging the feature vectors from three common serif fonts and quantizing at 15 percent and 85 percent (Q = 85 percent).

Figure 2 shows the effect of changing the level of quantization (Q) of the reference bits on 520 Courier characters with F = 0.10. It is puzzling that the lowest final error rate and the fastest decrease in the number of rejects is attained at about 60 percent, while 85 percent has been found best for labelled alphabets.

Some of the same runs were repeated with F = 0.25 to see the effect of changing the rate of decrease of the reject threshold. Whether F = 0.10 or 0.25 has little effect on the ultimate performance, but F = 0.25 yields somewhat faster convergence. These runs also favored a Q of 60 percent.

Figure 3 shows the influence of the number of samples used for training. The rejects/iterations curves are plotted for data sets of 52 to 884 characters, i.e., two to sixteen samples per class with Q = 60 percent and F = 0.10. All but the 130-character set converged to zero error rate. It appears that increasing the number of samples beyond 10 per class is not necessary with a statistically homogeneous data set.

The second series of experiments included two thousand samples each of twelve different fonts of serif and sans-serif varieties from electric and manual typewriters and several posting machines. The quality ranged from clear sharp impressions to characters perturbed by misaligned platens, bent type bars, uneven keyboard pressure, and ribbon life variations. An alphabet from each font is shown in Fig. 4. The initial (average) references used here were generated from nine other font styles (Fig. 5), which were deemed typical of the range encountered in practice. Several of the fonts in the test data are similar in style to fonts in the design data; others are quite different.

Self-corrective iterations with Q = 60 percent and F = 0.25 were run on 500 characters of each of the twelve fonts. For the sake of clarity, only the best and worst results are shown in Fig. 6. Table I contains the error and reject rates obtained when the iteratively derived references are matched against the training features with a reject constant of two to obtain a practicable error/reject ratio. The performance of the initial references (same reject threshold) is tabulated for comparison; a significant improvement is obtained by the algorithm in every case. Note that the decrease in the number of rejects in Figs. 3 and 6 is due to both the changes in the reject threshold and the improvement in the references; the real proof of the pudding is in Table I.

Also shown in Table I are the error and reject rates with the derived references on 1500 new characters from the same fonts. Although the performance deteriorates slightly, it is still clearly superior to that of the initial references. This again suggests that on large homogeneous batches of data, it may be sufficient to iterate only on the first few hundred characters to obtain satisfactory performance on the whole batch.

Another experiment on the same 24 000 characters consisted of deriving references by setting the reject constant to zero on the first cycle and never allowing it to change. Rejects thus arose only when two scores were identical. Table II lists the error and reject rates obtained with the references derived in this manner on both the 500 character training set and 1500 new characters, with the reject threshold set to two. These results are surprisingly similar to the first set; apparently the effect of the decreased confidence in the decision is offset by the increased number of accepted correctly identified characters. The error plots, showing the action of the algorithm, appear in Fig. 7.

In an attempt to link the speed of convergence and the final error rate to some intrinsic relation between the starting parameters and the alphabet under consideration, the distance between the initial references and the averages of the feature vectors of that font, quantized at the same level as the initial references, was calculated. The mean value of this modified Hamming distance, in ternary digits (tits), is shown on Table III.

The entries in the table represent the amount of modification the initial references must undergo to serve as the final references used for recognition. For example, the entry "26" for the Underwood Pica font shows that for each character, on the average, 26 "don't care" points might have to become "cares" or perhaps, 13 "care" points should change sign.

Unfortunately, no significant correlation was observed between these distances and the required number of iterations; nor do particularly large entries for a single character necessarily correspond to channels with high error rates. Evidently, a useful analysis of the process will have to be based on a finer characterization of the "feature space."

To study the effect of the natural distribution of letters in text, 4300 characters from a typescript of legal material, in lower case prestige elite font, were scanned and processed. Twelve iterations (with F = 0.25, Q = 60 percent, initial reject threshold = 5, and the same initial references as in the previous experiments) were required to stabilize the references. With a reject threshold of 2, the final error rate was 0.05 percent with 0.7 percent rejects.

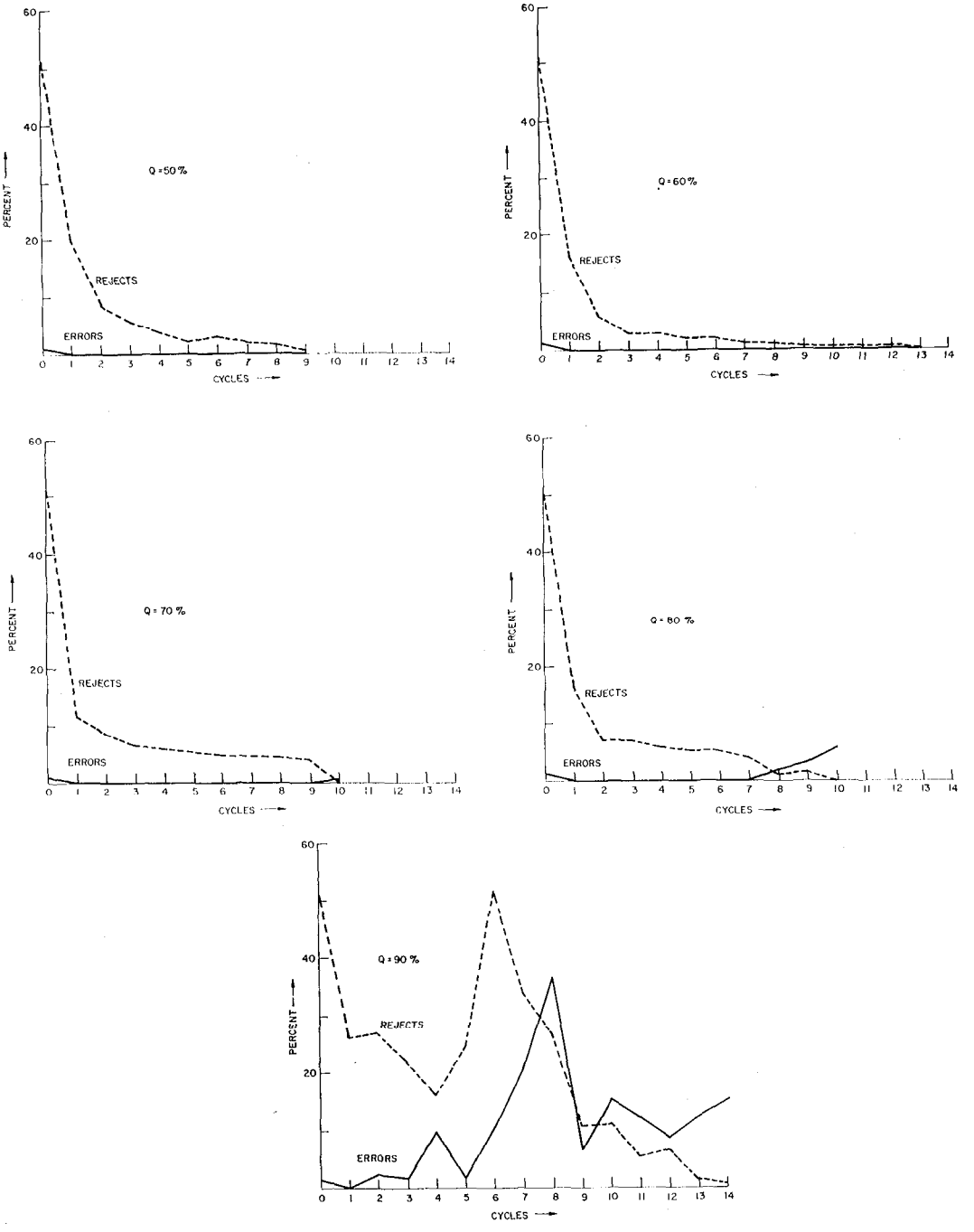

Fig. 2. Effect of level of quantization on convergence. The error and reject rates are plotted against the number of iterations for various values of Q, the quantization level of the ternary references.

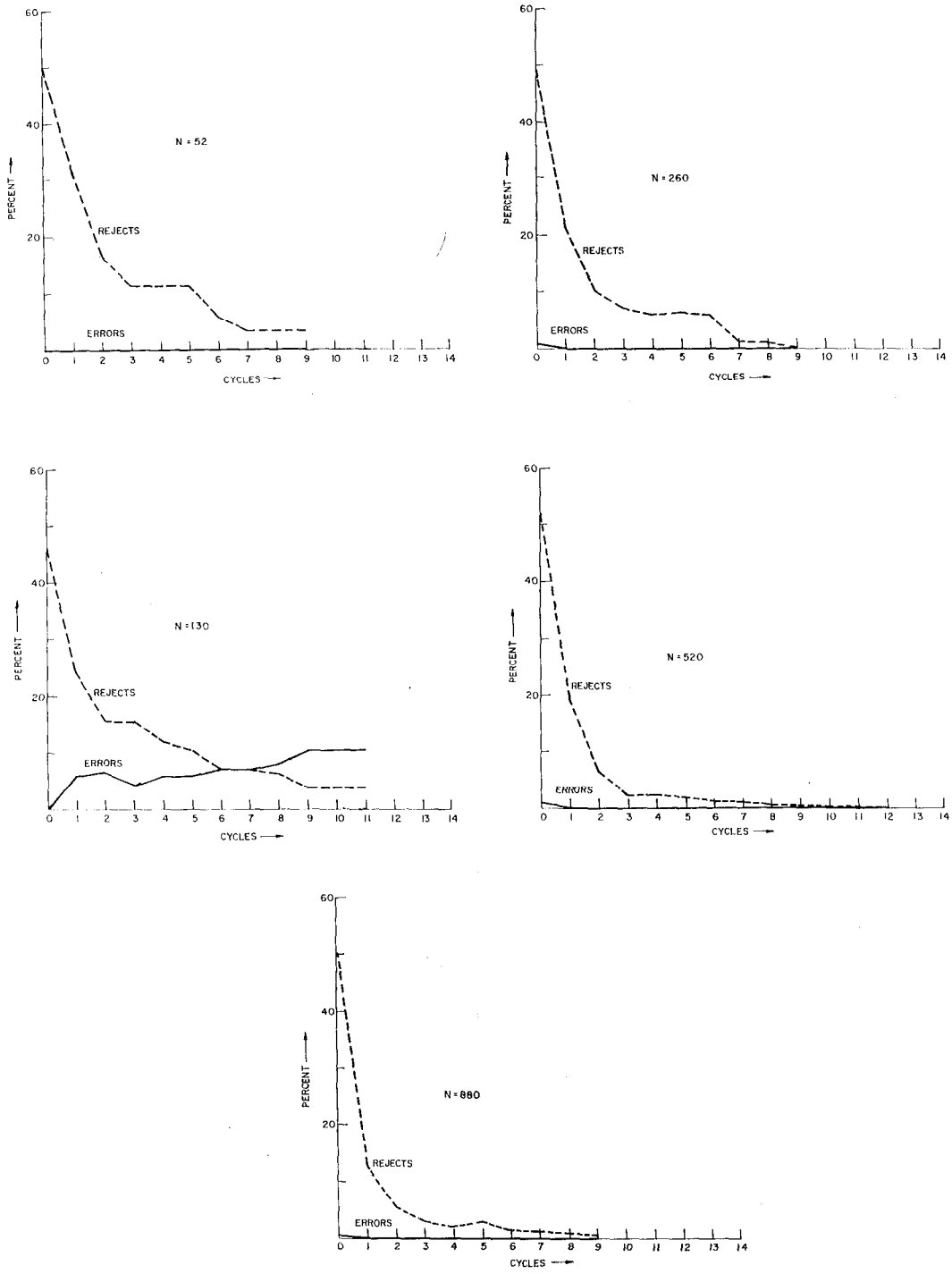

Fig. 3. Effect of number of training samples on convergence. The error and reject curves are shown for several values of N, the number of characters in the training loop.

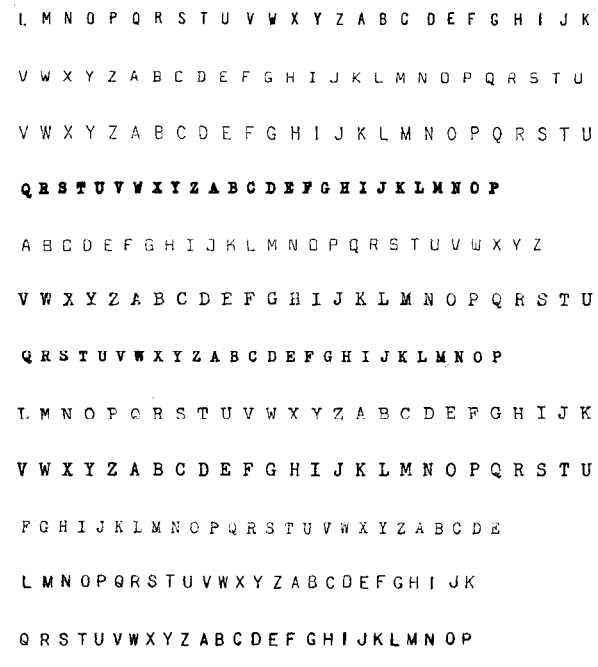

Fig. 4. Examples of impact printed characters. The twelve fonts used in several of the experiments are shown in the following order: Burroughs Posting Machine, Hermes Technical Pica, IBM Model B, L. C. Smith Elite, Olympia Senatorial, Remington Pica, Royal Elite, Smith-Corona Pica, Underwood Pica, Underwood Elite, IBM 403-1, and IBM 403-2.

Fig. 5. Fonts used in the preparation of initial references. From top to bottom: Hermes Techno-Elite, Selectric, Selectric Delegate, IBM Model B Artisan, Selectric Adjudant, Royal Manual Standard Elite, IBM Model B Courier, IBM Model B Dual Basic, and IBM Model B Prestige Elite.

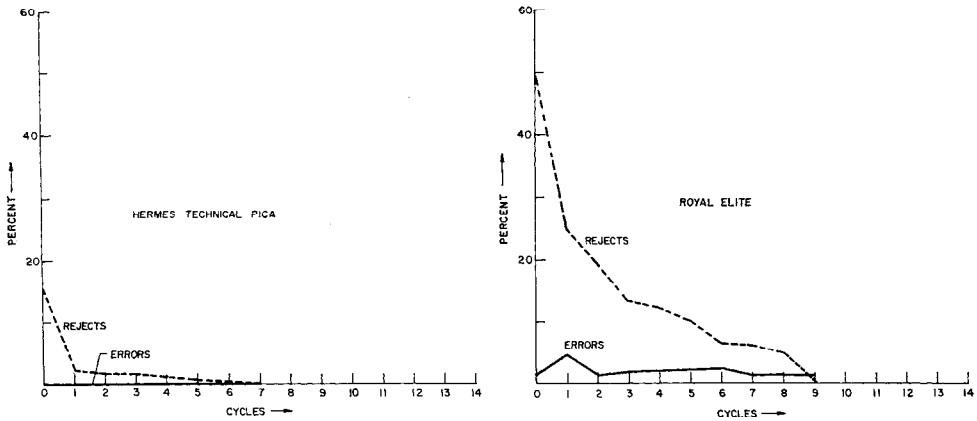

Fig. 6. Performance extremes with decreasing reject threshold. Best and worst convergence curves are shown for $F = 0.25$. The reject threshold decreases gradually from 5 to 0.

TABLE I

PERFORMANCE AFTER TRAINING WITH DECREASING REJECT THRESHOLD

Font	Self-Corrective Algorithm: 500-character training set, Error %	Self-Corrective Algorithm: 500-character training set, Reject %	Self-Corrective Algorithm: 1500 new characters, Error %	Self-Corrective Algorithm: 1500 new characters, Reject %	Initial References: ~2500 characters, Error %	Initial References: ~2500 characters, Reject %
Burroughs Posting Machine	0.0	0.8	0.2	1.5	4.8	18.0
Hermes Technical Pica	0.0	0.6	0.0	0.5	0.2	2.6
IBM Model B	0.0	1.4	0.4	0.9	1.2	8.8
L. C. Smith Elite	1.0	5.2	0.8	10.2	4.8	24.9
Olympia Senatorial	0.4	1.8	0.1	3.1	2.2	17.0
Remington Pica	0.2	1.4	0.1	4.0	2.8	16.2
Royal Elite	4.0	7.8	2.4	10.8	6.2	20.6
Smith Corona Pica	1.8	5.0	2.3	4.5	4.4	19.4
Underwood Pica	0.2	1.2	0.1	2.6	2.4	6.0
Underwood Elite	0.0	1.4	0.2	2.7	6.2	13.0
IBM 403-1	0.0	1.0	0.3	1.2	1.3	9.5
IBM 403-2	1.2	2.0	1.2	2.1	5.8	26.0
Average	0.7	2.5	0.7	3.7	3.5	15.2

TABLE II

PERFORMANCE AFTER TRAINING WITH REJECT THRESHOLD AT ZERO

Font	Self-Corrective Algorithm			
	500-character training set		1500 new characters	
	Error %	Reject %	Error %	Reject %
Burroughs Posting Machine	3.0	1.2	1.5	2.7
Hermes Technical Pica	0.0	0.4	0.0	0.6
IBM Model B	0.0	1.2	0.1	1.3
L. C. Smith Elite	1.8	8.2	0.8	14.5
Olympia Senatorial	0.4	2.4	0.1	3.3
Remington Pica	0.2	1.4	0.1	4.1
Royal Elite	1.2	6.4	0.4	8.7
Smith Corona Pica	1.4	1.4	1.3	5.3
Underwood Pica	0.4	1.8	0.0	2.6
Underwood Elite	0.0	0.2	0.2	3.0
IBM 403-1	0.0	0.8	0.1	1.9
IBM 403-2	0.0	0.4	0.0	0.7
Average	0.7	2.1	0.4	4.0

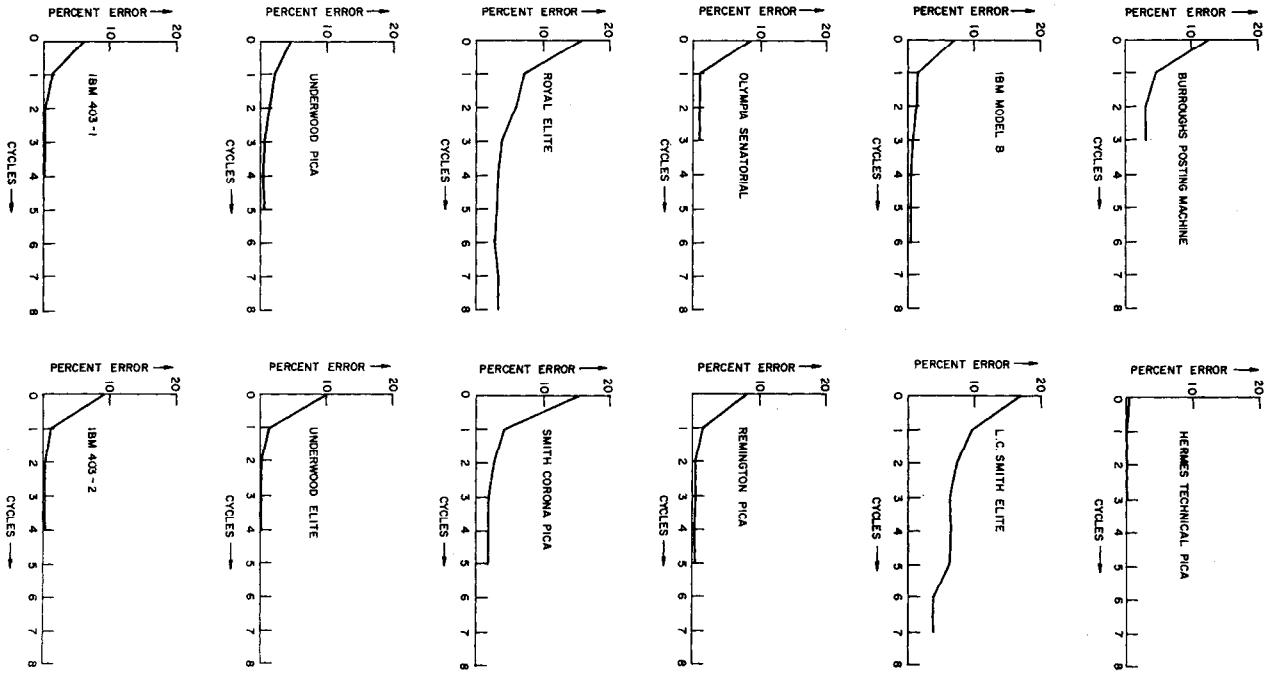

Fig. 7. Performance with zero reject threshold. These error rates, were obtained in the "forced decision" experiments. Rejects, which occur only when two references yield equal scores, are not shown.

TABLE III

DISTANCE BETWEEN INITIAL REFERENCES AND DESIRED REFERENCES IN TERNARY DIGITS, FOR VARIOUS FONT STYLES

Font	Mean Distance Tits	Maximum Distance Tits
Burroughs Posting Machine	29.5	50 (*S*)
Hermes Technical Pica	27.2	49 (*S*)
IBM Model B	26.5	45 (*S*)
L. C. Smith Elite	30.3	44 (*E*)
Olympia Senatorial	32.8	59 (*W*)
Remington Pica	29.8	51 (*M*)
Royal Elite	31.5	49 (*K*)
Smith-Corona Pica	29.8	44 (*Q*)
Underwood Pica	26.0	41 (*X*)
Underwood Elite	29.5	49 (*E*, *M*)
IBM 403-2	34.3	62 (*J*)

Discussion

We have shown that even with a mediocre set of initial decision parameters it is often possible to obtain improved ultimate performance by repeatedly reintroducing all accepted characters into a parameter computation algorithm.

Empty channels, or the fact that at first certain characters are not ever correctly recognized, do not necessarily prevent eventual convergence to low error rates. As performance in other channels ameliorates, likely confusions are resolved. The backward channels begin to accept a few characters; this, in turn, promotes further improvement.

Nor does unequal letter frequency distribution offer particular obstacles, as shown by the experiment on textual material. In an automatic reading machine it might be sufficient to iterate on the first few thousand characters until a satisfactory acceptance level is obtained on the most common characters. The machine may then modify the coefficients for the rare symbols as they appear.

For any but the most outlandish character sets, the self-corrective algorithm generates decision coefficients at least as good as those obtained by more conventional statistical or adaptive techniques. Thus, one obvious application is as a design tool which obviates the need for laboriously prepared labelled training alphabets.

In a recognition machine with modifiable decision parameters, the algorithm would lend itself readily to the processing of batched data. Experiments are now underway to test the range of the system with half a dozen or so starting references for each character, representing variations such as serif, sans-serif, bold face, and book face fonts, and changes in aspect ratio. The assumption that the input is "stationary" is of course retained, so that only a single set of decision coeffiicnets per character need be built up by the program.

Virtually the same approach is also being tried on hand printed numerals. Because it is difficult to obtain really good measurements on such materials, analog rather than ternary coefficients are used in the decision mode. The small number of classes involved and the relatively good recognition rates obtained by various invcstigators on systems tuned to a single person's writing [16], [18] allow some hope that the scheme may prove usable on documents containing only a few dozen numerals (in the same person's writing).

Despite the low occurrence rate of arrant misbehavior by the self-corrective algorithms, some form of performance monitoring is evidently highly desirable. In the case of hand printing, one would probably have to rely on some form of check digits, but for text reading the whole rich world of context is available. As a start, a letter and digram frequency estimator is being coded to give some measure of the correspondence between the digram frequencies found in text by the iterative character recognizer and the probabilities of such digrams in English text. A grave discrepancy here might require additional exception procedures or even outright rejection of the particular batch of documents under scrutiny.

Experiments of the type discussed in this paper cannot, of course, fully delineate the domain of applicability of the self-corrective algorithms. At most, we have driven a few experimental fence posts; whether the gaps must be staked in by trial and error, or whether an elegant analytical fence will one day neatly surround the whole area, remains to be seen.

Acknowledgment

The authors wish to express their appreciation to Miss Martha Miller for her extensive contribution in programming the algorithm, and to C. Marr and A. Sebastiano for performing the various experiments on the recognition system.

References

[1] T. W. Anderson and R. Bahadur, "Classification into two multivariate normal distributions with different covariance matrices," *Ann. Math. Stat.*, vol. 33, pp. 420–431, June 1962.
[2] A. M. Andrew, "A self optimizing system of coding," *Proc. Fourth London Symp. on Information Theory.* Washington, D. C.: Butterworth, 1961.
[3] H. D. Block, N. J. Nilsson, and R. O. Duda, "Determination and detection of features in patterns," *1964 Proc. Symp. on Computer and Information Sciences.* Washington, D. C.: Spartan, pp. 75–111.
[4] D. B. Cooper, "Adaptive pattern recognition and signal detection using stochastic approximation," *IEEE Trans. on Electronic Computers (Correspondence)*, vol. EC-13, pp. 306–307, June 1964.
[5] R. O. Duda and R. C. Singleton, "Training a threshold logic unit with imperfectly classified patterns," presented at the 1964 WESCON, Los Angeles, Calif., paper 3.2.
[6] S. C Fralick, "The synthesis of machines which learn without a teacher," Stanford University Tech. Rept. 6103-8, April 1964.
[7] W. H. Highleyman, "Linear decision functions with application to pattern recognition," *Proc. IRE*, vol. 50, pp. 1501–1514, June 1962.
[8] L. A. Kamentsky and C. N. Liu, "Computer automated design of multifont print recognition logic," *IBM J. Res. Developm.*, vol. 7, pp. 2–13, January 1963.
[9] C. N. Liu, "A programmed algorithm for designing multifont character recognition logics," *IEEE Trans. on Electronic Computers*, vol. EC-13, pp. 586–594, October 1964.
[10] M. Minsky, "Steps towards artificial intelligence," *Proc. IRE*, vol. 43, pp. 8–30, January 1961.
[11] E. A. Patrick and J. C. Hancock, "The nonsupervised learning of probability spaces and recognition of patterns," *IEEE Internat'l Conv. Rec.*, pt. 7, pp. 244–251, March 1965.
[12] R. J. Potter, "An optical character scanner," *J. Soc. Photographic Instrumentation Engineers*, vol. 2, pp. 75–78, February/March 1964.
[13] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms.* Washington, D. C.: Spartan, 1962., ch. 9.
[14] R. W. Sears, Jr., "Adaptive representation for pattern recognition," *IEEE Internat'l Conv. Rec.*, pt. 6, pp. 181–190, March 1965.
[15] O. G. Selfridge and U. Neisser, "Pattern recognition by machine," *Sci. American*, vol. 203, pp. 60–68, August 1960.
[16] W. Teitelman, "New methods for real-time recognition of hand-drawn characters," M.A. thesis, M.I.T., Cambridge, 1963.
[17] B. Widrow and F. W. Smith, "Pattern-recognizing control systems," *1964 Proc. Symp. on Computer and Information Sciences.* Washington, D. C.: Spartan, pp. 288–318.
[18] R. Wolfe, "Summary of handprinted character recognition workshop," unpublished memo., IBM Thomas J. Watson Res. Center, Yorktown Heights, N. Y., November 20, 1964.