the number of v-tuples which our method requires be investigated in order to determine the column designators of the representatives of all the nondegenerate equivalence classes. It is interesting to note that in at least one case, $v=4$ and $r=4$, no inspection of the rows obtained from any v-tuple is necessary since

$$\binom{2^{r-1}-1}{v}$$

equals Gilbert's number, (5).

In Table II, we present all of the rectangular arrays obtained when $r=4$ and $v=3$. Note that the 3-tuples (1, 2, 3), (1, 4, 5), (1, 6, 7), (2, 4, 6), (2, 5, 7), and (3, 4, 7) result in arrays having a pair of equivalent rows.

R. Bianchini
Ford Instrument Co.
New York, N. Y.
C. Freiman
IBM Res. Ctr.
Yorktown Heights, N. Y.
Formerly with Dept. Elec. Engrg.
Columbia University
New York, N. Y.

Further Results on the N-tuple Pattern Recognition Method*

Highleyman and Kamentsky, of Bell Telephone Laboratories, have given the results of some work[1] on the pattern recognition method (the n-tuple method) introduced by Browning and Bledsoe at the Eastern Joint Computer Conference, December, 1959.[2]

Evidently, Highleyman and Kamentsky did not understand that the parameter n (in the n-tuple method) should be chosen to best suit the particular data being read, because they used only $n=2$ in their computations. Some studies at Sandia Corporation in February, 1960, on these same data (provided by Highleyman) with $n=6$, 8 and 12 yielded results comsiderably different from those given. The result of some of this work is summarized in Tables I and II, where the numbers represent the per cent recognized.

The variability of the handwritten characters used in these studies is high and not well represented by the 50 alphabets used. For example, the last ten alphabets are considerably different than the first 40. For this reason, it will be necessary to have a much larger sample (perhaps 1000 alphabets) before one can decide with any certainty how successfully the n-tuple method will read characters with this much variability.

Other more powerful techniques, such as described in the original paper,[2] were not

TABLE I
Handwritten Letters and Numerals

n	40 Alphabets of Experience 10 Alphabets Read (Different)		50 Alphabets of Experience 50 Alphabets Read (Same)	
	Not Centered	*Centered*	*Not Centered*	*Centered*
6	25	32	71	80
8	29	40	87	86
12	31	Not computed	98	Not computed

TABLE II
Machine Printed Numerals

n	40 Alphabets of Experience 10 Alphabets Read (Different)	40 Alphabets of Experience 40 Alphabets Read (Same)
	Not Centered	*Not Centered*
6	94	100
8	98	100

tried on these characters but would undoubtedly improve these percentages.

W. W. Bledsoe
Advanced Research
Palo Alto, Calif.

* Received by the PGEC, September 16, 1960.

[1] W. H. Highleyman and L. A. Kamentsky, "Comments on a character recognition method of Bledsoe and Browning," IRE Trans. on Electronic Computers, vol. EC-9, p. 263; June, 1960.

[2] W. W. Bledsoe and I. Browning, "Pattern recognition and reading by machine," *Proc. EJCC*, pp. 225–232; December, 1959.

A Possibly Misleading Conclusion as to the Inferiority of One Method for Pattern Recognition to a Second Method to which it is Guaranteed to be Superior*

Highleyman and Kamentsky[1] recently reported having repeated portions of work done by Bledsoe and Browning[2] in a way that may lead the reader to what appear to be unfortunate conclusions. Bledsoe and Browning identified input patterns by matching the states into which they threw randomly chosen 2-tuples with similar lists of states for previously processed patterns. For each state, a 1 was stored if any example of the pattern in memory had ever thrown that 2-tuple into that state; otherwise, a 0 was stored. Bledsoe and Browning reported 78 per cent success with this method over an array of five different hand-printed alphabets. Highleyman and Kamentsky report only 20 per cent success over 50 different alphabets hand-printed by 50 different people. The second experiment does appear to indicate the weakness of this method, in its present state of sophistication, as soon as restrictions on transformations over the input matrix are relaxed. (In the first experiment, one presumably friendly person, while in the second, 50 quite likely skeptical people, prepared the input materials.) This is precisely what Kirsch,[3] in his discussion of the first paper, had suggested.

But a comparison test was made with a second method which, on the same set of inputs, gave 77 per cent success. Not enough detail is given to make absolutely clear what this method was. But the report makes it sound like a method that would, in fact, have Bledsoe and Browning's success rate as a theoretical upper limit. This, of course, is not quite an accurate statement, as the empirical results indicate. But the point is that the superiority of the second method could not possibly lie in its basic logic, but must, therefore, result only from its retention and use of a greater amount of information about previously processed patterns in its memory lists. That is, information and correlation methods that could just as well be used by either method were used only by the second, and the improvement in the second guarantees at least as great improvement in the first. Thus, the comparative experiment does not show that one logic was superior to another, but does show that additional use of information by one method led to great improvements in results.

The Highleyman-Kamentsky procedure "... involves the comparison of an unknown input pattern ... to a set of average characters. The average characters are described by a set of 12×12 matrices (one for each character) in which each element represents the probability of occurrence of a mark in that element for the character which it represents." This statement is taken to mean that over the 50 examples of the alphabet given the program in its learning phase, for each cell in the input matrix, the percentage of times that cell had been filled by the input pattern was stored as its probability for that pattern. For the Bledsoe-Browning method, the one of the four possible states of the $N/2$ random 2-tuples of cells in the matrix was given a probability of 1 if *any* of the five examples threw that 2-tuple into that state. Thus, the differences between the two methods appear to lie in 1) looking at 1-tuples as opposed to 2-tuples, plus the extraneous differences (in that they could be used with either method), 2) computing probabilities on a scale with n intervals rather than possibilities on a scale with 2 intervals (possible vs never yet), 3) using 50 vs 5 previous trials, and 4) examining probabilities vs cross correlating.

But the first difference is not a difference at all for the state of two randomly conjoined cells is simply the state of the conjunction of the same two individual cells. The 2-tuple and 1-tuple methods should, in fact, give identical results in the case of one alphabet. When several variant alphabets are learned, the 2-tuple method is bound to be superior, simply because information is stored not only about which state of a single cell is produced by an input pattern but also about which state of a randomly chosen second cell is produced in conjunction with this first cell state by this pattern.

The Highleyman-Kamentsky method is, in fact, exactly the same as the "1-tuple method" as used by Bledsoe and Browning in their original experiments, and the methods of among others, Uttley and, probably, Rosenblatt. Bledsoe and Browning ran several comparison experiments, in which they found what could be predicted theoretically—that with one alphabet there is no differ-

* Received by the PGEC, October 8, 1960; revised manuscript received, October 27, 1960.

[1] W. H. Highleyman and L. A. Kamentsky, "Comments on a character recognition method of Bledsoe and Browning," IRE Trans. on Electronic Computers, vol. EC-9, p. 263; June, 1960.

[2] W. W. Bledsoe and I. Browning, "Pattern recognition and reading by machine," *Proc. EJCC*, pp. 225–232; December, 1959.

[3] R. A. Kirsch, "Discussion of problems in pattern recognition," *Proc. EJCC*, pp. 233–234; December 1–3, 1959.

Authorized licensed use limited to: CUNY- Graduate Center. Downloaded on October 20,2020 at 02:10:04 UTC from IEEE Xplore. Restrictions apply.

ence between the two methods (unless, as was apparently the case in the original Bledsoe-Browning work, a correlation method that happens to favor one over the other is chosen to assess similarity), but with five alphabets the 2-tuple is clearly superior to the 1-tuple.

There is a great need for stringent tests and comparative studies of different pattern recognition methods. But an experiment should make explicit what are the factors being varied, and lead to unambiguous statements as to the sources of effects demonstrated. In the present case, it would seem that Highleyman and Kamentsky have demonstrated the limitations of the basic 2-tuple method. But they have also demonstrated ways whereby it can be strikingly improved. The most important conclusion to be drawn from their replication would seem to be that if the 1-tuple method can be made to work so well, so easily, then it is to the larger n-tuple methods, which are guaranteed to work even better, that these improvements should be made.

LEONARD UHR
Mental Health Res. Inst.
University of Michigan
Ann Arbor, Mich.

Further Comments on the N-tuple Pattern Recognition Method*

The primary purpose of our original letter[1] was to dispel a false conclusion to which a reader might be led by Bledsoe and Browning's paper,[2] *i.e.*, that a machine based on $n=2$ is sufficient for the recognition of hand-printing with an accuracy of 80 per cent or so. We considered these results to be somewhat misleading because their limited data source was not described in the paper. We are happy to note the greatly improved results which Bledsoe and Browning obtained with higher n when operating on our data, since we do feel that their method has merit when applied properly. The value of n required is quite important, however, since the complexity of the resulting machine, as measured by the number of memory cells required, increases almost exponentially with n.

With regard to Dr. Uhr's comments, I would like to make the following observations.

1) We chose the correlation method because we felt that it was based upon an easily understood technique. Such a technique would indicate to some extent the variability of the data to which it was applied.

2) The correlation technique which we used is not equivalent to Bledsoe and Browning's method for $n=1$, in which the memory matrix is comprised of the probabilities of occurrence of the various states. The difference lies in an appropriate normalization of the probabilities in the correlation technique such that the sum of the squares of the probabilities in a particular matrix is unity. Bledsoe and Browning simply added the unnormalized probabilities. It is a simple matter to construct examples for $n=1$ which show the need for proper normalization. For example, consider two patterns represented by a two-element matrix, as in Fig. 1(a). Assume that the noise characteristics are such that the unnormalized probability matrices are as shown in Fig. 1(b). Obviously, using these matrices, an ideal pattern A will always be classified as pattern B. However, if both matrices are normalized as described previously, shown in Fig. 1(c), the ideal patterns are classified correctly.

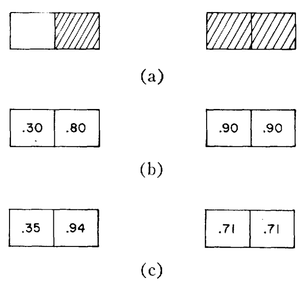

Fig. 1—Two-element matrix. (a) Pattern A, left, and pattern B, right; (b) unnormalized probability matrices; (c) normalized probability matrices.

3) We feel that the use of probabilities rather than binary weights would improve the method of Bledsoe and Browning; this is a point which they also made in their paper. In fact, we attempted recognition using 2-tuples where the memory matrix consisted of the (unnormalized) probabilities of state occurrences based on 50 samples of each hand-printed character (the same data as were used to construct the probability matrices for the correlation test). The recognition rate was improved from 19.7 per cent with the binary matrix to 30.7 per cent with the probability matrix. However, it can be argued that the need for proper normalization (as discussed above) is also existent for $n>1$. The problem of whether a meaningful normalization exists for these cases is yet to be studied. The normalization argument, incidentally, holds also for a matrix composed of binary weights.

4) Dr. Uhr's comment that the correlation technique retained more information than the method of Bledsoe and Browning in the case of a binary matrix is a good point. However, in the above experiment, the 2-tuple method retained as much information (in fact, more information, since the probabilities of pair states were retained) as the correlation method. Yet it still resulted in significantly poorer performance (30.7 per cent recognition rate vs 77.2 per cent), perhaps because of the lack of an appropriate normalization. We had also tried other random arrangements of pairs, with essentially the same results.

W. H. HIGHLEYMAN
Bell Telephone Labs., Inc.
Murray Hill, N. J.

* Received by the PGEC, November 30, 1960.

1 W. H. Highleyman and L. A. Kamentsky, "Comments on a character recognition method of Bledsoe and Browning," IRE TRANS. ON ELECTRONIC COMPUTERS, vol. EC-9, p. 263; June, 1960.

2 W. W. Bledsoe and I. Browning, "Pattern recognition and reading by machine," *Proc. EJCC*, pp. 225–232; December, 1959.

Computer Model of Gambling and Bluffing*

I wish to outline a project as yet not complete, which may be of some interest.

The machine simulation of human behavior in the mental states of uncertainty, such as estimation, prediction, choice, risk-taking, decision-making, makes more comprehensive these difficult conceptual and logical problems for the social scientist, psychologist, military strategist, etc.

Interesting studies can be pursued with digital computers on the playing of games.[1–8] An important subclass of games is the one in which the players make probability judgements, and can have hidden plans, etc.—in contrast to the games in which the information on the previous history and present position is perfect. Fairly exact experimentation would be possible with a poker-playing machine since here a human opponent's motivated responses are primarily controlled by simple numerical properties of the stimulus situation. Such a program may serve as a model of human gambling and bluffing in business competition, critical military situations, etc., by describing the objective vs subjective probability scales of conservative, mathematically fair (if any), and extravagant players. We could explain, for example, why and how gamblers characteristically overvalue long shots (low probability of high winnings) and undervalue short shots (high probability of low winnings).

A sketchy flow-chart of a poker program under construction can be seen on Fig. 1. The game is a variant of Draw-Poker, known as "open on anything." For the sake of simplicity, the step of paying the ante is left out; moreover, the opponent always makes the first bid. The steps are as follows:

1) Deal 5 cards for each, the machine and the opponent.

2) Calculate[9] the optimum number of cards to be exchanged by the machine at the second dealing n_{opt}; moreover, calculate the expected value of the probability of the machine's winning after the second dealing $E(p_2)=p_1$.

3) The opponent has bid M chips.[10]

* Received by the PGEC, October 10, 1960.

1 A. Bernstein, *et al.*, "A chess playing program for the IBM 704," *Proc. WJCC*, Los Angeles, Calif., pp. 157–159; May, 1958.

2 N. V. Findler, "Some remarks on the game 'Dama' which can be played on a digital computer," *Computer J.*, vol. 3, pp. 40–44; April, 1960.

3 N. V. Findler, "Programming games," [Pt. (a) of Paper BI 3.3], Summarized Proc. of the First Conf. on Automatic Computing and Data Processing, Australia; May, 1960.

4 J. Kister, *et al.*, "Experiments in chess," *J. Assoc. Computing Mach.*, vol. 4, pp. 174–177; April, 1957.

5 A. Newell, "The chess machine," *Proc. WJCC*, Los Angeles, Calif., pp. 101–108; March, 1955.

6 A. Newell, *et al.*, "Chess-playing programs and the problem of complexity," *IBM J. Res. & Dev.*, vol. 2, pp. 320–335; October, 1958.

7 A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM J. Res. & Dev.*, vol. 3, pp. 211–229; July, 1959.

8 C. E. Shannon, "Programming a computer for playing chess," *Phil. Mag.* (7), vol. 41, pp. 256–275; March, 1950.

9 Since, in the general case, when the whole stock of cards is played off before a new shuffling takes place, tabulated probabilities are obviously awkward and cumbersome, the Monte Carlo technique is to be used with the steps 2 and 8.

10 The notations M and $M+N$ always represent the current value of chips in the pot, regardless of how many bidding cycles have lead to it.

Authorized licensed use limited to: CUNY- Graduate Center. Downloaded on October 20,2020 at 02:10:04 UTC from IEEE Xplore. Restrictions apply.