Discussion of Problems in Pattern Recognition

W. W. BLEDSOE, J. S. BOMBA, I. BROWNING, R. J. EVEY, R. A. KIRSCH, R. L. MATTSON, M. MINSKY, U. NEISSER, AND O. G. SELFRIDGE

Various problems encountered in pattern recognition were examined by an eight-man panel during the Thursday afternoon session. The panel included O. G. Selfridge, M.I.T. Lincoln Laboratory, Session Chairman; R. A. Kirsch, National Bureau of Standards; M. Minsky, Massachusetts Institute of Technology; U. Neisser, Brandeis University; and the authors of the four preceding papers — R. J. Evey, I.B.M. Corporation, "Use of a Computer to Design Character Recognition Logic;" R. L. Mattson, Lockheed Missiles and Space Division, "A Self-Organizing Binary System;" J. S. Bomba, Bell Telephone Laboratories, "Alpha-Numeric Character Recognition Using Local Operations;" and W. W. Bledsoe and I. Browning, Sandia Corporation, "Pattern Recognition and Reading by Machine." Following comments by Messrs. Kirsch, Minsky and Neisser, the panelists answered a number of questions from members of the audience.

R. A. KIRSCH

I WOULD like first to make a few technical points. The first comment relates to the connection between the Bledsoe-Browning paper and the Mattson paper and the relationship between these two papers and some of the other work in this field that you may know about. If we have an input retina in a pattern recognition device which has n cells, then a pattern may be defined as a Boolean function of the n variables. Each of the various representations of a pattern is another one of the terms in the disjunctive normal form of the Boolean function which characterizes the pattern. This is substantially the way in which Mattson looks at the notion of a pattern. Mattson's criterion for similarity between two patterns is the usual metric of the symmetric difference of the two patterns or, what is the same, the area of the modulo-two sum of the two patterns. This measure has, of course, useful mathematical properties and leads to an interesting and perhaps fruitful analysis, but the important relevant question is whether this particular criterion for similarity corresponds to any very important class of pattern similarities. The evidence that Mattson offers shows that at least some of the types of similarity that we would like to attribute to sets of patterns are, in fact, reflected in his measure of similarity.

The economy in terms of number of devices required for a Mattson type of self-organizing pattern recognition system is determined by the extent to which the pattern classes represent linearly separable Boolean functions. Interestingly enough in the examples that Mattson gives, the functions appear to consist of single clusters of points in Boolean n-space. However, we know that there are important pattern classes which are not so economically realized by a Mattson type of device. In fact, Mattson recognizes that the alternate symmetric function is a very difficult one to implement in terms of number of devices required. The alternate symmetric function happens not to correspond to an important type of pattern, but certain other functions do; for example, a function whose disjunctive form contains all terms having exactly half of the variables complemented and half of them uncomplemented. This corresponds to the pattern class which has 50% black points and 50% white points. This, although not as difficult to implement with a Mattson device as the alternate symmetric function, nevertheless is reasonably difficult and perhaps corresponds to an important pattern class.

Bledsoe and Browning give some detailed discussion of the effects on their recognition program of changing the size of n for their n-tuples. It is interesting to note that the case of n equals 150, which they reject on the basis of the size of memory required to implement it, is the case that corresponds to the situation chosen by Mattson for his recognition scheme. Mattson gets around the large memory requirement by the use of his similarity criterion which, as I pointed out before, is somewhat debatable in terms of its usefulness. Bledsoe and Browning quite correctly point out that for a case of $n = 1$ and patterns subject to all possible translations the memory will saturate. Their discovery that for $n = 2$ saturation does not occur can be related to the fact that characters subject only to translation have a constant autocorrelation function, and a set of patterns with a constant autocorrelation function do not cause saturation for $n = 2$. If we allow the patterns also to change linearly in size, or also to rotate, the autocorrelation function changes and hence $n = 2$ saturation becomes possible. Again we are confronted, as we were in the Mattson paper, with an experimental question: "Does the autocorrelation function of a character tell us more about that character for purposes of recognition than does the character itself?"

It seems likely that we can invent important pattern classes that will confound the autocorrelation function method of recognition, which is implicit in the Bledsoe-Browning approach, just as we can do similarly with the Mattson approach. This should not be taken as an implied criticism of either of these approaches. Both of them have implicit in them, and, in fact, all pattern recognition schemes have implicit

in them, certain built-in heuristic criteria which make them useful for certain types of pattern recognition problems. Since no pattern recognition scheme is truly universal, and we need not expect that any scheme will be truly universal, the fact that a heuristic will work in certain situations and will not work in others need not necessarily be a shortcoming. The significant question is how important that heuristic will be for the problems that are to be solved. It is for this reason that reference such as has been made, not in these papers but in other papers in the field, to general purpose pattern recognition devices seems at least misleading and probably just not true.

I would like also to make a few remarks of a less technical nature. The fact that there are some strong similarities between at least two of the papers in this session, and that these similarities are superficially not obvious, leads one to feel that what is needed in the pattern recognition field is at least some type of unified terminology and perhaps even a unifying theory which will make comparison of results simpler. It is not superficially obvious, although I nevertheless believe that it is true, that the two types of devices that we have seen here can accomplish the same type of recognition and no more nor less, for example, than that of the Perceptron device of Rosenblatt. If this is true, it is certainly a fact made more obscure by Rosenblatt's talk of neurons, Mattson's talk of Boolean functions, and Bledsoe and Browning's talk of n-tuples in a photocell mosaic. This lack of a unifying terminology may perhaps explain why so many of the workers in the pattern recognition field fail to give credit to their predecessors, who very often have contributed useful ideas.

Another possible explanation is the generally mistaken notion that proprietary interests in pattern recognition research must be protected because of important commercial applications; for example, to the problem of character recognition machine development. We have seen here in this session the example of Mr. Evey of how a successful character recognition machine may be developed without the necessity of being concerned with fundamental pattern recognition problems. Conversely we have seen in such papers as the Bledsoe and Browning paper, the Mattson paper, and, perhaps to an extent, in the Bomba paper that pattern recognition research can continue without necessarily contributing very directly to the development of character recognition machines. Confusing pattern recognition research with specific machine development does not accrue to the benefit of either of the two types of programs.

There is one final point that I would like to make. Pattern recognition research has been, I think, to some extent held back by a lack of proper data in several of the research efforts. About two years ago at the Eastern Joint Computer Conference in 1957, when I gave a paper on picture processing, I offered to the workers in this field the data that we at the National Bureau of Standards had generated with our picture digitalizing device. It was disappointing then and still is to see workers in the pattern recognition field preparing data by hand and attempting to investigate a problem which is so largely experimental in nature with data that is quite inadequate. What one needs for successful prosecution of pattern recognition research is large quantities of information in machine form and perhaps automatically generated information. I would like again to offer our service for helping make this data available to people in the field, and I would invite any of my colleagues who have such data to do the same. Where such data is easily generated and can be made available to active workers in the field, the general status of research in pattern recognition can be considerably advanced.

M. MINSKY

I would like to make some remarks on how the character-recognition systems presented today may be related to more general pattern-recognition problems.

What are patterns? It is hard to define precisely this intuitive concept but I think we will agree first that the things we call patterns are *classes* of signals or figures, not single figures. They are classes of figures which are grouped together because, for some purpose or other, they can be treated alike. Now in the character-recognition problem it is perfectly clear in what sense the figures so grouped are to be treated alike. In the case of *printed* characters the patterns have a structure derived from the manner in which the figures are produced. In each case one starts with an ideal "prototype." To read a text requires that one decide, for each image on the paper, which of the prototypes was responsible. Each of the proposed systems computes the values of a set of functions of the image. Once these values are computed the system faces a statistical inference problem: on the basis of the evidence (the set of computed values) what is the prototype most likely responsible?

The figures are related to the prototypes by various kinds of noise and distortions. Simplest, perhaps, are the "additive" noises, in which pigment is added to or subtracted from the true image in a manner independent of the image. For these distortions one may do well with a simple correlation or area-matching analysis. If the noise is not extreme, or if the ensemble of patterns is small, it may be possible to confine the analysis to just a few of the points, or pairs of points, etc. With more complicated figure-dependent distortions (for example, smearing) more complicated functions may be necessary.

In the case of more global distortions involving, for example, variations in size and position, the simple area-matching tests will give poor results.

There is a question in my mind about the extent to which some of the methods discussed here can handle, or be easily extended to handle, that kind of problem. In the Bledsoe-Browning system the figures are constrained so that rather little translation is admitted. The method requires discovery of *simple* combinations which separate patterns, and these will be rather hard to find as the admitted distortions become more extreme. And when we look beyond prototype-derived patterns to, for example, patterns defined by connectivity, I think we will find that one needs to go to essentially more complicated functions; one cannot get by with larger numbers of simple ones.

The method of Mattson is useful, it seems to me, only where the area-matching kind of comparison is appropriate. The technique of Evey works on translated figures because the figure is actually "rolled" or scanned in Pitts and McCulloch fashion through the vertical translates after being normalized with respect to the horizontal translates. This seems to give excellent results for the chosen problem, but it really is difficult to make a case for the scanning method when faced with more complex classes of distortions.

The Bomba system has, I feel, rich possibilities for extension to harder problems. For it is based on the use of complicated but rather meaningful properties of the figures. The "meaningful" properties of figures can be used to define and recognize patterns based on ideas more abstract than that of the distortion of prototypes. (I certainly do not mean to imply that even the printed-character problem is trivial. Consider, for instance, the examples illustrated in Mary Stevens' Bureau of Standards report.)

In each model there is an inference stage in which one combines the values of a number of rather independent functions of the figures. One must require that the functions be at least partly insensitive to the differences between instances of the same pattern — otherwise the system will be overwhelmed by the need for a special treatment of each case. In the case of systems which generate test functions at random, through net connections or the like, we can expect that most of the functions defined will be useless or nearly so and a learning procedure will be needed to separate out and weight the relatively good ones. When the patterns in question are very complex, the chance of finding any significant functions at all in this way becomes prohibitively poor, and such systems certainly cannot "handle all kinds of patterns with equal facility" except when one is restricted to patterns of equal simplicity.

The papers suggest different ways to combine the results of the different tests or functions. Evey's system sets up fourteen "character triggers" with different tests and requires a stringent matching for one and only one of them — this harsh requirement is, of course, perfectly appropriate to the problem he is working with (namely, handling money). In the Bledsoe-Browning system one compares scores based on summations of contributions from many tests. They describe at least two ways of combining many fragments of evidence, none particularly decisive alone, to obtain conclusions which are conclusive and accurate. (A rather general discussion of "parallel" methods of combining evidence from many tests can be found in a paper of Selfridge, whose "Pandemonium" computer concept embraces many of the models being explored these days.)

In the Bomba system, with its decision tree structure, we find a rather different way of combining evidence. As Dr. Neisser will note, a program which has been run at our laboratory uses test functions which are not very remote from those of Bomba, but which are combined in a parallel decision rather than a sequential tree method. This program, due to W. Doyle, is of interest here, I think, in that it can be regarded as using tests related to the kind in Bomba's program, a decision method not far from that of Bledsoe and Browning, and with, I believe, comparable success. One can use the decision tree structure only when one is very sure that the decisions at the top of the tree are very reliable — of course, in that case one can avoid all the computations in the discarded portions of the tree. Otherwise one has to obtain in some way the kind of redundancy supplied by the parallel methods.

That is all I have to say, in general. I find I can't refrain from remarking (with all due apology to Mr. Evey, who is not responsible) that the choice of type-face "E13" in the ABA magnetic-ink system seems unfortunate to me. I don't find the characters very legible. Speaking as a human being I will grant that we do seem to have here a solution to the problem of character-reading-by-machine, but we are left with something of a problem in the way of character-reading-by-people!

U. Neisser

I will try to do a couple of things. One is to distinguish the two basic types of program in this field; the other is to emphasize the things they have in common which are worth considering in future work. The types I have in mind are those mentioned by Minsky: decision trees and parallel processing.

It does seem that where you cumulate separate decisions you are more apt to make mistakes. Bomba's program was the only one of these four involving a tree; you could see it most clearly in the diagram of

decision logic that he showed. Other tree programs have been reported in the literature, however. I think all of them will have great difficulty in learning from experience. If the program makes a wrong decision at any fork it will necessarily be wrong at the end, with no obvious way of finding out what should be changed in order to improve performance. In Bomba's program, for example, the entire decision process is spelled out in advance by the programmer; the machine will never improve on his original descriptions of the characters.

In fact, all the tree programs have used quite well-printed characters as inputs. If the characters were made a bit sloppily, or slanted, the program could no longer distinguish one from the other. I would worry about that if your objective is to recognize hand-printed characters: *really* hand-printed, as when you ask somebody to print their name and address. If you look in detail at what people do under these conditions it is miserable.

In parallel programs the basic idea is to take a lot of intermediate functions of the stimulus pattern, and then make a decision on the basis of some more or less sophisticated maximum procedure. It seems to me that there are two important differences among parallel programs. The first is the degree to which the features used are shape-dependent rather than position-dependent. A program like Evey's is extremely position-dependent. His particular sure-bits are either on or off, and that's it; the shape of the input character is irrelevant except as it happens to fill particular positions. To some extent this amount of position-dependence characterizes the typical "neural-net" programs as well. In some of these, however, there is a good deal of preprocessing done; the character is centered and sized before the neural net takes over. One of the exciting things about the Bledsoe-Browning paper is that in considering pairs of bits, and larger n-tuples, their program is partly shape-dependent. The vertical *I* in the 7×10 matrix never excites bits on the left and the right side simultaneously because of its *shape*, not because of its original position.

The other important aspect of parallel programs is the sophistication of the decision process. The simplest procedure is to compute a lot of functions like the outputs of the n-tuples in Bledsoe-Browning and take the one with the highest score. They have done much more than that. In one of their procedures they look at the entire distribution of outputs rather than only at the largest; in another, as we have heard, they successfully introduced context into the decision process. There is a Lincoln Laboratory program for character recognition which is also parallel. It uses a large number of shape-dependent operations: features like Bomba's, directions of curvature, and things of that sort. It computes the outputs of all of these tests in parallel, and uses a very simple decision procedure. Even with this, it performs quite well.

My worry about Bledsoe and Browning's program stems from the small size of their matrix. It's hard to know how their system would perform if they introduced anyone's handwriting besides Ibn Browning's! Indeed, one of the impressive things about the character recognition problem is that you never know if something will work until you try it. Theory doesn't help; until a program is running you cannot know whether it will saturate or run out of storage or what have you, when confronted with sloppy characters. This may be a sign that we have a problem genuinely related to the simulation of higher mental processes.

Let me turn now to some of the similarities among the programs. One is the emphasis that several of these papers (as well as others in the literature) have shown in recognizing hand-printed characters rather than machine print. Let us all take warning from Mr. Evey, who showed how difficult even the IBM-font is if you have high standards of accuracy.

A second point is that most of these programs use some amount of pre-processing before they get into the real work. Bomba thins out lines and cleans up stray points. Unger's program, reported elsewhere, does similar things; so does the Lincoln Lab program. Evey moves the characters to one side of the matrix; Bledsoe-Browning have experimented with shifting them to one corner. Roberts' perceptron-type program, at M.I.T., uses centering and scaling. In other words, these programs involve two levels of processing. I think this will become increasingly important as we get to the more complex patterns that we would like to recognize successfully. Suppose, for example, that you are dealing with triadic patterns and you would like the computer to select that part which is maximally different from the other two. Such a task can't be accomplished with present types of programs; a different sort of analysis is necessary. You have to do several stages of processing, and I think that is what we are getting into.

One last point of similarity is a deficiency the programs have in common. None of them has yet looked at the problem of segmenting words into letters, either in cursive writing or in print. Yet you cannot actually read until you have some idea where one letter ends and the next begins. Even Bledsoe and Browning, as I understand, computed one letter at a time in their work with context. We will have to tackle segmentation pretty soon if character recognition is to be realistic; if it is to compare with the way people perform. It is quite clear, if you take handwritten characters (at least in my handwriting) that it is impossible to identify a letter directly. What you actually recognize is the word itself. So far, no paper that I have seen has faced this problem.

Discussion

Mr. Selfridge: While the question cards are being collected, I will give you a couple of biographies available in this field. The first is by Otis Minot, U. S. Navy, San Diego, California. This is PM 364, "Automatic Devices for the Recognition of Two-Dimensional Patterns." The other is by Mary E. Stevens of the National Bureau of Standards. The number is 5463, and it is called, "A Survey of Character Recognition."

J. K. Moore (Smith, Kline & French): How long did it take the 704 to recognize the sentence?

Mr. Browning: We did not attempt to recognize sentences per se; rather, we scored letters, then, with the context program, we scored words. At the beginning we were recognizing on the average one letter every 3/5 of a second. This is because, in order to understand the discrimination process, we had an elaborate print-out of the relevant data. We believe that this time could be reduced to perhaps a few hundredths of a second per letter. For contextual recognition of words, the time requirement is increased and will depend on the size of the dictionary.

D. Baumann (MIT): How does your device locate the character at the right side of trigger matrix before its recognition?

Mr. Evey: The character actually comes into the matrix from the left hand side and is moved across the matrix — we can think of it as this — and the machine decides that it has the character in the matrix when it sees any two bits in the first column.

E. B. Cohen (Auerbach Electronics): Have you considered other kinds of character distortion besides poor printing? Can the 1210 sorter read tilted or displaced characters?

C. E. Dorrell (IBM) What work is being done or what is the value of observing the profile density or edgeview of the characters?

Mr. Selfridge: If you take a character and project it to one side, you get a one-dimensional distribution with many fewer bits, which is presumably a good thing in processing. A number of people have considered this — Gerald Dinneen and I, for example, some years ago. We never did anything about it. It didn't look that advantageous. I think that the kind of program arising and being considered today might find projection an acceptable technique. I don't know of a program carrying through techniques like that today.

G. A. Barnard (Ampex): Please expand on the contextual positioning aspect of your method.

J. J. Murphy (Sylvania): Were the *n*-tuples ever other than randomly associated?

Mr. Bledsoe: Let me take the second question first. Since we had no preconceived indea of what patterns we wanted to recognize, we felt that randomly associated *n*-tuples would be a logical starting point. In a couple of test cases we found the non-random associations did not discriminate as well. We feel that, for a particular family of patterns to be read, some particular non-random association would be optimum.

Mr. Selfridge: Another general point was brought up about segmentation. This is going to be a real problem on this. Very few people hand print. The only ones I know are people who go to Radcliffe; they do tend to hand print. This is as in speech recognition; in identifying words you are aware that segmentation is the real primary problem.

M. Jacoby (Remington Rand): What is the spacing of the individual heads on the reading element?

R. P. Niquette (Ramo-Wooldridge): What kind of character rates are to be expected from the system you described? Failure rates?

Mr. Evey: The area of the check that has to be read is about a half inch across the bottom of the check. Actually in the IBM system there are 30 tracks which cover this half inch and you get it down to 10 tracks by tying every tenth track together, so you have three tracks together. So you actually have a positioning problem in the matrix. This is mentioned in the paper. I didn't want to take time to discuss this in the talk. So you have 30 tracks covering this half inch and each track ends up about 17.5 thousandths wide. There is a dead space in between each track of about six thousandths. The failure rate is actually part of the problem here because the specifications laid down by the ABA, where they wanted machines used by banks, were something less than one-tenth of one per cent reject and about a tenth or so of that for substitution. That is, the actual rates as originally spelled out were one reject in every 2,500 checks, which figures out to four-tenths of one per cent and about a tenth of that for substitutions, which figures out to about one substitution in a million characters.

Mr. Neisser: Won't serifs confuse your technique? How will you handle them?

Mr. Selfridge: Is the trèé decision technique absolutely essential to your problem?

Mr. Bomba: No, the tree decision technique isn't essential to the program. The basic idea that I presented was to extract figures. Now, whether I use a tree method or some sort of statistical method with features as the input variables, is unimportant. The use of the tree here did enable me to show that I could recognize all of the characters in the alphabet. This particular type of logic which I chose in order to illustrate the feature extraction process was arbitrary.

With regard to serifs and other types of distortion, in some cases, particularly typewritten characters, the addition of a serif will be quite significant in changing the type of the character from a feature viewpoint. Actually the character might vary considerably, and thus the recognition procedure might well call a character such as a "T with serifs on it" by another name until the final recognition is done. So serifs do make a difference. However, small distortions, small serifs, would tend to be ignored by the feature extraction process. Seriphs cannot be ignored as they are often as large as the features which distinguish two characters. For example, on a pica typewriter font, the serifs on the T are as large as the feature at the lower right of the G, which distinguishes it from the C.

R. Marcus (MIT): Doesn't your system essentially consider each point in the matrix independently; that is, no consideration of correlation between different points?

O. N. Minot (USN Electronics Lab.): Could you give an example of a problem which leads to the sort of double peaks which you mention as requiring additional units in the adjusting equipment?

Mr. Mattson: The answer to the first question. This device considered individual combinations of bits but not single bits by themselves. It is the combination of certain key bits that enables the machine to recognize characters and other patterns.

The function which results in the double peak for the process is considered a function where, say, the 1111 point wanted to be mapped as a 1 and the 0000 point wanted to be mapped as a 0. It is impossible to have both of these points on one side of the plane and the others on the other side, so this function would yield two peaks.

E. B. Cohen (Auerbach): Can your approach learn to distinguish the handwriting of different individuals?

T. T. Rocchi (BTL): How do you explain that percent recognition increased, other things being similar, when the number of alphabets learned was increased?

Mr. Browning: In answer to the first question, it seems impossible that wth the present definition — which we had with 150 photocells — that we could recognize the handwriting of different individuals. After all, that number of photocells is approximately one-tenth of the number that a gnat has. If the number were greatly increased, we might well be able to distinguish the handwriting of different individuals.

In answer to the second question, I would explain the percentage recognition increase as follows. If the memory matrix has learned only one alphabet, any attempt to recognize an unknown character will be essentially pattern-matching with a complete position sensitivity. If the memory-matrix has learned a number of alphabets, the probability is increased that an unknown character will match a similar pattern previously learned in this position. The novel feature of our type of memory-matrix access is that the learning of a few subsequent patterns does not destroy its memory of previously learned patterns.