MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Project MAC

A.I. Memo 94 Memorandum MAC-M-299

To: All MAC Participants

From: Arnold K. Griffith

Subject: A New Machine-Learning Technique Applied to the Game of Checkers

Date: March 9,1966

1.1 This paper describes a recent refinement of the machine-learning process employed by Samuel(1) in connection with his development of a checker playing program. Samuel's checker player operates in much the same way a human player does; by looking ahead, and by making a qualitative judgement of the strength of the board positions it encounters. A machine learning process is applied to the development of an accurate procedure for making this strength evaluation of board positions. Before discussing my modifications to Samuel's learning process, I should like to describe briefly Samuel's strength evaluation procedure, and the associated learning process.

1.2 Samuel's playing program assigns a strength value to a board position on the basis of the values of a fixed set of 31 parameters. An example of such a parameter is the degree to which the move leading to the position in question contributes to control of the center of the board. The strength value for a board position is simply a weighted sum of the parameter values. Mathematically, the strength value is a linear function of the parameter values:

$$S(X_1,X_2,\ldots,X_{31})=\sum_{i=1}^{31} A_i X_i$$

where the A_i's are the weighting factors.

1.3 The purpose of Samuel's machine-learning process is to select these parameters, and to arrive at the weighting factors. This process is embodied in a learning program separate from the playing program, which analyzes transcribed games played by checker masters. More specifically, what is analyzed are sets of all possible positions immediately following a position occurring in the course of a transcribed game. Among these positions is the one resulting from the move actually made by the checker master. The learning program assumes this to be the strongest position of the set and designates it as such.

1.4 The remainder of the analysis is carried out on these sets of positions as follows. The value of each parameter is computed for all positions in the set, and the relation between a parameter's value for the strongest position and its values for other positions is noted. From this information, collected from many sets of positions, a correlation coeficient is computed which indicates the linear relation between the value of a particular parameter and the strength of corresponding board situations. For example, a parameter such as piece advancement might often have a high value for the strongest position in a set of positions, and a low value for for the weaker positions. This analysis would assign a high positive correlation coeficient to such a parameter. Of the many parameters tested, only those with a high coeficient were included among the 31 used by Samuel in his playing program. The correlation coeficient for a parameter was used as its weighting factor.

1.5 The modified learning process here described is analogous to the selection of weighting factors in the process described above. No judgement is made as to the utility of possible parameters; and exactly the set selected by Samuel are employed. The purpose of the learning process is to aid in the construction of a function which assigns strength values to board situations. Again, the process is based on an analysis of games played by checker masters.

1.6 The essential difference between the modified learning process and the original is that the strength function produced by the former is not restricted to being linear. The use of a linear strength function is equivalent to assuming that board strength varies linearly with the value of each parameter, and that the parameters are themselves not interrelated. Such assumptions are not entirely valid. Hence, it was felt that a more accurate strength evaluation function might be produced by a learning technique less restricted as to what sort of function it could produce. This flexibility is made possible by the use of tabulated functions as will now be described.

2.1 The new scoring function is defined in terms of eight auxiliary functions whose values are given in tables listing the value of each function for all possible sets of argument values. Such a function is practical in terms of space requirements if the number of arguments is small, and if each argument can take on only a small number of values. The eight functions of this type which we shall use have five arguments each, and the arguments take on only the values 0,1, or 2. For such a function, a table of only 243 entries is required. It should be noted that any function whatever of five three-valued arguments can be defined by such a table.

2.2 For this sort of tabulated function there exists a simple correspondence between five-tuples of argument values and locations in the table where a corresponding function value is to be placed. This is best explained by the following example. Consider, for such a function, the five-tuple of argument values 1,0,1,0,2. Regard this five-tuple as a base three integer, 10102. This number is 92 base ten, and F(1,0,1,0,2) is located in the 92nd location of the function's table.

2.3 The form of the new scoring function S' is:

$$S'(X_1, X_2, \ldots X_{31}) = \sum_{i=1}^{8} F_i'(Y_{i,1}', Y_{i,2}', \ldots, Y_{i,5}')$$

Where:

a. Each of the eight F_i''s is a tabulated function as described above,

b. Each $Y_{i,j}'$ is one of the X_i''s. Hence the five-tuple of argument values presented to a subfunction F_i' is a particular subset of the arguments presented to S'.

c. The argument values presented to the function S' are the values of the 1st to 31st parameters selected by Samuel for use by his playing program. However, they have been reduced to having the value 0 if negative, 1 if zero, and 2 if positive. The reduction of argument values to the range 0,1,or 2 is clearly necessitated by the nature of the component F 's, since it is subsets of values of these parameters that are used as the arguments presented to these functions. The possibility of constructing a successful board-strength function which uses only this limited amount of information about a parameter value was suggested by the nature of the parameters themselves. The parameters tend to be only qualitative in nature, so that little information about what a parameter is supposed to measure is gained from an exact numerical value.

2.4 Let us consider a simple example of the operation of this function. Assume that for each i, $Y_{i,1}', Y_{i,2}', \ldots, Y_{i,5}'$ are chosen to be $X_1, X_2, \ldots X_5$ respectively. Note that this is a rather trivial case. We wish to demonstrate how the value of the function S' for the parameter values -5,-2,1,-3,0,0,...,0 is computed. These values are first reduced to values 0,1,or 2: 0,0,2,0,1,1,...,1. For each i, the values presented to the component F_i''s, i.e. the values of $Y_{i,1}, Y_{i,2}, \ldots, Y_{i,5}$ are respectively 0,0,2,0,1. Thus:

$$S' = F_1(0,0,2,0,1) + F_2(0,0,2,0,1) + \ldots + F_8(0,0,2,0,1)$$

Since 00201 as a base three integer is 19 base ten, the value of $F_i(0,0,2,0,1)$ is located in the 19th entry in the table of values for the i-th function. Thus the value of S' for the given arguments is the sum of the 19th entries in each of the eight tables.

3.1 The object of the modified learning procedure is to determine appropriate tabulated function values from an analysis of transcribed games of checker masters. Again, sets of alternative positions, among which is designated a strongest, are recorded. For each position, the values of the 31 parameters are computed, and reduced to the values 0,1,or 2. From this set of values, eight five-tuples of values are chosen, each five-tuple being the appropriate set of arguments for a component function. These five-tuples are regarded as integers whose value is a serial location in a function-value table, according to the scheme described in section 2.2. Hence to each position in the set of positions corresponds eight locations, one in each table. At thhs stage of construction, each table location contains two components, a right and a left half, both initially zero. For each position in a set which is not a strongest position, a one is accumulated in the right half of each of the eight table entries to which that position corresponds. For the strongest position in the set, a one is accumulated in the left half of each of the corresponding table locations. This process is repeated for thousands of sets of positions.

3.2 At this point, the entry pairs are converted to the form in whhch they are to be used. This is accomplished by dividing the left half by the right half, and replacing the half-entries by the quotient. Let us note what such a quotient represents. Consider a particular entry in a particular table. To the table under consideration corresponds a set of five parameters, the values of whhch are used as the arguments of the function whose values are placed in this table. Call these parameters P1,P2,...,P5. To the entry in question corresponds a set of values for these parameters. Assume, for example that these values are 2,0,1,0,1. The reader may verify by referring to section 2.2 that we have in mind the 172nd table entry. The value of the table entry under consideration represents approximately the probability that a position with the values 2,0,1,0,1 for the parameters P1,P2,...,P5 is a strong position. This may be seen by observing that ones were accumulated in the table entry under consideration exactly when a position with the values 2,0,1,0,1 for parameters P1,P2,...,P5 was encountered in the reconstruction of book games. Whenever such a position was designated as the strongest in its set, a one was accumulated in the numerator of the fraction whose value was ultimately to replace the two half-entries. Similarly, for a non-strongest position, a one was accumulated in the denominator. Enough positions were examined that the numerators and denominators of these fractions were in general significant.

3.3 It would seem that to make full use of the type of function I have described, an optimal choice should be made for the parameter sets to be used as the arguments of the component functions. A rather elaborate technique was devised for this purpose; and the argument sets used in the function whose performance I shall describe were arrived at by this technique. However, insufficient data exists to make any evaluation of this technique, and I shall not describe it here.

4.1 The performance of this function was extensively tested on actual checker situations. Again, tabulated games of checker masters were used. As before, sets of alternative positions were recorded, with that of the checker master designated as strongest in the set. The function was applied to each position in a set, and the extent to which the scoring function agreed with the checker master's opinion was noted. The scoring function was considered to have made an accurate evaluation for a set of positions if the highest or next-highest score in the set was assigned to the checker master's choice. An assignment of the next-highest score to the designated strongest position was considered to be accurate, since often there are two best moves from a given position. Thus in many such cases the position given the highest score is as strong as the one designated as strongest. The function was considered to have made a blunder for a set of positions if the checker master's choice was given a score below the median of the scores assigned to the positions in the set.

4.2 The tables of function values were arrived at from an analysis of 12,000 sets of positions. The scoring function using these tables was tested on 1,000 positions which were not among those analyzed. The percentage of blunders was about 21, the percentage of accurate evaluations was about 50, 28% perfect, and 22% next perfect.

4.3 Samuel measured the accuracy of his polynomial S in terms of a single index. This index has the value +100 if the assignment of scores to the positions in a set is always such that the highest score is assigned to the strongest position. The index takes on a value around 0 if the scoring function is performing no better than it would have by making a random assignment of scores to the positions in a set. For comparison purposes, this index was also computed for the function herein described. The function S' had a performance index of 34 on the sample mentioned in section 4.2. For the same sample Samuel's S had a performance index of about 27.

4.4 The index mentioned in the last paragraph also provides an indication of the relation between the performance of the new scoring function and the number of position sets used in constructing it. The function was constructed from the first 2,000 position sets of the 12,000 described in 4.2. Then its performance index on the test sample was computed. This procedure was repeated using the first 4,000 positions, the first 6,000 positions, and finally with the entire set of 12,000 positions. As might be expected, the more position sets used to generate the function, the better the performance, up to a certain saturation point. This effect is shown by the lower line in Fig. 1.

4.5 This index was also computed for the performance of the function on a sample of position sets among those from which the function had been constructed. This sample consisted of the first 1,000 position sets of the 12,000 mentioned in 4.2. Again, the function was constructed from the first 2,000 position sets, the first 4,000, the first 6,000 and all 12,000. The function's performance index was computed in each case. The performance on this sample falls from an initially very high value to somewhat above the value ultimately attained for performance on the sample mentioned in 4.2. This effect is illustrated by the upper line in fig. 1.

5.1 The new function S' has shown itself to be more accurate than Samuel's linear polynomial S. An even more accurate scoring function might be possible if the set of 31 parameters is augmented by the addition of non linear parameters, or sets of interrelated parameters. A non linear parameter is one which, for example, takes on the value 2 or 0 for strong positions and the value 1 for weak positions. An example of the behavior of a related pair of parameters, A and B, is as follows. The values 2 and 2 of A and B respectively, and the values 0 and 0 of A and B indicate a strong board situation. Other combinations indicate a weak position. Notice that parameters of this type could easily have very low linear correlations with board strength. They would thus be rejected as useless by Samuel's learning process. However they would not necessarily be useless when employed in connection with the sort of non linear function produced by the modified procedure.

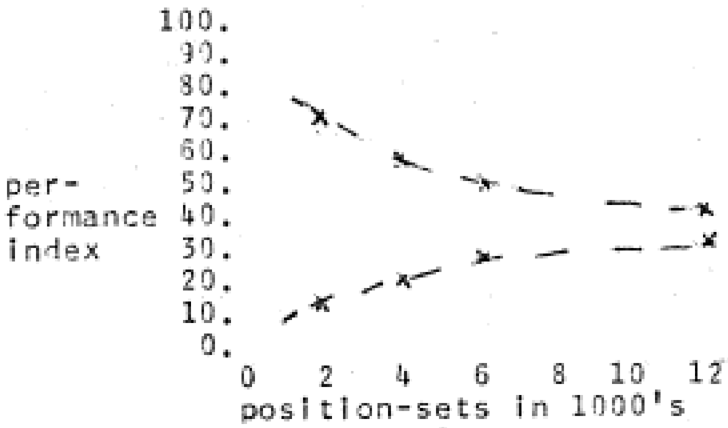

Figure 1.

(1) Samuel, A.L., Some studies in Machine Learning Using the Game of Checkers, I.B.M. Journal of Research and Development 3(3):210-229