REFERENCES

[1] J. G. Cleveland, R. H. Ramsey, and P. Walters, "Storm water pollution from urban land activity," in *Combined Sewer Overflow Abatement Technology*, Water Pollution Control Research Series 11024-06/70, Federal Water Quality Administration, U.S. Dep. Interior, 1970.

[2] American Public Works Association, "Assessment of combined sewer problems," in *Combined Sewer Overflow Abatement Technology*, Water Pollution Control Research Series 11024-06/70, Federal Water Quality Administration, U.S. Dep. Interior, 1970.

[3] Cornell, Howland, Hayes, and Merrifield, "Rotary vibratory fine screening of combined sewer overflows," in *Combined Sewer Overflow Abatement Technology*, Water Pollution Control Research Series 11024-06/70, Federal Water Quality Administration, U.S. Dep. Interior, 1970.

[4] Rex Chain Belt, Inc., "The use of screening/dissolved—air flotation for treating combined sewer overflows," in *Combined Sewer Overflow Abatement Technology*, Water Pollution Control Research Series 11024-06/70, Federal Water Quality Adminstration, U.S. Dep. Interior, 1970.

[5] "Dispatching system for control of combined sewer losses," Water Pollution Control Research Series 11020 FAQ 03/71, Water Quality Office, Environmental Protection Agency, 1971.

[6] W. Bell and C. B. Winn, "Minimization of pollution from combined storm-sewer systems," presented at the Int. Syst. Symp., Purdue Univ., Lafayette, Ind., Oct. 1972.

[7] W. Bell, G. Johnson, and C. B. Winn, "Simulation and control of flow in combined sewers," presented at the 6th Annu. Simulation Symp., Tampa, Fla., June 1973.

[8] B. D. O. Anderson and J. B. Moore, *Linear Optimal Control*. Englewood Cliffs, N.J.: Prentice-Hall, 1971.

[9] K. Hitz and B. D. O. Anderson, "An iterative method of computing the limiting solution of the matrix Riccati equations," *Proc. IEEE*, to be published.

Punish/Reward: Learning with a Critic in Adaptive Threshold Systems

BERNARD WIDROW, NARENDRA K. GUPTA, AND SIDHARTHA MAITRA

Abstract—**An adaptive threshold element is able to "learn" a strategy of play for the game blackjack (twenty-one) with a performance close to that of the Thorp optimal strategy although the adaptive system has no prior knowledge of the game and of the objective of play. After each winning game the decisions of the adaptive system are "rewarded." After each losing game the decisions are "punished." Reward is accomplished by adapting while accepting the actual decision as the desired response. Punishment is accomplished by adapting while taking the desired response to be the opposite of that of the actual decision. This learning scheme is unlike "learning with a teacher" and unlike "unsupervised learning." It involves "bootstrap adaptation" or "learning with a critic." The critic rewards decisions which are members of successful chains of decisions and punishes other decisions. A general analytical model for learning with a critic is formulated and analyzed. The model represents bootstrap learning per se. Although the hypotheses on which the model is based do not perfectly fit blackjack learning, it is applied heuristically to predict adaptation rates with good experimental success. New applications are being explored for bootstrap learning in adaptive controls and multilayered adaptive systems.**

INTRODUCTION

ADAPTIVE LINEAR threshold logic elements have been studied closely over the past decade or so. Examples of such work are contained in [1]–[9]. Analyses of the dynamic and steady-state behavior of such units can be found in [10]–[14]. Applications of such elements in both supervised and unsupervised training situations abound in the literature. Training algorithms for "learning with a teacher" [1]–[16] and also for "unsupervised learning" [17]–[26] exist and have been analyzed. A mixture of the two has also been proposed [27].

The purpose of this paper is to describe a different type of learning process involving adaptive linear threshold logic elements. By means of this process, called *learning with a critic* or *selective bootstrap adaptation*, an adaptive logic element learns what is required of it solely through the receipt of favorable or unfavorable reactions resulting from the application of an overall performance criterion to the outcome of a series of decisions made by the element.

Until recently, adaptive threshold elements have been used primarily as trainable pattern-classifying systems. When these elements are being trained, a desired response (representing the pattern class) is specified for each input pattern vector (input signal vector). This kind of process is called "learning with a teacher." More recent work in the field has developed adaptation algorithms which permit "unsupervised learning," sometimes called "learning without a teacher," or "decision-directed learning" [28], [29]. The adaptive process reported here cannot be considered a learning-with-a-teacher process; neither can it be described as an unsupervised-learning process. We are concerned here with an adaptive process wherein the desired response cannot be supplied for each input pattern, but the outcome of a series of decisions can be judged.

Applications for such adaptive procedures arise in certain sequential-decision processes, in the automatic synthesis of optimal strategies for gaming and control, and in convergent adaptation schemes for multilayered and more generally connected networks of adaptive threshold elements.

Manuscript received September 28, 1972.
The authors are with the Department of Electrical Engineering, Stanford University, Stanford, Calif. 94305.

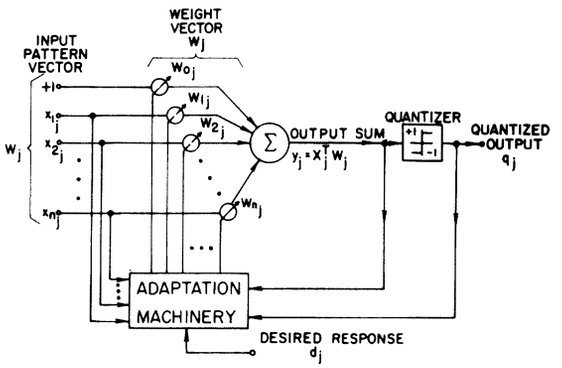


Fig. 1. Automatically adapted threshold logic element (Adaline).

Conventional Adaptation Processes for Threshold Elements

In order to understand the process of learning with a critic, we begin by briefly describing the process of learning with a teacher. Fig. 1 shows a functional diagram and schematic symbol for an adaptive linear threshold logic element (sometimes called "Adaline") [1]. The diagram indicates the terminology used and the input–output relationships. The zeroth input-signal component is always $+1$. Thus the zeroth weight w_0 controls the threshold partitioning level. Before adaptation, an error ε (defined as the difference between the output y and the desired response d) exists for each input pattern. The jth input pattern would have an error of

$$\varepsilon_j \triangleq d_j - y_j = d_j - X_j^T W_j \tag{1}$$

where X_j is the jth input pattern vector and W_j is the jth weight vector. It is assumed here that the weight vector is adapted with each new input vector X_j.

If the inputs X_j and d_j are statistically stationary, then the mean-square error (mse) is a quadratic function of the weights and there exists an optimal Wiener weight vector which minimizes it [30]–[32]. Learning or updating of the weights can be done by a gradient-descent technique. The least-mean-square (LMS) algorithm developed by Widrow and Hoff [1], [4], [10], [32] uses the error as an estimate of the gradient. This leads to the weight iteration rule

$$W_{j+1} = W_j + \left(\frac{\alpha}{n+1}\right) \varepsilon_j X_j \tag{2}$$

where $n + 1$ is the total number of weights and α is a coefficient determining the fraction of the error ε_j corrected with each adaptation. The parameter α controls the stability of the adaptive process and the rate of convergence. The adaptive process has been shown [33] to be stable (convergent) if α is within the range $2 > \alpha > 0$. Choosing α in this range ensures that $|\varepsilon_j|$ is reduced by the jth adaptation.

The "learning curve" plot of mse versus the number of adaptation cycles is a noisy exponential whose time constant can be shown to be [31], [32]

$$\tau_{\text{mse}} = \frac{(n+1)}{2\alpha} \text{ adaptations.} \tag{3}$$

Formula (3) is exact when all eigenvalues of the input correlation matrix $R \triangleq E[X_j X_j^T]$ are identical. A general formula for a time constant (there are as many time constants as there are distinct eigenvalues) is derived in [32]. Even when the eigenvalues differ substantially and the learning curve is not simply a single exponential plus noise but is a sum of exponentials plus noise, experience has shown that in most cases the learning curve can be well approximated by a single exponential having the time constant given by (3).

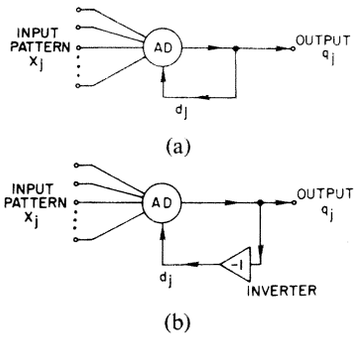


Fig. 2. Bootstrap adaptations. (a) Positive. (b) Negative.

Selective Bootstrap Adaptation: Learning by "Reward and Punishment"

Learning with a teacher is a straightforward matter, as described in the preceding section. The question is, what should be done when an adaptive element is connected to an environment that provides a stream of input patterns, but in which the desired response for each input pattern is not known?

One possibility is to connect the quantized output q_j of the threshold element to the desired-response input as shown in Fig. 2(a). Under this plan, when a new pattern is applied the adaptive element assumes that its own binary output decision is the correct desired output. It adapts its weights accordingly, applying the LMS algorithm or some other adaptation algorithm, always moving the output y_j closer to its own signum ($+1$ or -1, as the case may be).[1] The tendency here is to maintain the binary responses that already exist (i.e., q responses established by the initial weight settings), although some analog responses (y values) close to the zero threshold may reverse during this process. Essentially, the adaptive element has the attitude "don't bother me with the facts, my mind is made up." Let this procedure be called *positive bootstrap adaptation* or *learning by reward.* Positive bootstrap adaptation is the basis of decision-directed learning and of learning without a teacher in adaptive threshold element systems.

An alternative means of supplying the desired response from the output signal is shown in Fig. 2(b). Here the output signal goes through an inverter which forms its complement. The inverted output is then taken as the desired output. Let this form of adaptation be called *negative bootstrap adaptation* or *learning by punishment.* Now, whenever a new input pattern is applied, adaptation takes place

[1] $\text{sgn } y_j \triangleq \begin{cases} +1, & y_j \geq 0 \\ -1, & y_j < 0. \end{cases}$

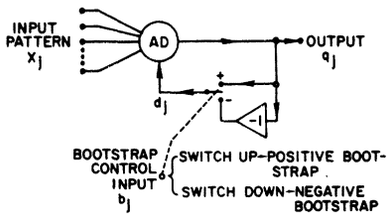


Fig. 3. Selective bootstrap adaptation.

to change the output y_j closer to the complement of its own signum. A sustained application of negative bootstrap adaptation will eventually cause all weight values to approach zero, which will neutralize the effects of initial weight conditions. A threshold element adapting in this fashion would have the attitude "everything I do is wrong."

A combination of positive and negative bootstrap adaptation is illustrated in Fig. 3. In this configuration, two kinds of input information are again required to produce an adaptation: the input pattern X_j and a bootstrap control signal b_j. When b_j is positive (switch up in Fig. 3), positive bootstrapping (rewarding) takes place; when b_j is negative (switch down), negative bootstrapping (punishing) is performed. Let this process be called *selective bootstrap adaptation* or *learning by reward and punishment.*

The kind of information supplied as b_j in Fig. 3 will be quite different in practice from that supplied as d_j in Fig. 1. If an external evaluator (the "critic") indicates that the present decision is a member of an aggregate of decisions whose consequences have produced relatively successful results, then b_j is made positive; otherwise b_j is made negative. Thus selective bootstrap adaptation (henceforth simply referred to as bootstrap adaptation) involves learning with a *critic*, as opposed to learning with a *teacher.* The critic is qualitative. The teacher is specific.

Application of Bootstrap Adaptation to Simulated Blackjack Play

In order to make the idea of selective bootstrap adaptation clearer and to stimulate ideas for its application, an example will be presented relating to the playing of the game blackjack or twenty-one [34], [35]. It has been found that using selective bootstrap adaptation, a single threshold element is able to learn to play this game very well without knowing the rules or the objectives of the game. All that is needed is the knowledge, at the end of each game, of whether the game was won or lost.

Blackjack is a card game in which the player, after seeing one of the dealer's cards, draws a series of cards. At any stage, the player has the choice of drawing or not drawing ("hit" or "stick"). If the sum of values of cards in his hand crosses 21, he "busts" (loses). Otherwise, after the player sticks, the dealer draws a series of cards, playing a fixed "house" strategy. He has no choice. If the dealer crosses 21, he busts. Otherwise, whoever comes out nearer to 21 wins. For an experimental study, the mechanical dealer was simulated on a computer which "dealt" using a random-number generator. The computer did all score keeping and periodically typed out the performance of the "player," an

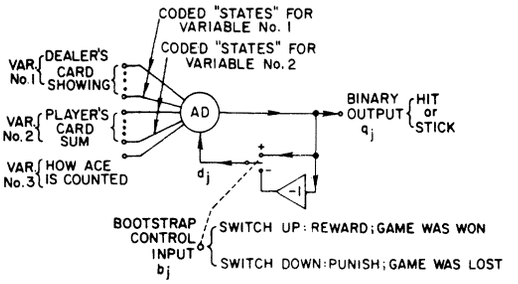


Fig. 4. Bootstrap adaptation applied to game of blackjack.

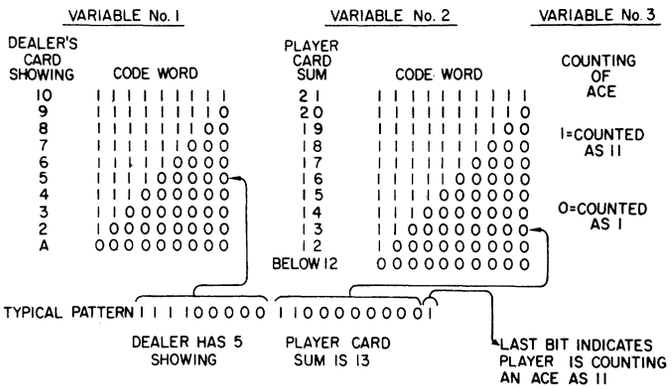


Fig. 5. Blackjack game states encoded as patterns for input to threshold logic element "player."

adaptive threshold element, which it also simulated. As a matter of incidental interest, we should point out that the game of blackjack was simplified by removing all special features such as "splitting pairs," "doubling down," and "insurance." "Blackjacks" were counted. The "card deck" was reshuffled after each draw.

Fig. 4 shows how the simulated adaptive threshold element was able to perform the function of player in the blackjack game. The decisions made by the player are based on the dealer's card showing and on the sum of the face values of the cards in the player's hand. These data, together with an indication of how the ace is counted, constituted the inputs to the threshold element. These inputs were encoded as shown in Fig. 5. Notice that the different input states were encoded with binary words which are algebraically linearly independent.

The adaptive player begins making decisions with a given set of initial weights. During a given game, several hit or stick decisions are made with the weights fixed. For each state of the game, i.e., for each input vector, the decision made by the player is recorded by the computer. At the end of the game, the computer notes whether the player has won or lost. Then adaptation is effected by replaying the game. If the player has won, either by luck or by good strategy, all of the decisions that were made in the game are rewarded during the replay by adapting keeping the "b_j switch" up. If the player has lost, then these decisions are punished by adapting with the "b_j switch" down (see Fig. 4). The resulting weights are then used in the next game, after which the cycle of bootstrap adaptation is repeated. The experience accumulated over many games is stored in the weights. The weights in turn completely govern the strategy of play.

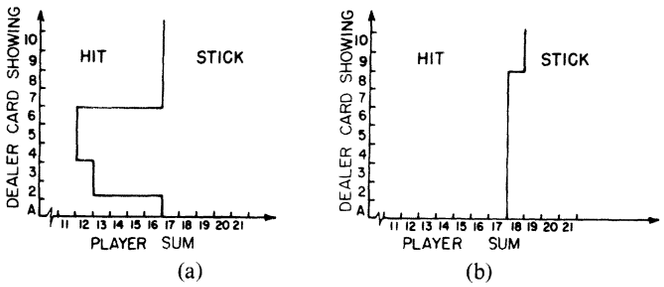


Fig. 6. Optimal blackjack strategy (Thorp's "basic strategy"). (a) Player counts all aces as 1. (b) Player counts ace as 11.

With suitable state-variable encoding, a single fixed-weight threshold element can realize the optimal "basic blackjack strategy" of Thorp. This fact was first noted by Smith[2] in 1963. When the value of the dealer's face card is encoded in a linearly independent binary code [36], and when the sum of the cards dealt to the player is also encoded in this way, the binary patterns representing the states of the game, together with the associated binary decisions (hit or stick) corresponding to the Thorp optimal strategy, constitute a linearly separable set [4], [37], yet they represent a nonlinear discriminant function. Thus, through the encoding procedure, a nonlinear function is made perfectly realizable by a single linear threshold logic element. The strategy needed to play the game of blackjack is related to that required for the "bang-bang" (contactor) control of a variety of dynamic systems [38]. In both cases, binary decisions must be made based on the values of several analog or multilevel state variables.

The optimal strategy (minimum probability of loss) for the simplified game of Thorp is presented in Fig. 6. It should be noted that when playing this game with the optimal strategy, the player will lose at a certain small average rate. The adaptive player must learn to minimize its losses.

The learning process just described has several unique features. Learning was not directed by a teacher along each step of the way. The effects of individual decisions could not be independently evaluated. They all had a statistical effect on the final outcome based on a composite of the quality of the responses to a series of patterns (states) of the game which were for all practical purposes selected at random. In addition, because of the element of chance drawing of "cards," games were sometimes won when playing with poor strategies and games were sometimes lost when playing with excellent strategies. The net result is that adaptation on a given pattern vector did not always proceed in the proper direction. Consequently, bootstrap learning takes place at a slower rate than conventional learning with a teacher.

[2] F. W. Smith, while a graduate student in the Department of Electrical Engineering at Stanford University, proposed in 1963 that the game of blackjack be used in a test application of bootstrap learning principles. He was inspired by an article in *Time Magazine*, issue of January 25, 1963, relating to Thorp's work on optimal blackjack play. The *Time* article showed the optimal switching line in "state space," very similar to that shown in Fig. 6. Smith was doing a Ph.D. dissertation on the realization of nonlinear separating boundaries with linear threshold elements whose inputs were suitably encoded.

Analytical Model of Bootstrap Adaptation

The purpose of the following analysis is to predict the rate of learning of the bootstrap process using a model based on a set of hypotheses. The hypotheses hold in a general way for a wide variety of bootstrap learning applications, including the game of blackjack, but do not exactly correspond to the latter application in all details. The purpose of the model is to represent bootstrapping per se, with the particular objective of studying the theoretically achievable learning rate and the method by which the best rate can be realized.

In order to arrive at a definition of terms, we begin by considering a coin-toss situation in which the coin is unsymmetrically weighted or biased. After 1000 or so tosses, we notice that 60 percent of the tosses come out "heads" and 40 percent come out "tails." Now imagine building a system to predict the outcomes of tosses with this coin. The optimal system (one having the minimum statistical expectation of error) would always make a fixed prediction: heads. Although all the decisions made by the optimal coin-toss predictor are by definition optimal (i.e., best over the long range), some of these decisions will turn out to be "right," and some will be "wrong."

Consider next another coin-toss predictor whose performance is less than optimal. Some of its decisions will agree with those of the optimal predictor, and the rest will disagree. Thus some of its decisions will be optimal, while the rest will be antioptimal; some will be right, the rest will be wrong. A given decision could be optimal or antioptimal, and right or wrong. The notions of right/wrong, optimal/antioptimal are useful in the mathematical study of bootstrap adaptation.

We next develop an idealized mathematical model for bootstrap learning. It is based on a set of hypotheses which were motivated by practical experience. A block diagram of the model is shown in Fig. 7. It contains an adaptive system that learns by bootstrap, a "perfect-knowledge" system whose decisions are always right, an optimal system whose decisions are always optimal, and a critic that evaluates the decisions of the adaptive system relative to the perfect-knowledge system.

In the model of Fig. 7, input-signal vectors are assumed to be applied to the adaptive system, to the perfect-knowledge system, and to the optimal system. The perfect-knowledge system gets additional "super-knowledge" inputs, unavailable to the other two systems, in order that it may always be able to make right decisions. Super-knowledge is, for example, perfect knowledge of which cards will be drawn from a deck, of which way coin tosses will go, etc.

In physical situations, the adaptive system will exist, will make decisions, and will learn from them. The perfect-knowledge system will not exist directly; otherwise there would be no need for the learning system. The outputs of the perfect-knowledge system are generally available in an

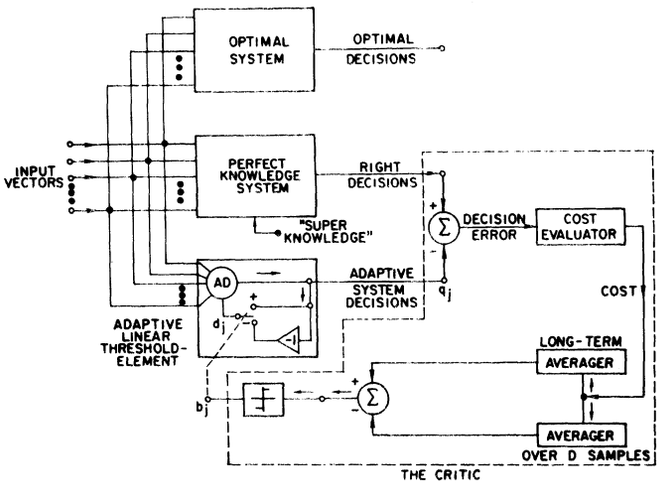
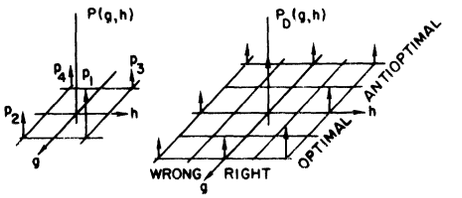


Fig. 7. Bootstrap adaptation model.



(a) (b)

Fig. 8. Probability density function of number of right and number of optimal decisions. (a) One decision. (b) D decisions.

ex post facto sense—after the fact, after a set of decisions have been made. The performance of the adaptive system relative to the perfect-knowledge system is appraised by the critic for adaptive purposes after the chain of decisions has been made.

In the model, the function of the critic is hypothesized as follows. If the number of right decisions made by the adaptive system in the given chain is greater than that averaged over many recent chains, then these decisions are rewarded. Otherwise they are punished.

The optimal system shown in the block diagram of Fig. 7 has no direct effect on the adaptive process and is generally unavailable to it. Its performance upper bounds that of the adaptive system. In the next section it is argued from a heuristic point of view that the adapting weight vector approaches (albeit more slowly) the same weight that would have resulted if learning with the optimal system as a teacher were possible.

Analysis of Selective Bootstrap Adaptation

In a sequence of decisions made by the adaptive threshold element of Fig. 7, it is likely that some will be optimal and right (O,R), some will be optimal and wrong (O,W), some will be antioptimal and right (A,R), and some will be antioptimal and wrong (A,W). These four are the only possibilities. Arrayed in a group of length D, these decisions might occur as follows:

$$(O,R),(A,W),(A,W),(O,R),(O,W),(A,R),(O,R),\cdots,(A,W).$$

Let the probability of (O,R) be p_1, the probability of (O,W) be p_2, the probability of (A,R) be p_3, and the probability of (A,W) be p_4.

A sketch of the joint probability density for a single decision as a function of the number of right and the number of optimal decisions is shown in Fig. 8(a). This function is

$$P(g,h) = p_1\delta(1-g,1-h) + p_2\delta(1-g,1+h) + p_3\delta(1+h,1-g) + p_4\delta(1+h,1+g) \quad (4)$$

where h is the axis of right/wrong decisions and g is the axis of optimal/antioptimal decisions. Note that δ is a two-dimensional Dirac function defined to have a unit volume.

The joint probability density $P_D(g,h)$ (Fig. 8(b)) is a function of the number of right and the number of optimal decisions in a chain of D decisions. The value of the g parameter is the sum of the number of optimal decisions minus the number of antioptimal decisions; the value of the h parameter is the sum of the number of right decisions minus the number of wrong decisions. Assume that the decisions in the sequence are statistically independent. It then follows that the joint probability-density function for a chain of D decisions is a D-fold convolution of the density function for a single decision:

$$P_D(g,h) = P(g,h) * P(g,h) * \cdots * P(g,h). \quad (5)$$

In order to derive an expression for the learning rate of the bootstrap process, it is necessary to obtain the probability p_+ of an individual adaptation being in the optimal direction. The probability of adapting in the antioptimal direction is $p_- = (1 - p_+)$. If the bootstrap adaptation process is to be useful it is important that a critical parameter $(p_+ - p_-)$ be greater than zero. To calculate $(p_+ - p_-)$, a certain type of moment will have to be evaluated for the discrete joint probability density $P_D(g,h)$. In order to simplify this moment calculation, it will be assumed that D is sufficiently large so that $P_D(g,h)$ could be replaced for purposes of moment calculation by a two-dimensional Gaussian density function. The justification for this is the central limit theorem. The parameters of a Gaussian approximation function $\hat{P}_D(g,h)$ will have the same mean values, the same variances, and the same correlation coefficient as $P_D(g,h)$.

The first step is to find the means, the variances, and the covariance of the density function $P(g,h)$ of the single decision (Fig. 8(a)). The means are

$$\bar{g} = p_1 + p_2 - p_3 - p_4 \quad (6)$$

$$\bar{h} = p_1 + p_3 - p_2 - p_4. \quad (7)$$

The variance along the g axis is

$$\sigma_g^2 \triangleq \overline{g^2} - (\bar{g})^2 = p_1 + p_2 + p_3 + p_4 - (\bar{g})^2 = 4(p_1 + p_2)(p_3 + p_4). \quad (8)$$

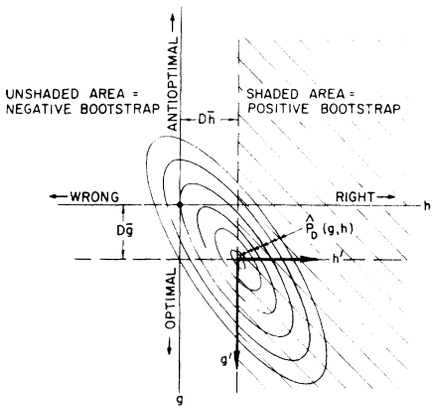


Fig. 9. Regions of positive and negative bootstrapping.

The variance along the h axis is

$$\sigma_h^2 \triangleq 4(p_1 + p_3)(p_2 + p_4). \tag{9}$$

The covariance is

$$\begin{aligned}\sigma_{gh}^2 &\triangleq \overline{gh} - (\bar{g})(\bar{h}) \\ &= p_1 + p_4 - p_2 - p_3 - (\bar{g})(\bar{h}) \\ &= 4p_1p_4 - 4p_2p_3.\end{aligned} \tag{10}$$

The correlation coefficient is

$$\begin{aligned}\rho &\triangleq \frac{\sigma_{gh}^2}{(\sigma_g^2 \times \sigma_h^2)^{1/2}} \\ &= \frac{p_1p_4 - p_2p_3}{\sqrt{(p_1 + p_2)(p_3 + p_4)(p_1 + p_3)(p_2 + p_4)}}.\end{aligned} \tag{11}$$

These parameters can now be easily calculated for the probability density $P_D(g,h)$. The means of this density function are

$$D\bar{g} \quad \text{and} \quad D\bar{h}. \tag{12}$$

The variances are

$$D\sigma_g^2 \quad \text{and} \quad D\sigma_h^2. \tag{13}$$

The correlation coefficient is the same as in (11).

The Gaussian approximation function $\hat{P}_D(g,h)$ will have parameters as determined by (11)–(13) and can be written

$$\begin{aligned}\hat{P}_D(g,h) = {} & \frac{1}{2\pi D\sigma_g\sigma_h\sqrt{1-\rho^2}} \exp\frac{-1}{2(1-\rho^2)} \\ & \cdot \left[\frac{(g - \bar{g}D)^2}{D\sigma_g^2} - \frac{2\rho(g - \bar{g}D)(h - \bar{h}D)}{D\sigma_g\sigma_h} \right. \\ & \left. + \frac{(h - \bar{h}D)^2}{D\sigma_h^2}\right].\end{aligned} \tag{14}$$

A contour map of $\hat{P}_D(g,h)$ is shown in Fig. 9.

According to the previously stated rules of adaptation, positive bootstrapping will be effected when measured performance is better than average, i.e., when the number of right decisions in the chain of D decisions exceeds the long-term average number of right decisions. It is assumed that, on the average, each decision in a chain of D decisions has equal expected effect upon measured performance.

Events where positive bootstrap adaptation takes place ($h > D\bar{h}$) are therefore indicated by the shaded area in Fig. 9. The unshaded area represents all other events, where negative bootstrap adaptation takes place ($h < D\bar{h}$).

Consider all chains of events where performance is better than average. Let the probability of such chains be $\mathscr{P}(h > D\bar{h})$. Then the probability of chains with below-average performance is $\mathscr{P}(h < D\bar{h}) = 1 - \mathscr{P}(h > D\bar{h})$. Since the joint Gaussian density $\hat{P}_D(g,h)$ is symmetrical,

$$\begin{aligned}\mathscr{P}(h > D\bar{h}) &= \int_{-\infty}^{\infty}\int_{D\bar{h}}^{\infty} \hat{P}_D(g,h)\,dg\,dh \\ &= \mathscr{P}(h < D\bar{h}) = \tfrac{1}{2}.\end{aligned} \tag{15}$$

Consider only chains with above-average right/wrong performance. Among these chains, all of which will (by the rules) experience positive bootstrap adaptation, the expected number of optimal decisions minus the expected number of antioptimal decisions is given by

$$E[g \mid h > D\bar{h}] = \frac{1}{\mathscr{P}(h > D\bar{h})}\int_{-\infty}^{\infty} g\,dg \int_{D\bar{h}}^{\infty} \hat{P}_D(g,h)\,dh. \tag{16}$$

For chains with below-average right/wrong performance, all of which will (by the rules) experience negative bootstrapping when adapted, the expected number of antioptimal decisions minus the expected number of optimal decisions is

$$\begin{aligned}E[-g \mid h < D\bar{h}] = {} & \frac{1}{\mathscr{P}(h < D\bar{h})} \\ & \cdot \int_{-\infty}^{\infty} -g\,dg \int_{-\infty}^{D\bar{h}} \hat{P}_D(g,h)\,dh.\end{aligned} \tag{17}$$

With positive bootstrapping ($h > D\bar{h}$), adaptation in the optimal direction takes place when the threshold-element decisions are optimal; the expected number of optimal adaptations minus the expected number of antioptimal adaptations is given by (16). With negative bootstrapping ($h < D\bar{h}$), adaptation in the optimal direction takes place when the threshold-element decisions are antioptimal; the expected number of optimal adaptations minus the expected number of antioptimal adaptations is accordingly given by (17). The average (over all adaptations) number of optimal adaptations minus the average number of antioptimal adaptations is therefore

$$\begin{aligned}&(p_+ - p_-)D \\ &= E[g \mid h > D\bar{h}]\mathscr{P}(h > D\bar{h}) \\ &\quad + E[-g \mid h < D\bar{h}]\mathscr{P}(h < D\bar{h}) \\ &= \int_{-\infty}^{\infty} g\,dg \int_{D\bar{h}}^{\infty} \hat{P}_D(g,h)\,dh - \int_{-\infty}^{\infty} g\,dg \int_{-\infty}^{D\bar{h}} \hat{P}_D(g,h)\,dh \\ &= \int_{-\infty}^{\infty} g\,dg \left\{\int_{D\bar{h}}^{\infty} \hat{P}_D(g,h)\,dh - \int_{-\infty}^{D\bar{h}} \hat{P}_D(g,h)\,dh\right\}.\end{aligned} \tag{18}$$

Changing variables $h' = h - D\bar{h}$ and $g' = g - D\bar{g}$, we have

$$(p_+ - p_-)D = \int_{-\infty}^{\infty} (g' + D\bar{g}) \left[\int_0^{\infty} \hat{P}_D(g',h')\, dh' - \int_{-\infty}^{0} \hat{P}_D(g',h')\, dh' \right] dg' \quad (19)$$

where

$$\hat{P}_D(g',h') = \frac{1}{2\pi D\sigma_g\sigma_h\sqrt{1-\rho^2}} \exp \frac{-1}{2(1-\rho^2)} \cdot \left(\frac{g'^2}{D\sigma_g{}^2} - \frac{2\rho g'h'}{D\sigma_g\sigma_h} + \frac{h'^2}{D\sigma_h{}^2} \right). \quad (20)$$

Equation (19) can be written

$$(p_+ - p_-) = \frac{1}{D} \int_0^{\infty} \int_{-\infty}^{\infty} \hat{P}_D(g',h')(g' + D\bar{g})\, dg'\, dh' - \frac{1}{D} \int_{-\infty}^{0} \int_{-\infty}^{\infty} \hat{P}_D(g',h')(g' + D\bar{g})\, dg'\, dh'. \quad (21)$$

Now

$$\int_{-\infty}^{\infty} \hat{P}_D(g',h')(g' + D\bar{g})\, dg' = \frac{\exp\left[-(h'^2/2D\sigma_h{}^2)\right]}{2\pi D\sigma_g\sigma_h\sqrt{1-\rho^2}} \cdot \int_{-\infty}^{\infty} (g' + D\bar{g}) \exp\left(\frac{-(g' - (\rho\sigma_g h')/\sigma_h)^2}{2(1-\rho^2)D\sigma_g{}^2} \right) dg' = \frac{\exp\left[-(h'^2/2D\sigma_h{}^2)\right]}{\sqrt{2\pi D}\,\sigma_h} \left[\frac{\sigma_g\rho h'}{\sigma_h} + D\bar{g} \right]. \quad (22)$$

Using this in (21), we have

$$(p_+ - p_-) = \left[\frac{\rho\sigma_g}{\sqrt{2\pi D}} + \frac{D\bar{g}}{2} \right] - \left[-\frac{\rho\sigma_g}{\sqrt{2\pi D}} + \frac{D\bar{g}}{2} \right] = \frac{2\rho\sigma_g}{\sqrt{2\pi D}}. \quad (23)$$

The expressions for σ_g and ρ, (8) and (11), may be substituted in (23) to give

$$(p_+ - p_-) = \frac{4(p_1p_4 - p_2p_3)}{\sqrt{2\pi D(p_1 + p_3)(p_2 + p_4)}}. \quad (24)$$

The next step is to find the probabilities p_1,p_2,p_3,p_4 for the individual adaptive-system decision. These probabilities can be related to the "physics" of the process by using the following expressions:

$$p_1 = P(\mathrm{O,R}) = P(\mathrm{R} \mid \mathrm{O})P(\mathrm{O}) \quad (25)$$

$$p_2 = P(\mathrm{O,W}) = P(\mathrm{W} \mid \mathrm{O})P(\mathrm{O}) \quad (26)$$

$$p_3 = P(\mathrm{A,R}) = P(\mathrm{R} \mid \mathrm{A})P(\mathrm{A}) \quad (27)$$

$$p_4 = P(\mathrm{A,W}) = P(\mathrm{W} \mid \mathrm{A})P(\mathrm{A}). \quad (28)$$

Let the probability of error of the optimal system (Fig. 7) be designated $P_{\min}$. This limiting performance can only be reached by the adaptive system when the optimal system is a linear threshold function.

The decisions of the adaptive threshold system will in general not always agree with the optimal decisions, i.e., those that would be made by the optimal system. It will be assumed, however, that when there is agreement, the probability that these optimal decisions are wrong is the same as that of any optimal decision. Accordingly, the probability of an optimal decision made by the adaptive threshold element being wrong is

$$P(\mathrm{W} \mid \mathrm{O}) = P_{\min}. \quad (29)$$

The probability of an optimal decision made by the adaptive element being right is therefore

$$P(\mathrm{R} \mid \mathrm{O}) = (1 - P_{\min}). \quad (30)$$

When the adaptive system disagrees with the optimal system, its decisions are antioptimal. Assume that the probability that these antioptimal decisions are right is the same as that of any antioptimal decision being right. Completely antioptimal decisions would result from the inversion or complementation of the output signals of the optimal system. Accordingly, the probability of an adaptive-system decision being right, given that the decision is antioptimal, is

$$P(\mathrm{R} \mid \mathrm{A}) = P(\mathrm{W} \mid \mathrm{O}) = P_{\min}. \quad (31)$$

Also,

$$P(\mathrm{W} \mid \mathrm{A}) = P(\mathrm{R} \mid \mathrm{O}) = (1 - P_{\min}). \quad (32)$$

All that remains to be found before p_1,p_2,p_3,p_4 can be determined is $P(\mathrm{O})$ and $P(\mathrm{A})$. At any stage of adaptation, let the error probability of the adaptive threshold system be defined as P_{adapt}. A normalized measure of the excess error probability, similar in concept to "misadjustment" [32] for adaptive linear systems, is the ratio of the excess error probability to the minimum error probability obtainable by the optimal system:

$$\begin{pmatrix} \text{excess} \\ \text{error} \\ \text{probability} \\ \text{normalized} \end{pmatrix} \equiv \psi = \frac{P_{\mathrm{adapt}} - P_{\min}}{P_{\min}}. \quad (33)$$

This can also be written

$$P_{\mathrm{adapt}} = P_{\min}(1 + \psi). \quad (34)$$

The error probability P_{adapt}, i.e., the probability that the adaptive system is wrong, can also be written

$$P_{\mathrm{adapt}} = P(\mathrm{W} \mid \mathrm{O})P(\mathrm{O}) + P(\mathrm{W} \mid \mathrm{A})P(\mathrm{A}) = P_{\min}[1 - P(\mathrm{A})] + (1 - P_{\min})P(\mathrm{A}) = (1 - 2P_{\min})P(\mathrm{A}) + P_{\min}. \quad (35)$$

Using (33) and (35),

$$P(\mathrm{A}) = 1 - P(\mathrm{O}) = \frac{P_{\min}\psi}{(1 - 2P_{\min})}. \quad (36)$$

The probabilities p_1,p_2,p_3,p_4 may now be found by substituting (29)-(32) and (36) into (25)-(28). The quantity $(p_+ - p_-)$ may then be found by substituting the expressions for p_1,p_2,p_3,p_4 into (24). The result is

$$(p_+ - p_-) = \frac{4}{\sqrt{2\pi D}} \frac{\psi\sqrt{P_{\min}}\,[1 - (2 + \psi)P_{\min}]}{(1 - 2P_{\min})\sqrt{(1 + \psi)[1 - (1 + \psi)P_{\min}]}}. \quad (37)$$

For practical cases having small $P_{\min}$ and ψ, (37) simplifies to

$$(p_+ - p_-) \cong \frac{4\psi\sqrt{P_{\min}}}{\sqrt{2\pi D}}. \quad (38)$$

Applications of Bootstrap Learning Model

In the previous sections of this paper, a mathematical model of the bootstrap punish-reward learning process has been proposed and analyzed. The key result, the derivation of $(p_+ - p_-)$, is given by (37) and (38). It is expected that this derivation will be very useful in understanding the behavior of bootstrap learning, although the set of hypotheses on which the analytical model is based may not always precisely agree with the physical situation in a given application.

Effects of $(p_+ - p_-)$ upon Rate of Adaptation

With reference to the adaptive model illustrated in Fig. 7, we conjecture that the adaptive threshold system will self-adapt toward forming a best least-squares fit to the optimal system as long as $(p_+ - p_-) > 0$. Reasoning heuristically, consider a situation wherein $(p_+ - p_-) = 0.2$. On the average, in making 10 adaptations, 6 will be in the optimal direction and 4 will be in the antioptimal direction. The net result is a preponderance of 2 adaptations out of 10 in the optimal direction. The rate of learning in this case would be 0.2 as fast as when learning directly with a teacher. The factor $1/(p_+ - p_-)$ is the ratio of the time constant of bootstrap learning to the time constant of learning with a teacher. It has been found by experiment that use of this factor allows one to make reasonably close estimates of learning-curve time constants for bootstrap learning.

To obtain a theoretical learning curve for bootstrap adaptation, we apply the $1/(p_+ - p_-)$ factor to (3). Thus the time constant for the LMS bootstrap process is

$$\tau_{\substack{\text{mse}\\\text{bootstrap}}} = \frac{n + 1}{2\alpha(p_+ - p_-)} \text{ adaptations}. \quad (39)$$

Two different kinds of learning curves are of interest, one being a plot of mse versus number of iterations, the other being a plot of error probability versus number of iterations. It has been pointed out in [1] that error probability and mse are approximately proportional over a wide range of conditions. Therefore, error-probability learning curves have similar time constants to those of mse learning curves. Formula (39) will be used in deriving an approximate error-probability learning curve for bootstrap adaptation.

The general differential equation for a simple exponential process is

$$\frac{d\psi}{dt} + \frac{\psi}{\tau} = 0 \quad (40)$$

where the parameter τ is the time constant. Since τ is a function of $(p_+ - p_-)$ and thereby is a function of ψ, this differential equation becomes, using (38) and (39),

$$\frac{d\psi}{dt} + \frac{8\alpha\sqrt{P_{\min}}}{(n + 1)\sqrt{2\pi D}}\psi^2 = 0.$$

Integrating yields

$$\psi = \frac{(n + 1)\sqrt{2\pi D}}{8\alpha\sqrt{P_{\min}}} \frac{1}{(t - t_0)} \quad (41)$$

where t_0 is a constant of integration, depending upon starting conditions, and t is the number of adaptations.

The learning curve for bootstrap adaptation is thus seen to be a rectangular hyperbola, as against an exponential for learning with a teacher. The asymptotic behavior of a hyperbola near optimal performance leads to poorer convergence than that of an exponential.

Improving Convergence by Strong Reward/Weak Punishment

The bootstrap learning process is quite efficient in the early stages, but deteriorates radically near optimal performance. At this stage most of the decisions of the adaptive system are optimal and deserve more rewarding than punishing. Different adaptation coefficients, α_+ (reward) and α_- (punish), are indicated.

When $\alpha_+ = \alpha_- = \alpha$, the average movement in adapting in the optimal direction is proportional to $\alpha(p_+ - p_-)$. The effect upon the learning time constant is given by (39). When $\alpha_+ \neq \alpha_-$, the average movement in the optimal direction is, using (23), proportional to

$$\alpha_+\left(\frac{\rho\sigma_h\sigma_g}{\sigma_h\sqrt{2\pi D}} + \frac{\bar{g}}{2}\right) - \alpha_-\left(\frac{-\rho\sigma_h\sigma_g}{\sigma_h\sqrt{2\pi D}} + \frac{\bar{g}}{2}\right) = \alpha_{\text{ave}}\left(\frac{2\rho\sigma_g}{\sqrt{2\pi D}} + \frac{(\alpha_+ - \alpha_-)}{(\alpha_+ + \alpha_-)}\bar{g}\right) \quad (42)$$

where $\alpha_{\text{ave}} \triangleq (\alpha_+ + \alpha_-)/2$. Hence, the time constant is

$$\tau_{\substack{\text{mse}\\\text{bootstrap}\\\alpha_+ \neq \alpha_-}} = \frac{(n + 1)}{2\alpha_{\text{ave}}\left[\frac{4\sqrt{P_{\min}}}{\sqrt{2\pi D}}\psi + \frac{(\alpha_+ - \alpha_-)}{2\alpha_{\text{ave}}}\left(1 - \frac{2P_{\min}}{1 - 2P_{\min}}\psi\right)\right]}. \quad (43)$$

This leads to the approximate differential equation

$$\frac{d\psi}{dt} + \frac{\psi}{(n + 1)}\left[\frac{8\sqrt{P_{\min}}}{\sqrt{2\pi D}}\alpha_{\text{ave}}\psi + (\alpha_+ - \alpha_-)\left(1 - \frac{2P_{\min}}{1 - 2P_{\min}}\psi\right)\right] = 0. \quad (44)$$

Solving (44), we have

$$\psi = \frac{C_1 \exp\left[-C_1(t - t_0)\right]}{1 - C_2 \exp\left[-C_1(t - t_0)\right]} \tag{45}$$

where C_1 and C_2 are given by

$$C_1 = \frac{\alpha_+ - \alpha_-}{(n + 1)}$$

$$C_2 = \left(\frac{8\alpha_{\text{ave}}\sqrt{P_{\min}}}{\sqrt{2\pi D}} - (\alpha_+ - \alpha_-)\frac{2P_{\min}}{1 - 2P_{\min}}\right)\Big/(n + 1).$$

From the foregoing we see that as learning proceeds the denominator of the expression for ψ in (45) approaches 1, and the value of the numerator therefore governs the rate of learning. The learning process near optimality is now exponential. Thus bootstrap learning can be improved while adapting near optimality by rewarding more strongly than punishing, i.e., by making the coefficient α_+ several times greater than α_-.

Application to Blackjack Learning Curve

The idealized bootstrap adaptation model applies to the blackjack example in the following way. The optimal system implements the Thorp optimal strategy. This is the system that the learning system attempts to emulate. It learns with the critic, which indicates at the end of each game the particular success or failure of the chain of decisions. The adaptive system has won (performance better than average, reward it) or lost (performance poorer than average, punish it). At the end of each decision chain (at the end of each game), perfect knowledge (the game was won or lost) is imparted ex post facto by the critic. The number of decisions D per game is close to four on the average.

The minimum error probability $P_{\min}$ of the optimal system may be estimated in the following manner. The Thorp optimal strategy for the simplified game wins 49.5 percent of the games. In the majority of games played, three right decisions are made first. The fourth and last decision is the critical one, and this decision is right roughly half the time (corresponding to the winning games). Therefore, $P_{\min}$ is estimated to be 1/8.

The quantity ψ appearing in (38) can be determined for blackjack by subtracting the minimum rate of loss of the optimal system (50.5 percent) from the rate of loss of the learning system and dividing this difference by the minimum rate of loss.

Substituting $P_{\min} = 1/8$ and $D = 4$ in (41), for $\alpha_+ = \alpha_- = \alpha$, we have

$$\psi = \frac{9.3}{\alpha(t - t_0)}. \tag{46a}$$

Furthermore, for $\alpha_+ \neq \alpha_-$, from (45), we have

$$\psi = \frac{C_1 \exp\left[-C_1(t - t_0)\right]}{1 - C_2 \exp\left[-C_1(t - t_0)\right]} \tag{46b}$$

where

$$C_1 = \frac{\alpha_+ - \alpha_-}{21}$$

$$C_2 = \frac{\alpha_{\text{ave}}}{9.3} - \frac{(\alpha_+ - \alpha_-)}{15.75}.$$

In both equations, t is expressed in number of games and t_0 is an undetermined constant of integration that must be found for each experiment. Its value depends upon initial conditions.

Equations (46a) and (46b) should be regarded as only approximate because the model does not perfectly fit the blackjack game for the following reasons.

1) The decisions (in chains of length D) are not independent: once a stick decision is made, subsequent decisions are automatically decided.

2) Input vectors are not uncorrelated. The sum of the player's cards is cumulative and therefore is first-order Markov.

3) The average number of cards drawn per game being approximately four, D is a small number. The Gaussian approximation (application of central limit theorem) used in deriving (37) is therefore quite crude.

4) A small percentage of blackjack games cannot be won by the player, even with perfect knowledge of the dealer's deck. It is thus possible to lose making right (perfect-knowledge) decisions. In such cases, the concept of right/wrong is not applicable.

Despite these discrepancies, it has been shown by extensive experimentation that observed blackjack learning curves agree remarkably well with theoretical learning curves based on the idealized model.

Experimental and Theoretical Results

A series of computer-simulated experiments was carried out to check the applicability of the theoretical model and the assumptions made in deriving (46a) and (46b). Typical experimental and theoretical learning curves are shown in Figs. 10–12. Percent games won versus number of games played are plotted. In each case, the undetermined constant t_0 in the equations was chosen to achieve best fit between experimental and theoretical curves.

When $\alpha_+ = \alpha_-$, (41) gives the theoretical learning curve in terms of ψ. Expressed in terms of winning rate, the derived hyperbola is superposed on the experimental curve in Fig. 10. The fit is quite good. For this experiment, $\alpha_+ = \alpha_- = \alpha = 0.4$.

The dotted experimental curve of Fig. 10 was derived in the following manner. An ensemble of 1000 learning experiments was performed, each run starting with the same initial weight vector. During each run, 10 games were played, with bootstrap adaptation after each game, and the average percentage of games won was computed. A new average was computed over the next 10 games, and so on, until 1000 games were played. The weight vector was then reset to the initial condition and a new experiment was

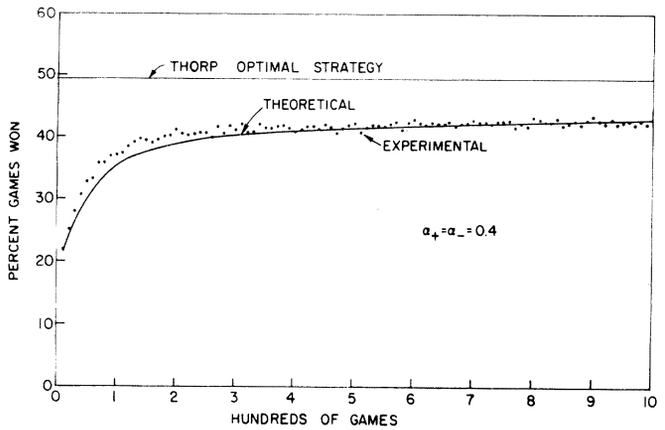


Fig. 10. Learning curves for Adaline playing blackjack ($\alpha_+ = \alpha_- = 0.4$).

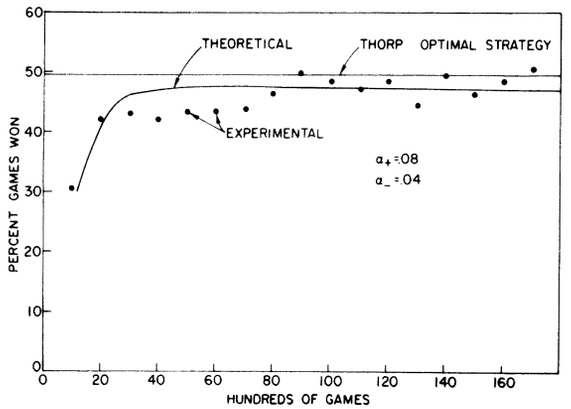


Fig. 11. Learning curves for Adaline playing blackjack ($\alpha_+ = \alpha_- = 0.04$).

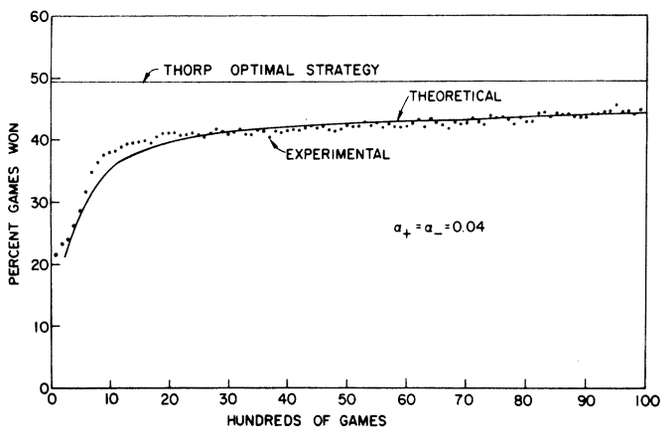


Fig. 12. Learning curves for Adaline playing blackjack ($\alpha_+ = 0.08$, $\alpha_- = 0.04$).

begun. Each point of the dotted curve is an average percentage of games won, derived from 1000 ensemble members, 10 games per ensemble member per point.

Each experimental point of Fig. 10 represents a time and ensemble average derived over 10 000 games of play. The averaging provides a performance evaluation with a standard deviation of error in mean of approximately $\frac{1}{2}$ percent. It should be noted that the adaptive system was able to do a substantial amount of learning within several hundred games, having no knowledge of the rules and objectives of play. The nonexponential nature of the learning process is evident from this experiment. Asymptotic convergence is a very slow process. The asymptotic level of performance is somewhat lower than that of the Thorp optimal system. Finite speed of adaptation causes misadjustment [32] due to adaptation noise in the weight vector, precluding optimal performance.

By adapting more slowly, performance closer to the Thorp optimal is attainable. When α is reduced by a factor of 10, the results, which are shown in Fig. 11, are very similar to those of Fig. 10, except that the time scale is compressed tenfold and the asymptotic performance approaches much more closely that of the Thorp optimal system (it should be about 10 times closer, but this is difficult to determine experimentally).

The dotted curve of Fig. 11 was obtained by averaging over blocks of 100 games per ensemble member and then averaging over the ensemble of 100 experiments. Each point represents a time and ensemble average over 10 000 games.

Experiments were performed with unequal α_+ and α_- to obtain exponential rather than hyperbolic asymptotic behavior. A typical experiment is shown in Fig. 12. The theoretical curve was obtained using (46b). Each experimental point was obtained from averaging over 1000 games, rather than over 10 000 games, as was done in the previous experiments. There was no ensemble averaging. In this case, $\alpha_+ = 0.08$ and $\alpha_- = 0.04$. The general speed of adaptation lies between those in the previous experiments. The asymptotic approach is much surer and more nearly exponential. Sometimes the performance exceeds Thorp optimal, but this only happens, on the average, over a finite number of games.

Current and Future Research

Preliminary studies have been made with some success toward the development of adaptation algorithms for multilayered networks of adaptive threshold elements using the selective bootstrap principle. If performance observed at a set of output terminals is better than average, every element in the net is rewarded. If output-terminal performance is poorer than average, then all elements are punished. By this procedure, desired-response training signals are supplied to adaptive elements at intermediate stages. It is expected that (39) and (43) will be usable in predicting the rate of adaptation for such networks. Instead of decisions being made in a chain over time, they are made simultaneously, in a chain over "space."

The ultimate purpose of this research is to develop efficient adaptation algorithms for adaptive threshold-element networks of arbitrary configuration which are capable of realizing decision functions that are not necessarily linearly separable.

Applications of these principles to adaptive on–off control are also being pursued. Adaptive controllers have undergone useful learning when better-than-average performance is rewarded and poorer-than-average performance is punished. A second-order "broom-balancing" system has

been controlled by a linear threshold controller which is connected to the state variables and which learns by bootstrap adaptation. The research is being extended to fourth order systems and preliminary results are most encouraging. This work and the applications to multilayered adaptive nets will be reported in the future.

Acknowledgment

The authors wish to express their appreciation for the substantial efforts of Ms. Mabel Rockwell in the editing of this paper. Her many useful suggestions helped improve its clarity and organization. The idea of using the game blackjack for bootstrap learning experiments was proposed by Dr. Fred W. Smith. The earliest computer programs for blackjack play were prepared by Dr. James S. Koford. He was the first to make bootstrap learning work. The experiments reported in this paper were run by Dr. Irwin Sobel and Ms. Betty Earlougher. We are grateful for their contributions.

References

[1] B. Widrow and M. E. Hoff, "Adaptive switching circuits," in *1960 WESCON Conv. Rec.*, pt. 4.
[2] F. Rosenblatt, *Principles of Neurodynamics, Perceptions, and the Theory of Brain Mechanisms.* Washington, D.C.: Spartan, 1962.
[3] H. D. Block, "The Perceptron: a model for brain functioning, I," *Rev. Mod. Phys.*, vol. 34, pp. 123–135, Jan. 1966.
[4] N. L. Nilsson, *Learning Machines.* New York: McGraw-Hill, 1965.
[5] O. Firschein and M. Fischler, "Automatic subclass determination for pattern-recognition applications," *IEEE Trans. Electron. Comput.* (Corresp.), vol. EC-12, pp. 137–141, Apr. 1963.
[6] R. L. Mattson and J. E. Dammann, "A technique for determining and coding subclasses in pattern recognition problems," *IBM J. Res. Develop.*, vol. 9, pp. 294–302, July 1965.
[7] G. Nagy and G. L. Shelton, Jr., "Self-corrective character recognition system," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 215–222, Apr. 1966.
[8] G. Nagy, "The state of art in pattern recognition," *Proc. IEEE*, vol. 56, pp. 836–862, May 1968.
[9] Y. C. Ho and A. K. Agrawala, "On pattern classification algorithms: introduction and survey," *Proc. IEEE*, vol. 56, pp. 2101–2114, Dec. 1968.
[10] J. S. Koford and G. F. Groner, "The use of an adaptive threshold element to design a linear optimal pattern classifier," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 42–50, Jan. 1966.
[11] L. R. Talbert, "The sum-line extrapolative algorithm and its application to statistical classification problems," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-6, pp. 229–239, July 1970.
[12] I. Morishita, "Analysis of an adaptive threshold logic unit," *IEEE Trans. Comput.*, vol. C-19, pp. 1181–1192, Dec. 1970.
[13] A. M. Dorofeyuk, "Automatic classification algorithms," *Automat. Remote Contr.* (Rev. Paper), vol. 32, pp. 1928–1958, Dec. 1971.
[14] J. Mendel and K. S. Fu, *Adaptive Learning and Pattern Recognition Systems: Theory and Applications.* New York: Academic Press, 1970.
[15] C. H. Mays, "Effects of adaptation parameters on convergence time and tolerance for adaptive threshold elements," *IEEE Trans. Electron. Comput.* (Short Notes), vol. EC-13, pp. 465–468, Aug. 1964.
[16] K. Steinbuch and B. Widrow, "A critical comparison of two kinds of adaptive classification networks," *IEEE Trans. Electron. Comput.* (Short Notes), vol. EC-14, pp. 737–740, Oct. 1965.
[17] D. B. Cooper and P. W. Cooper, "Adaptive pattern recognition and signal detection without supervision," in *1964 Int. Conv. Rec.*, pt. I, pp. 246–256.
[18] E. M. Braverman, "The potential functions method in the problem of unsupervised pattern recognition machine learning," *Avtomat. Telemekh.*, no. 11, 1965.
[19] G. Ball and D. Hall, "ISODATA, a novel method of data analysis and pattern classification," Stanford Res. Inst., Menlo Park, Calif., Tech. Rep. NTIS AD-699-616, Apr. 1965.
[20] E. A. Patrick and J. C. Hancock, "Nonsupervised sequential classification and recognition of patterns," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 362–372, July 1966.
[21] Ž. J. Nicolić and K. S. Fu, "An algorithm for learning without external supervision and its application to learning control systems," *IEEE Trans. Automat. Contr.*, vol. AC-11, pp. 414–422, July 1966.
[22] J. Spragins, "Learning without a teacher," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 223–230, Apr. 1966.
[23] S. C. Fralick, "Learning to recognize patterns without a teacher," *IEEE Trans. Inform. Theory*, vol. IT-13, pp. 57–64, Jan. 1967.
[24] W. C. Miller, "A modified mean square error criterion for use in unsupervised learning," Stanford Electron. Lab., Stanford, Calif., Tech. Rep. SEL-67-066 (TR 6778-2), Aug. 1967.
[25] Ya. Z. Tsypkin, "Self-learning—what is it?," *IEEE Trans. Automat. Contr.*, vol. AC-13, pp. 608–612, Dec. 1968.
[26] C. G. Hilborn, Jr., and D. G. Lainiotis, "Unsupervised learning minimum risk pattern classification for dependent hypotheses and dependent measurements," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-5, pp. 109–115, Apr. 1969.
[27] D. B. Cooper and J. H. Freeman, "On the asymptotic improvement in the outcome of supervised learning provided by additional nonsupervised learning," *IEEE Trans. Comput.*, vol. C-19, pp. 1055–1063, Nov. 1970.
[28] R. W. Lucky and H. R. Rudin, "An automatic equalizer for general purpose communication channels," *Bell Syst. Tech. J.*, vol. 46, pp. 2179–2209, Nov. 1967.
[29] A. Lender, "Decision-directed digital adaptive equalization technique for high-speed data transmission," in *Proc. 1970 Int. Conf. Commun.*, pp. 4-18–4-21.
[30] B. Widrow, "Adaptive filters I—fundamentals," Stanford Electron. Lab., Stanford, Calif., Rep. SEL-66-126 (TR No. 6764-6), 1966.
[31] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive antenna systems," *Proc. IEEE*, vol. 55, pp. 2143–2159, Dec. 1967.
[32] B. Widrow, "Adaptive filters," in *Aspects of Network in Systems Theory*, R. E. Kalman and N. DeClaris, Eds. New York: Holt, Rinehart and Winston, 1971, pp. 563–587.
[33] J. I. Nagumo and A. Noda, "A learning method for system identification," *IEEE Trans. Automat. Contr.*, vol. AC-12, pp. 282–287, June 1967.
[34] E. O. Thorp, *Beat the Dealer: A Winning Strategy for the Game of Twenty-One.* New York: Random House, 1966.
[35] J. Scarne, *Scarne's Complete Guide to Gambling.* New York: Simon and Schuster, 1961.
[36] F. W. Smith, "A trainable nonlinear function generator," *IEEE Trans. Automat. Contr.*, vol. AC-11, pp. 212–218, Apr. 1966.
[37] R. C. Singleton, "A test for linear separability as applied to self organizing machines," in *Self Organizing Systems*, M. C. Youits, G. T. Jacobi, and G. D. Goldstein, Eds. Washington, D.C.: Spartan, 1962.
[38] F. W. Smith, "Design of quasi-optimal minimum-time controllers," *IEEE Trans. Automat. Contr.*, vol. AC-11, pp. 71–77, Jan. 1966.