September 14, 1992 LIDS -P-2131

A Nonparametric Training Algorithm for Decentralized Binary Hypothesis Testing Networks *

John Wissinger and Michael Athans
Room 35-406
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139

Abstract

We derive a nonparametric training algorithm which asymptotically achieves the minimum possible error rate, over the set of linear classifiers, for decentralized binary hypothesis testing (detection) networks. The training procedure is nonparametric in the sense that it does not require the functional form of the probability densities or the prior probabilities to yield an optimal set of decision thresholds in the limit. However, knowledge of the network topology is required by the algorithm. We suggest that models of the variety in this study provide a paradigm for the study of adaptation in human organizations.

1 Introduction

We have been attempting to model and analyze adaptation in coupled organized systems by devising algorithms

*This work was supported by the National Science Foundation through an NSF Graduate Fellowship and under Grant NSF/IRI-8902755 (under a subcontract from the University of Connecticut)

This paper was submitted to the 1993 American Control Conference.

which train decentralized detection networks to optimize a given measure of organizational performance.

In this paper, we present one of a collection of training algorithms we have developed. The algorithm uses successive approximation to solve for the optimal decision rules, with stochastic approximations taking the place of function evaluations. Each network node, hereafter denoted decision maker (DM), solves approximately using stochastic approximation a subproblem which is coupled to subproblems being solved by the other DMs in the network. The computation is truly distributed as the DMs must communicate partial results with one another and update their subproblems upon receiving new information.

2 Decentralized Binary Hypothesis Testing

This section presents two examples of decentralized binary hypothesis testing problems which will be used to demonstrate the algorithm. The examples are small networks which give clear evidence of the noncombinatorial type complexity typical of these problems. We note that an excellent and very general overview of the field of decentralized detection theory is presented in [12], while the small teams which concern us have been extensively studied in [4].

A critical assumption in this work is that the observations of the DMs are conditionally independent. Without this assumption, the decision rules not only become messy, but the problem has been shown to be NP-complete [12], [13]. We also assume for simplicity that the observations are scalar valued, although generalization to vector iid observations presents no difficulty.

The decision criterion on which we focus, namely the minimum probability of error criterion, will penalize only the incorrect decisions of the so called *primary* DM, or the DM which outputs the final decision of the network. The role of the other DMs in the organization is simply to contribute to the decision process in a way which minimizes the probability of the primary DM being incorrect. Their individual decisions are not reflected in the cost. We have chosen the minimum probability of error criterion for its intuitive compatability with the organizational paradigm as well as its general usefulness.

A critical issue which arises in the team problem is the size of the message set available to each DM. We restrict ourselves to the case in which each DM chooses messages from the set $\{0, 1\}$. Thus, each DM will be allowed only one bit of communication capacity. Allowing more messages clearly improves the performance of the team since in the limit each DM will be able to transmit its entire observation, thus achieving the centralized solution which provides an unattainable lower bound on the performance of the team.

2.1 Two-Member Tandem Architecture

The first topology we will examine is illustrated in Fig.1. This type of structure is referred to as *tandem*, and the two-member tandem team will hereafter be referred to as 2-Tand. The operation of 2-Tand may be described as follows: DM A receives a scalar observation y_A which it uses to choose a message $u_A \in \{0, 1\}$ to send to DM B. DM B, the primary DM, takes into account the message u_A from DM A as well as its own observation y_B to compute the overall team decision $u_B \in \{0, 1\}$. Thus, if $\mathbf{Y_A}$ and $\mathbf{Y_B}$ denote the respective observation spaces of A and B, then the decision rules employed by A and B are of the form $\gamma_A : \mathbf{Y_A} \rightarrow \{0, 1\}$ and $\gamma_B : \mathbf{Y_B} \times \{0, 1\} \rightarrow \{0, 1\}$ respectively. It remains to determine just what γ_A and γ_B should be in view of the minimum probability of error criterion.

The *necessary* conditions for optimality of the decision rules γ_A and γ_B, under the minimum probability of error criterion, were derived in [4]. Perhaps surprisingly, as in the centralized case, they are likelihood ratio tests (LRTs). With $\eta \triangleq \frac{p_0}{p_1}$, these LRTs are

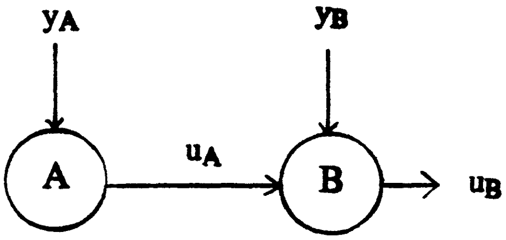


Figure 1: 2-Tand

For DM B:

$$\frac{p(y_B|H_1)}{p(y_B|H_0)} \underset{u_B=0}{\overset{u_B=1}{\gtrless}} \begin{cases} \eta\frac{1-P_F^A}{1-P_D^A} & \text{if } u_A = 0 \\ \\ \eta\frac{P_F^A}{P_D^A} & \text{if } u_A = 1 \end{cases} \tag{2.1}$$

For DM A:

$$\frac{p(y_A|H_1)}{p(y_A|H_0)} \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \eta\frac{P_F^{B1} - P_F^{B0}}{P_D^{B1} - P_D^{B0}} \tag{2.2}$$

where P_F^A, P_D^A are the probabilities of false alarm and detection of A and P_F^{Bi}, P_D^{Bi}, $i = 0, 1$ are the probabilities of false alarm and detection of B when A selects message i.

A significant reduction in the decision rules of equations 2.1 - 2.2 can be effected if we work with a particular class of decision problems, namely those in which the network's objective is to decide which of two constant signals occurred with each DM's measurement corrupted by zero-mean Gaussian noise. For this case the observations at A and B are of the form

$$y_A = \begin{cases} \mu_0 + N(0, \sigma_A^2) & : H_0 \\ \mu_1 + N(0, \sigma_A^2) & : H_1 \end{cases} \tag{2.3}$$

$$y_B = \begin{cases} \mu_0 + N(0, \sigma_B^2) & : H_0 \\ \mu_1 + N(0, \sigma_B^2) & : H_1 \end{cases} \tag{2.4}$$

We can use the ln(·) function to reduce 2.1 - 2.2 to the following set of equivalent optimal tests, which happen to be *linear* in the observations:
For DM B:

$$y_B \underset{u_B=0}{\overset{u_B=1}{\gtrless}} \begin{cases} \beta_0 & \text{if } u_A = 0 \\ \\ \beta_1 & \text{if } u_A = 1 \end{cases} \tag{2.5}$$

For DM A:

$$y_A \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \alpha \tag{2.6}$$

where the fixed *observation axis* thresholds α, β_0, β_1 are determined by a system of three coupled nonlinear algebraic equations. First define, for $k = 0, 1$, the functions

$$\Phi_\alpha(k) = \int_{-\infty}^{\frac{\alpha-\mu_k}{\sigma_A}} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx \tag{2.7}$$

$$\Phi_{\beta_0}(k) = \int_{-\infty}^{\frac{\beta_0-\mu_k}{\sigma_B}} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx \tag{2.8}$$

$$\Phi_{\beta_1}(k) = \int_{-\infty}^{\frac{\beta_1-\mu_k}{\sigma_B}} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx \tag{2.9}$$

Then the thresholds are given by

$$\beta_0 = \frac{\sigma_B^2}{\mu_1 - \mu_0} \ln\left(\frac{\Phi_\alpha(0)}{\Phi_\alpha(1)}\right) + \frac{\sigma_B^2}{\mu_1 - \mu_0} \ln\left(\frac{p0}{p1}\right) + \frac{\mu_0 + \mu_1}{2} \tag{2.10}$$

$$\beta_1 = \frac{\sigma_B^2}{\mu_1 - \mu_0} \ln\left(\frac{1 - \Phi_\alpha(0)}{1 - \Phi_\alpha(1)}\right) + \frac{\sigma_B^2}{\mu_1 - \mu_0} \ln\left(\frac{p0}{p1}\right) + \frac{\mu_0 + \mu_1}{2} \tag{2.11}$$

$$\alpha = \frac{\sigma_A^2}{\mu_1 - \mu_0} \ln\left(\frac{\Phi_{\beta_0}(0) - \Phi_{\beta_1}(0)}{\Phi_{\beta_0}(1) - \Phi_{\beta_1}(1)}\right) + \frac{\sigma_A^2}{\mu_1 - \mu_0} \ln\left(\frac{p0}{p1}\right) + \frac{\mu_0 + \mu_1}{2} \tag{2.12}$$

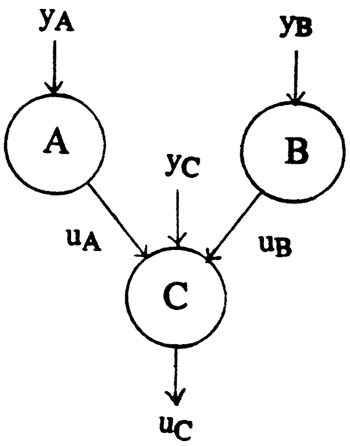


Figure 2: 3-Vee

2.2 Three-member V architecture

The three-member V structure is illustrated in Fig.2. The decision rules employed by DM A and DM B are of the form $\gamma_A : \mathbf{Y_A} \rightarrow \{0,1\}$ and $\gamma_B : \mathbf{Y_B} \rightarrow \{0,1\}$, while the decision rule of DM C is of the form $\gamma_C : \mathbf{Y_C} \times \{0,1\} \times \{0,1\} \rightarrow \{0,1\}$.

The necessary conditions for optimality of the decision rules γ_A, γ_B, and γ_C under the minimum probability of error criterion were derived in [4] and are given by
For DM C:

$$\frac{p(y_C|H_1)}{p(y_C|H_0)} \underset{u_C=0}{\overset{u_C=1}{\gtrless}} \begin{cases} \eta \frac{(1-P_F^A)(1-P_F^B)}{(1-P_D^A)(1-P_D^B)} & \text{if } u_A = 0, u_B = 0 \\ \eta \frac{(1-P_F^A)P_F^B}{(1-P_D^A)P_D^B} & \text{if } u_A = 0, u_B = 1 \\ \eta \frac{P_F^A(1-P_F^B)}{P_D^A(1-P_D^B)} & \text{if } u_A = 1, u_B = 0 \\ \eta \frac{P_F^A P_F^B}{P_D^A P_D^B} & \text{if } u_A = 1, u_B = 1 \end{cases} \tag{2.13}$$

For DM A:

$$\frac{p(y_A|H_1)}{p(y_A|H_0)} \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \eta\frac{(1-P_F^B)[P_F^{C(10)}-P_F^{C(00)}]+P_F^B[P_F^{C(11)}-P_F^{C(01)}]}{(1-P_D^B)[P_D^{C(10)}-P_D^{C(00)}]+P_D^B[P_D^{C(11)}-P_D^{C(01)}]} \tag{2.14}$$

For DM B:

$$\frac{p(y_B|H_1)}{p(y_B|H_0)} \underset{u_B=0}{\overset{u_B=1}{\gtrless}} \eta\frac{(1-P_F^A)[P_F^{C(01)}-P_F^{C(00)}]+P_F^A[P_F^{C(11)}-P_F^{C(10)}]}{(1-P_D^A)[P_D^{C(01)}-P_D^{C(00)}]+P_D^A[P_D^{C(11)}-P_D^{C(10)}]} \tag{2.15}$$

where $P_F^{C(ij)}, P_D^{C(ij)}$ denote the probabilities of false alarm and detection of DM C when receiving messages $u_A = i, u_B = j$ with $i, j \in \{0, 1\}$.

For the Gaussian problem, the decision rules of equations 2.13 - 2.15 may be reduced to the equivalent linear rules
For DM C:

$$y_C \underset{u_C=0}{\overset{u_C=1}{\gtrless}} \begin{cases} \xi_{00} & \text{if } u_A = 0, u_B = 0 \\ \xi_{01} & \text{if } u_A = 0, u_B = 1 \\ \xi_{10} & \text{if } u_A = 1, u_B = 0 \\ \xi_{11} & \text{if } u_A = 1, u_B = 1 \end{cases} \tag{2.16}$$

For DM A:

$$y_A \underset{u_A=0}{\overset{u_A=1}{\gtrless}} \alpha \tag{2.17}$$

For DM B:

$$y_B \underset{u_B=0}{\overset{u_B=1}{\gtrless}} \beta \tag{2.18}$$

where the fixed observation axis thresholds $\alpha, \beta, \xi_{00}, \xi_{01}, \xi_{10}, \xi_{11}$ are determined by the following system of six coupled nonlinear algebraic equations (with the functions Φ defined similarly to those in equations 2.7-2.9):

$$\xi_{00} = \frac{\sigma_C^2}{\mu_1 - \mu_0} \ln\left(\frac{\Phi_\alpha(0)\Phi_\beta(0)}{\Phi_\alpha(1)\Phi_\beta(1)}\right) + \frac{\sigma_C^2}{\mu_1 - \mu_0} \ln\left(\frac{p0}{p1}\right) + \frac{\mu_0 + \mu_1}{2} \tag{2.19}$$

$$\xi_{01} = \frac{\sigma_C^2}{\mu_1 - \mu_0} \ln \left(\frac{\Phi_\alpha(0)(1 - \Phi_\beta(0))}{\Phi_\alpha(1)(1 - \Phi_\beta(1))} \right) + \frac{\sigma_C^2}{\mu_1 - \mu_0} \ln \left(\frac{p0}{p1} \right) + \frac{\mu_0 + \mu_1}{2} \tag{2.20}$$

$$\xi_{10} = \frac{\sigma_C^2}{\mu_1 - \mu_0} \ln \left(\frac{(1 - \Phi_\alpha(0))\Phi_\beta(0)}{(1 - \Phi_\alpha(1))\Phi_\beta(1)} \right) + \frac{\sigma_C^2}{\mu_1 - \mu_0} \ln \left(\frac{p0}{p1} \right) + \frac{\mu_0 + \mu_1}{2} \tag{2.21}$$

$$\xi_{11} = \frac{\sigma_C^2}{\mu_1 - \mu_0} \ln \left(\frac{(1 - \Phi_\alpha(0))(1 - \Phi_\beta(0))}{(1 - \Phi_\alpha(1))(1 - \Phi_\beta(1))} \right) + \frac{\sigma_C^2}{\mu_1 - \mu_0} \ln \left(\frac{p0}{p1} \right) + \frac{\mu_0 + \mu_1}{2} \tag{2.22}$$

$$\begin{aligned} \alpha &= \frac{\sigma_A^2}{\mu_1 - \mu_0} \ln \left(\frac{\Phi_\beta(0)[\Phi_{\xi_{00}}(0) - \Phi_{\xi_{10}}(0)] + (1 - \Phi_\beta(0))[\Phi_{\xi_{01}}(0) - \Phi_{\xi_{11}}(0)]}{\Phi_\beta(1)[\Phi_{\xi_{00}}(1) - \Phi_{\xi_{10}}(1)] + (1 - \Phi_\beta(1))[\Phi_{\xi_{01}}(1) - \Phi_{\xi_{11}}(1)]} \right) \\ &+ \frac{\sigma_A^2}{\mu_1 - \mu_0} \ln \left(\frac{p0}{p1} \right) + \frac{\mu_0 + \mu_1}{2} \end{aligned} \tag{2.23}$$

$$\begin{aligned} \beta &= \frac{\sigma_B^2}{\mu_1 - \mu_0} \ln \left(\frac{\Phi_\alpha(0)[\Phi_{\xi_{00}}(0) - \Phi_{\xi_{01}}(0)] + (1 - \Phi_\alpha(0))[\Phi_{\xi_{10}}(0) - \Phi_{\xi_{11}}(0)]}{\Phi_\alpha(1)[\Phi_{\xi_{00}}(1) - \Phi_{\xi_{01}}(1)] + (1 - \Phi_\alpha(1))[\Phi_{\xi_{10}}(1) - \Phi_{\xi_{11}}(1)]} \right) \\ &+ \frac{\sigma_B^2}{\mu_1 - \mu_0} \ln \left(\frac{p0}{p1} \right) + \frac{\mu_0 + \mu_1}{2} \end{aligned} \tag{2.24}$$

3 The Training Algorithm

By *training* we refer to the use of a set of correctly classified examples to evaluate and subsequently improve the decision process. Each training example for the network consists of a set of sensor observations together with the desired network output, i.e. the correct hypothesis. We assume that during training the statistics according to which the observations are generated are stationary. Using this information, the DMs attempt to gradually adapt their decision rules, i.e. the locations of their *observation axis thresholds*, so that the organization will have a lower error rate in the future. Training schemes of this variety are known as "supervised learning" schemes.

Of course, the training problem posed here really represents an attempt to minimize the error probability through the (possibly extensive) use of labeled training data. The object of merit is the probability of error function $P_\epsilon(\cdot)$, and the set of thresholds resulting from training are to be judged by how close they come to the thresholds which result from optimizing $P_\epsilon(\cdot)$. This optimization, as can already be surmised, is nontrivial. It is discussed extensively in [10].

3.1 Successive Approximation

As we have seen for the Gaussian case, the optimal thresholds may be expressed as a system of coupled nonlinear algebraic equations which specify the necessary conditions for optimality. This system is often solved in practice using successive approximation to obtain a fixed point solution. The thresholds are arbitrarily initialized and then the equations are iterated until the thresholds converge. More precisely, for 2-Tand the process is as follows. Note that equations 2.10 - 2.12 are of the form

$$\begin{aligned} \beta_0 &= f(\alpha) \\ \beta_1 &= g(\alpha) \\ \alpha &= h(\beta_0, \beta_1) \qquad (3.25) \end{aligned}$$

This system can be solved by fixing β_0 and β_1 and then determining α, then plugging in the new α to update β_0 and β_1 and so on, provided that certain conditions are met. In particular, in order to guarantee the existence and uniqueness of a fixed point solution, as well as convergence from arbitrary starting values, the system must be shown to have a contraction property [1].

Unfortunately, showing this property directly is difficult since the Gaussian error functions are very algebraically cumbersome. However, our numerical studies have uncovered no problems whatsoever with the convergence of successive approximation. It appears numerically that these systems do possess the required contraction property, at least for those cases we have studied.

Prompted by the success of the successive approximation method, we developed a nonparametric training method based on the same approach. It required finding a way to make the correct updates without doing any function evaluations. We subsequently discovered, in the adaptive pattern classification literature, a modification of the Robbins-Monro stochastic approximation method [8] which was well suited for solving this problem and which we subsequently extended to the decentralized setting.

3.2 Stochastic Approximation

In this section, we modify the "window algorithm" presented in [9], [15], [16] to construct a nonparametric distributed training method.

The window algorithm is based on the following idea. Recall that the optimal minimum probability of error decision rule for the centralized binary classification problem is the LRT

$$\frac{p(y|H_1)}{p(y|H_0)} \underset{u=0}{\overset{u=1}{\gtrless}} \frac{p_0}{p_1} \qquad (3.26)$$

It is noted in [5] that if the functions $p_0 p(y|H_0)$ and $p_1 p(y|H_1)$ have a single point of intersection, the minimum probability of error threshold K^* is that value of K satisfying

$$p_0 p(y = K|H_0) \;=\; p_1 p(y = K|H_1) \qquad (3.27)$$

For the Gaussian detection problem, this equation is in fact satisfied at a unique point, i.e. there is only one point of intersection of the scaled densities. Condition 3.27 is intuitively clear since, for the value $y = K^*$, the LRT is satisfied with equality. If $p_0 = p_1 = \frac{1}{2}$, it is easily seen that the mimimum probability of error point corresponds to the point at which $P_F = P_M (= 1 - P_D)$. This is known as the "equal error" pt. since for this value of the threshold, equal proportions of both types of errors are made.

A natural approach is to try and determine K^* using stochastic approximation techniques. In particular, if the function $G(y) = p_1 p(y|H_1) - p_0 p(y|H_0)$ were a regression function, meaning that if it were equal to the expected value of some measurable random variable which is a function of y, then the Robbins-Monro method [8] could be applied directly to find K^* as the zero of $G(y)$. In [15] the authors point out that $G(y)$ does not seem to be a regression function. However, they are able to define a sequence of functions $\hat{G}(y, c)$, each of which is a regression function and which approximate $G(y)$ in the limit as $c \to 0$. Using these approximating functions, the following solution to the above classification problem is presented.

An adaptive classifier, which asymptotically coverges to K^* with probability 1 under very reasonable conditions, is given by the following algorithm from [15] where K_n is the threshold at iteration n, y_n is the nth observation, ρ_n is a stepsize decreasing as $\frac{1}{\sqrt{n}}$, and c_n is the window width which is also decreasing as $\frac{1}{\sqrt{n}}$:

$$K_{n+1} = \begin{cases} K_n - \rho_n & \text{if } |K_n - y_n| \leq c_n, \ y_n \text{ from } H_0 \\ K_n & \text{if } |K_n - y_n| > c_n \\ K_n + \rho_n & \text{if } |K_n - y_n| \leq c_n, \ y_n \text{ from } H_1 \end{cases} \tag{3.28}$$

This algorithm is actually a special case, using a rectangular window, of a more general algorithm which can employ several types of windows. Operation of the algorithm is depicted in Fig.3.

There are several interesting aspects to note. First, corrections to the current threshold are made up or down depending on whether or not an observation from a certain class falls within the rectangular window and do not depend on which side of the threshold the observation falls. This means that the actual decisions are irrelevant in the training process. Second, the solution is completely nonparametric. The functional forms of the densities are not required, and neither are the prior probabilities. Of course, the resulting linear classifier will only be optimal provided that the optimal classifier is linear, as in the Gaussian problems we are considering.

In [15] a generalization of the above algorithm is also con-

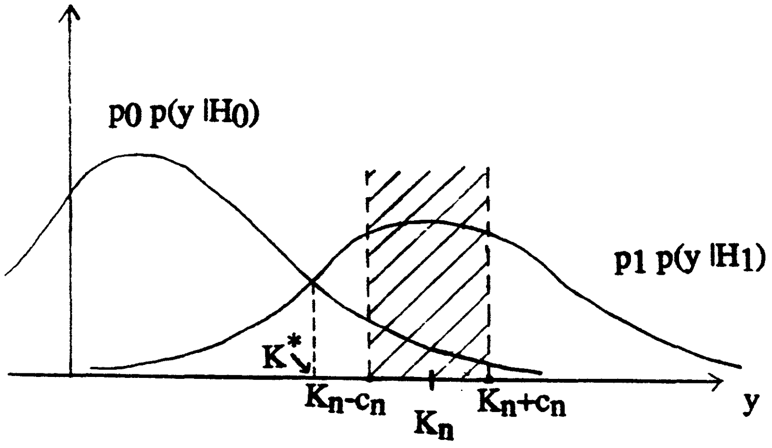


Figure 3: Window Algorithm using Rectangular Window

sidered, whereby unequal costs on the types of errors made are considered. This generalization says that the zero of an equation of the form $G(y) = \lambda_1 p_1 p(y|H_1) - \lambda_0 p_0 p(y|H_0)$, where λ_0, λ_1 are real constants, may be found with the modified algorithm

$$K_{n+1} = \begin{cases} K_n - (1-L)\rho_n & \text{if } |K_n - y_n| \leq c_n,\ y_n \text{ from } H_0 \\ K_n & \text{if } |K_n - y_n| > c_n \\ K_n + L\rho_n & \text{if } |K_n - y_n| \leq c_n,\ y_n \text{ from } H_1 \end{cases} \tag{3.29}$$

where

$$L = \frac{\lambda_0}{\lambda_0 + \lambda_1} \tag{3.30}$$

Thus, the modified algorithm takes into account the unequal costs by incorporating a stepsize bias which is proportional to the cost. This bias had to be incorporated directly since it is not inherent in the data.

The relevance of the unequal cost modification to the decentralized problem is that the coupling probabilities in the decentralized problem may be treated like costs. To see this, consider 2-Tand. If we define the three functions (from equations 2.1 - 2.2):

$$G_{B0}(y_B) = (1 - P_D^A)p_1 p(y_B|H_1) - (1 - P_F^A)p_0 p(y_B|H_0)$$

$$\begin{aligned} G_{B1}(y_B) &= P_D^A p_1 p(y_B|H_1) - P_F^A p_0 p(y_B|H_0) \\ G_A(y_A) &= [P_D^{B1} - P_D^{B0}] p_1 p(y_A|H_1) - [P_F^{B1} - P_F^{B0}] p_0 p(y_A|H_0) \end{aligned}$$

we can use the modified method to solve these equations. The coupling probabilities are incorporated into the algorithm as stepsize bias.

The training algorithm involves using the above method to have each DM solve a succession of local stochastic approximation subproblems which are coupled to subproblems being solved by the other DMs in the network. Each subproblem allows a DM to determine the proper setting of its observation threshold given information about the settings of the other DMs in the network, and it accomplishes this without doing any function evaluations.

The algorithm operates as follows for 2-Tand. Thresholds β_0 and β_1 of DM B are initialized and held fixed while estimates of the corresponding conditional probabilities $\tilde{P}_F^{B0}$, $\tilde{P}_D^{B0}$, $\tilde{P}_F^{B1}$, $\tilde{P}_D^{B1}$ are obtained using training data. The estimates are taken to be the observed relative frequency of each type of outcome over a "sufficiently large" number of trials. The estimated probabilities are then communicated to DM A which computes estimates of the coupling terms as $[\tilde{P}_D^{B1} - \tilde{P}_D^{B0}]$ and $[\tilde{P}_F^{B1} - \tilde{P}_F^{B0}]$ and then uses the stochastic approximation algorithm to find the corresponding value of α as the root of $G_A(y_A)$ using the algorithm

$$K_{n+1} = \begin{cases} K_n - (1-L)\rho_n & \text{if } |K_n - y_n| \le c_n,\ y_{A(n)} \text{ from } H_0 \\ K_n & \text{if } |K_n - y_n| > c_n \\ K_n + L\rho_n & \text{if } |K_n - y_n| \le c_n,\ y_{A(n)} \text{ from } H_1 \end{cases} \tag{3.31}$$

where

$$L = \frac{(\tilde{P}_F^{B1} - \tilde{P}_F^{B0})}{(\tilde{P}_F^{B1} - \tilde{P}_F^{B0}) + (\tilde{P}_D^{B1} - \tilde{P}_D^{B0})} \tag{3.32}$$

Once the algorithm has generated an approximate value of α, it is held fixed while the corresponding probabilities $\tilde{P}_F^A, \tilde{P}_D^A$ are estimated and communicated from A back to B. DM B then trains each of its thresholds β_0 and β_1 separately using

$$L_0 = \frac{(1 - \tilde{P}_F^A)}{(1 - \tilde{P}_F^A) + (1 - \tilde{P}_D^A)}, \qquad L_1 = \frac{\tilde{P}_F^A}{\tilde{P}_F^A + \tilde{P}_D^A} \tag{3.33}$$

respectively, and the iterations continue. The algorithm operates in a similar fashion for 3-Vee or any other tree [12] structure, with the DMs in the network communicating to resolve the coupling.

In spite of the fact that exact successive approximation converges for a given case, this algorithm comes with no guarantees of convergence. The reason is that the stochastic approximation subproblems converge to the proper thresholds only asymptotically, and in practice we must accept approximate solutions resulting from truncation. If these approximations become poor at some stage, the error propagates and the sequences of thresholds generated by the algorithm may no longer follow the sequences generated by the exact successive approximation solution. This still does not usually prevent convergence since the contraction property continues to pull the thresholds in the proper directions. It is also the case that stochastic approximation algorithms, by virtue of requiring no information on the statistics, tend to exhibit some variability in performance. We are also only estimating the coupling probabilities rather than computing them exactly. Nevertheless, in spite of all these caveats, a typical sequence of thresholds generated by the algorithm does display the same trends as the sequence generated by exact successive approximation, and in fact usually converges very near to the optimal solution. This behavior is illustrated in the next section.

4 Simulations

Due to space limitations, we present simulations only for the linear Gaussian case with parameters

$$\mu_0 = 1,\ \mu_1 = 3,\quad \sigma_A^2 = \sigma_B^2 = \sigma_C^2 = 1,\quad p1 = 0.25 \qquad (4.34)$$

Proper choice of the sequences ρ_n and c_n is critical to the performance of the algorithm. While theoretically the algorithm is guaranteed to converge for a wide range of these parameters, in practice many choices result in the convergence being impractically slow. Heuristics for choosing these sequences are discussed at length in [9]. For our simulations we used

$$\rho_n = \frac{1}{\sqrt{n}}, \qquad c_n = \frac{2.25}{\sqrt{n}} \tag{4.35}$$

as recommended in [15]. The algorithm is also sensitive to the initial starting conditions in practice, although again theoretically this is no problem. But as long as the starting conditions are reasonable, meaning that the algorithm is started in a region of sufficient probability density, it behaves well.

The simulations we present used 500 training examples for each subproblem, and each subproblem was rerun 15 times with independent noise on each pass and then averaged to smooth the curves. That is, each plotted threshold point represents an average over 15 points. In addition, at the end of each subproblem an additional 500 trials were used to effectively estimate the coupling probabilities. Thus, the total number of trials is given by

$$\begin{aligned} \#\mathit{trials} &= (\#\mathit{thresholds}) \cdot (\#\mathit{subproblems/threshold}) \\ &\cdot\ [(\#\mathit{trials/pass}) \cdot (\#\mathit{passes/subproblem}) \\ &+\ (\#\mathit{estimation\ trials/subproblem})] \end{aligned} \tag{4.36}$$

including the averaging. We have made no attempt to minimize the required computation to this point.

Fig.4 shows typical paths for thresholds α, β_0, and β_1 and the probability of error as a result of training 2-Tand. The paths for the exact successive approximation solution are shown along with the approximating paths resulting from the training algorithm. If stochastic approximation resulted in the subproblems being solved exactly, then the thresholds would be observed to converge exactly to the $\times$'s marking the values computed by the exact successive approximation.

The horizontal axis of each threshold graph shows the number of trials required by the subproblems, but in reality the total number of trials used in this simulation, computed as described above with the averaging, was $3 \cdot 10(500 \cdot 15 + 500) = 240,000$. From the point of view of our organizational learning studies, the averaging is somewhat arbitrary and is actually an implementation issue, so we prefer to take as an indicator of the number of trials for 2-Tand to learn the number $3 \cdot 10(500+500) = 30,000$. The integers on the horizontal axis of the probability of error graph mark completed cycles of updates of all three thresholds. The probability of error was computed after each such cycle of updates.

The spikes that appear on the graphs result from the fact that the initial steps of the window algorithm are large. Each spike marks the beginning of a new subproblem which is initiated after the DM receives the necessary coupling probabilities from the other DMs. The subproblems for each threshold were always initialized to the same numerical value. There is nothing to be gained by initializing the algorithm to the end of the previous subproblem since the algorithm requires only a region of sufficient probability to be effective. For these simulations, the window algorithm was always reinitialized to 2 for α, 0 for β_0, and 2 for $\beta 1$. These were simply arbitrary choices which appeared to result in good performance.

In order that the paths of the exact successive approximation and the training algorithm be comparable, both used the same initial values of $\beta_0 = 2.5$ and $\beta_1 = 1.5$. These initial guesses were reasonable choices given the problem, but convergence occurs for a wide variety of initial choices.

It is clear from the figures that the approximate solutions do exhibit the same trends as the exact solutions. Moreover, when the approximation is off of the exact successive approximation path, it is generally in the process of moving toward it and is simply being cut short. But more importantly, the rate of convergence of the approximations is comparable to the exact computations, and the final values are very close. We note that the simulations shown were chosen because they were typical rather than because they were particularly good.

Fig.5 shows the paths travelled by the thresholds of 3-Vee, α, β, ξ_{00}, ξ_{01}, ξ_{10}, ξ_{11} and the probability of error. The total number of trials to train this network was $6 \cdot 15(500 \cdot 15 + 500) = 720,000$. Initial values for the window algorithm were taken to be 2 for all thresholds. So that the paths of the exact successive approximation and the training would be comparable, both used the same initial values of $\beta = 0$,

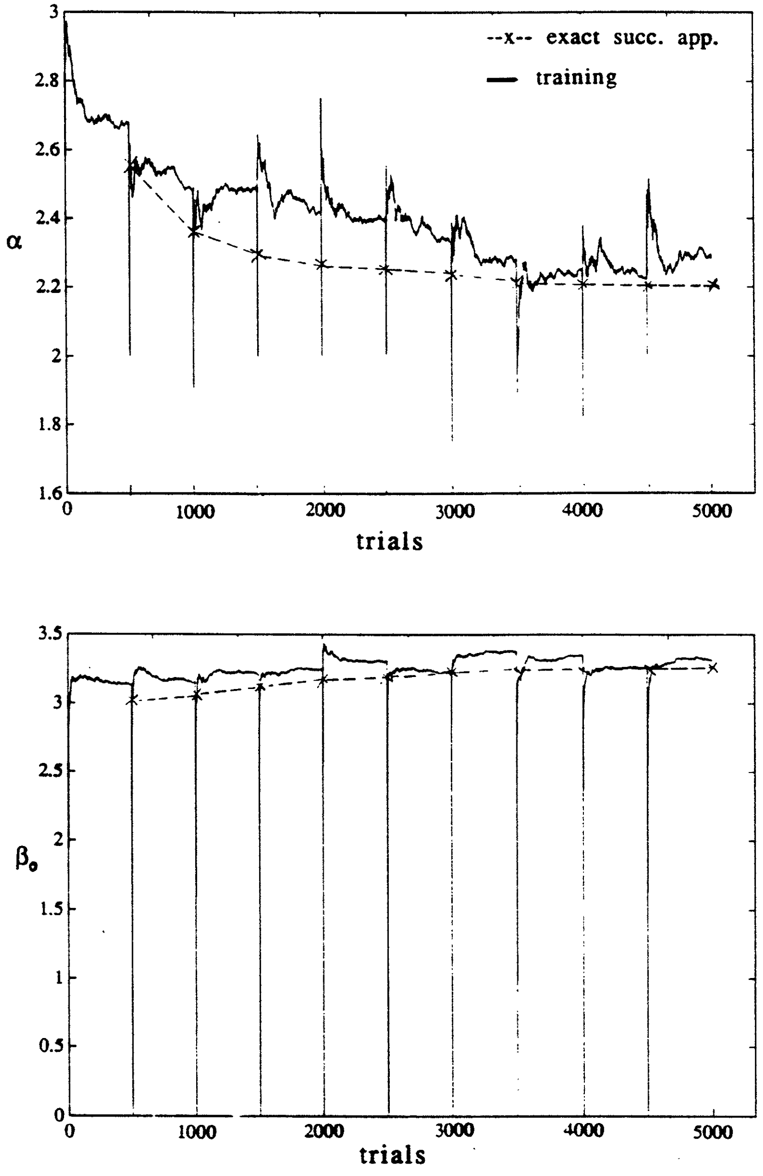


Figure 4: Motion of 2-Tand Thresholds during Training

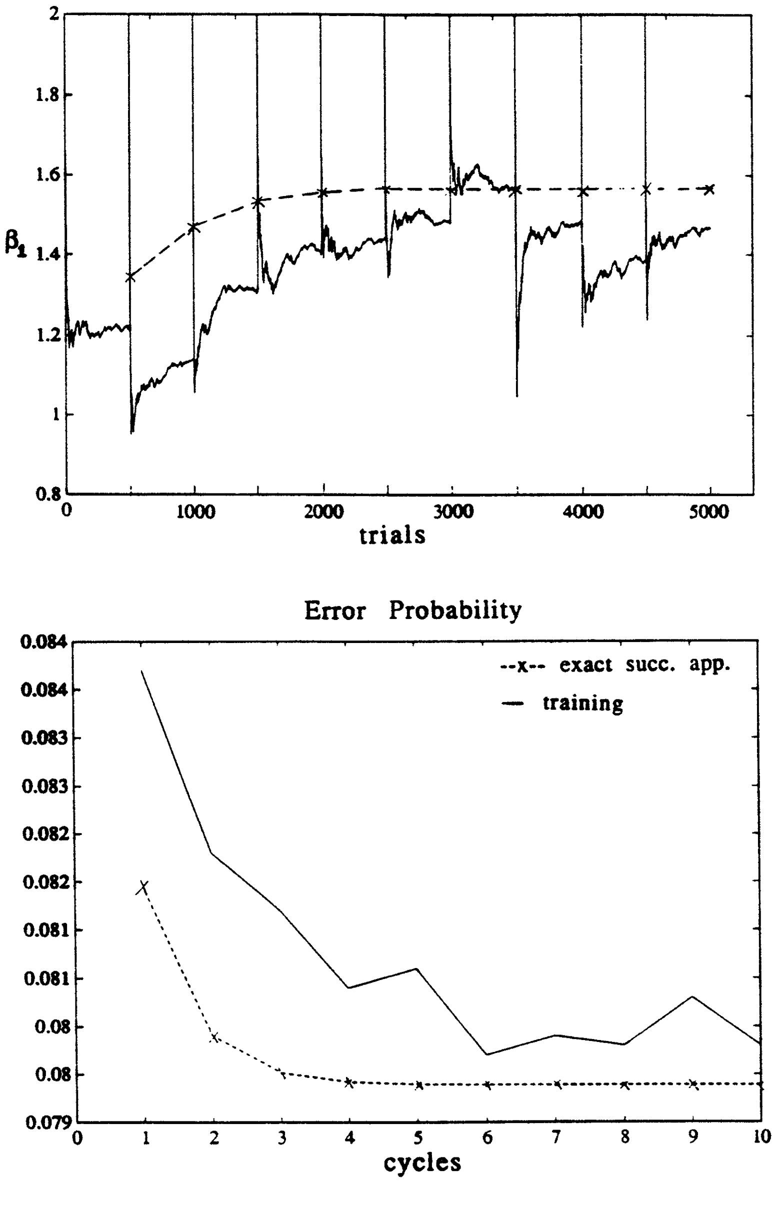


Figure4: continued

$\xi_{00} = 0$, $\xi_{01} = 1$, $\xi_{10} = 2$, $\xi_{11} = 3$.

5 Discussion

The training algorithm we have presented is successive approximation using stochastic approximations instead of function evaluations to solve for each DM's threshold(s) in terms of the others in the network. The amount of information required by the algorithm is minimal, but the price for this is that a great many trials appear to be required to solve the problems effectively, although we have in no way attempted to improve the algorithm in this or any other regard. The total number of trials required is a function of several things. The number of trials required to solve a given subproblem depends on the noisiness of the corresponding DM's observations, while the number of subproblems which must be solved is highly dependent on the degree of coupling between the DMs and the overall size of the network. It is also clear that the algorithm requires the network Likelihood Ratio Tests in order to structure the computation of the coupling probabilities. This is equivalent to saying that each DM must know how it is tied in structurally to the organization, but it can be initially naive to the capabilities of the other DMs since it can infer them during training.

We wish to point out that we have investigated less elaborate algorithms than the one in this paper for the training of decentralized binary detection networks. Examples of some of these more "natural" approaches are approximate gradient descent on the probability of error surface and back propagation based on the optimal control formulation from [10]. However, we believe this training algorithm is of interest because it represents an indirect method for solving the underlying optimization and it is completely nonparametric in the statistics.

An understanding of the processes by which human organizations adapt to improve performance has been slow in coming, primarily because there is a notable lack of normative theory which addresses the inherent difficulties and fun-

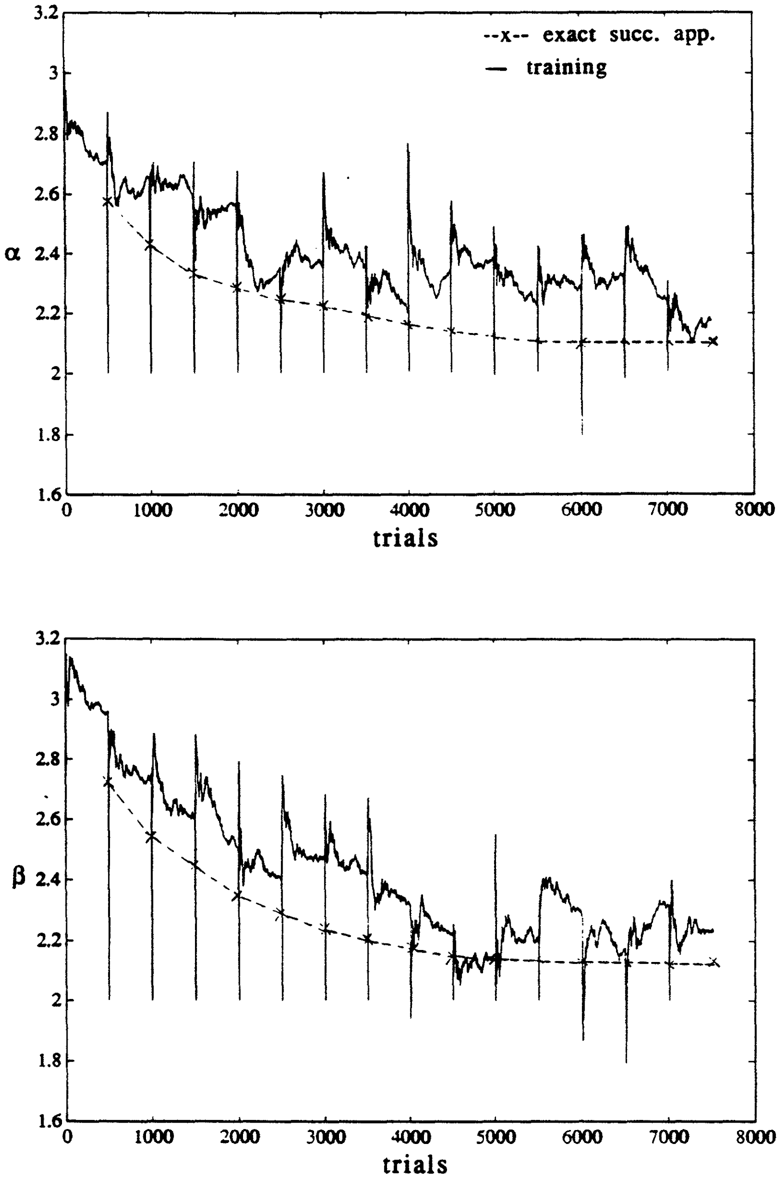
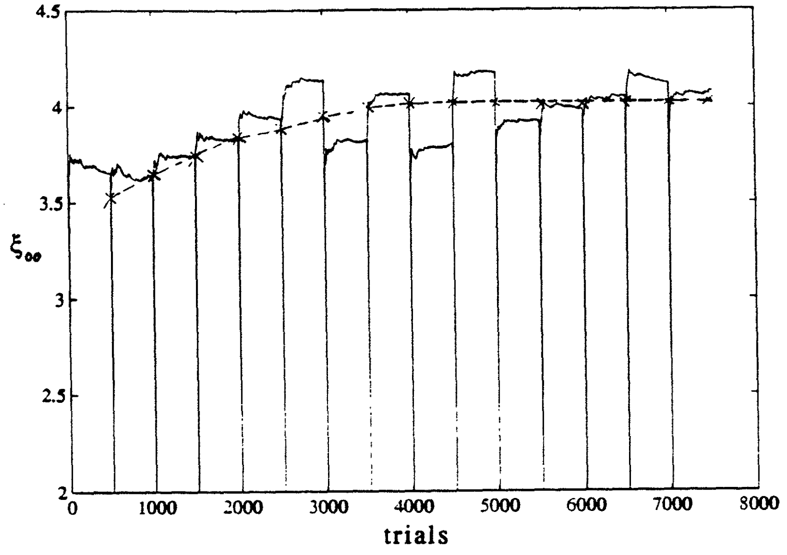


Figure 5: Motion of 3-Vee Thresholds during Training



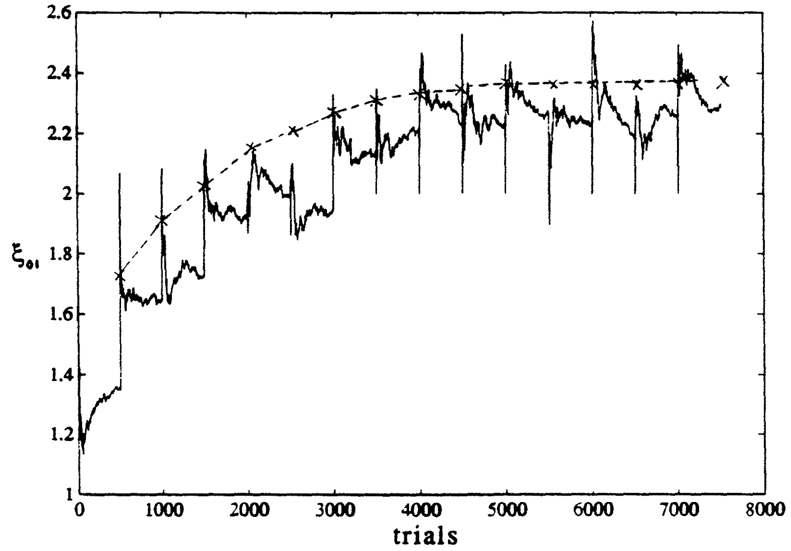


Figure 5: continued

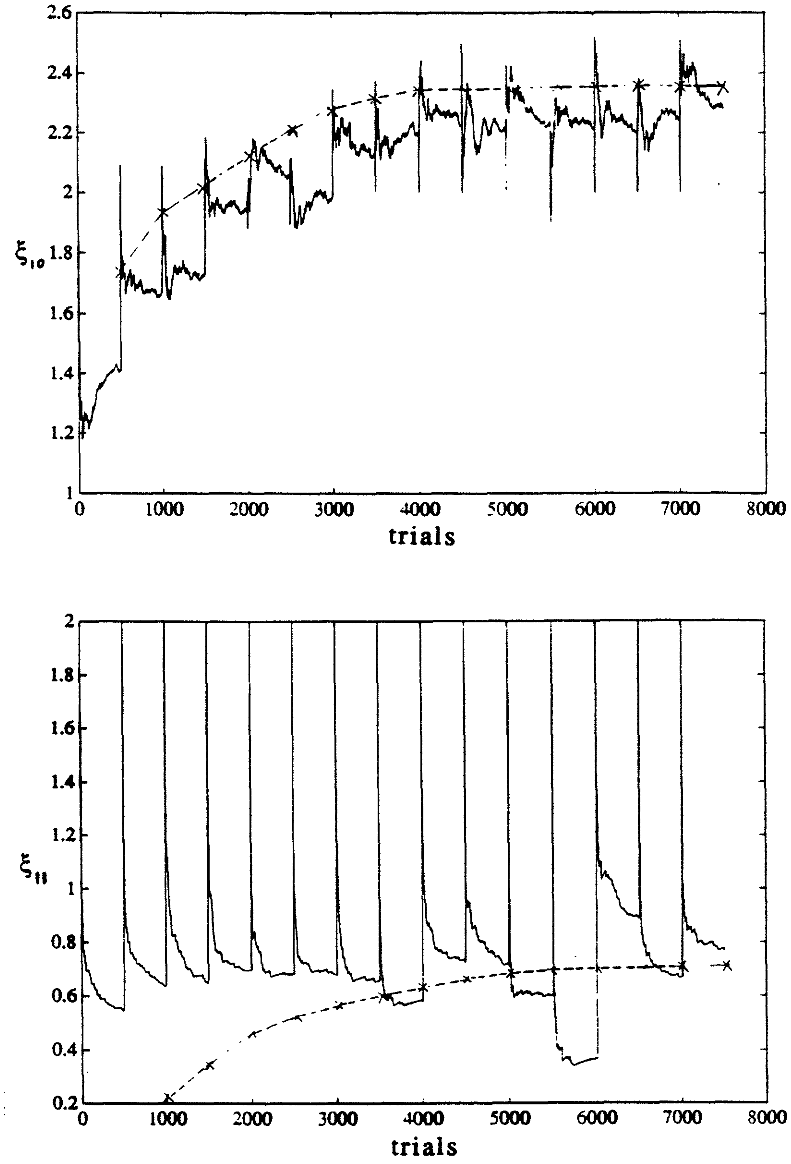


Figure 5: continued

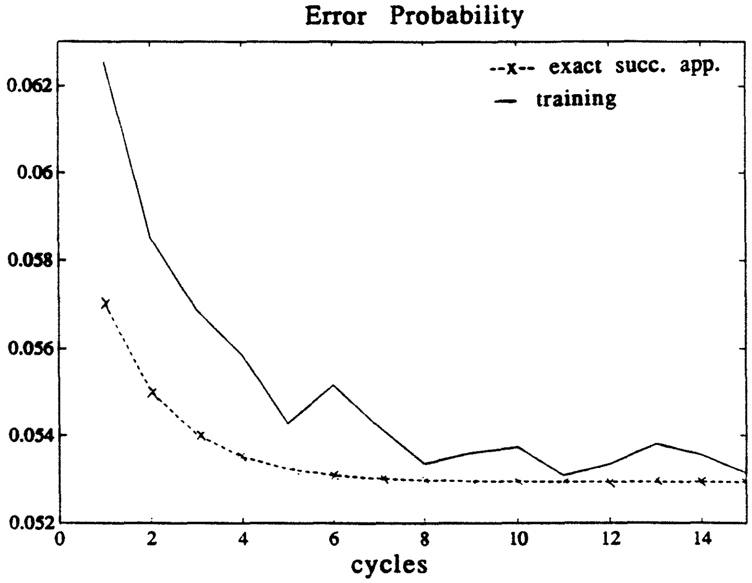


Figure 5: continued

damental limitations of learning in distributed environments. Enhancing this understanding is a long term goal of our research.

References

[1] Bertsekas, D.P. and J.N. Tsitsiklis. *Parallel and Distributed Computation.* Prentice Hall, 1989.

[2] Boettcher, K.L. and R.R. Tenney. Distributed Decisionmaking with Constrained Decisionmakers: A Case Study. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-16(6):813–822, 1986.

[3] Do-Tu, H. and M. Installe. Learning Algorithms for Nonparametric Solution to the Minimum Error Classification Problem. *IEEE Trans. on Computers*, C-27(7):648–659, 1978.

[4] Ekchian, L.K. *Optimal Design of Distributed Detection Networks.* PhD thesis, M.I.T., 1982.

[5] Kac, M. A Note on Learning Signal Detection. *IRE Trans. on Information Theory*, pages 126–128, Feb. 1962.

[6] Papastavrou, J. *Decentralized Decision Making in a Hypothesis Testing Environment.* PhD thesis, M.I.T., 1990.

[7] Pothiawala, J. Analysis of a Two-Sensor Tandem Distributed Detection Network. Master's thesis, M.I.T., 1989.

[8] Robbins, H. and S. Monro. A Stochastic Approximation Method. *Ann. Mathematical Statistics*, 22:400–407, 1951.

[9] Sklansky, J. and G. Wassel. *Pattern Classifiers and Trainable Machines.* Springer-Verlag, 1981.

[10] Tang, Z. *Optimization of Detection Networks.* PhD thesis, Univ. of Connecticut, 1990.

[11] Tenney, R.R. and N.R. Sandell, Jr. Detection with Distributed Sensors. *IEEE Trans. on Aerospace and Electronic Systems*, AES-17(4):501–510, 1981.

[12] Tsitsiklis, J.N. Decentralized Detection. In *Advances in Statistical Signal Processing, vol.2: Signal Detection.* , to appear.

[13] Tsitsiklis, J.N. and M. Athans. On the Complexity of Decentralized Decision Making and Detection Problems. *IEEE Trans. on Automatic Control*, AC-30(5):440–446, 1985.

[14] Van Trees, H.L. *Detection, Estimation, and Modulation Theory*, volume 1. J. Wiley, New York, 1968.

[15] Wassel, G. and J. Sklansky. Training a One-Dimensional Classifier to Minimize the Probability of Error. *IEEE Trans. on Systems, Man, and Cybernetics*, SMC-2(4), Sept. 1972.

[16] Wassel, G.N. *Training a Linear Classifier to Optimize the Error Probability.* PhD thesis, Univ. of California, Irvine, 1972.