The Covering Problem and μ-Dependent Adaptive Algorithms

James A. Bucklew, *Member, IEEE,* and William A. Sethares, *Member, IEEE*

Abstract— **This paper presents a family of techniques, called *Adaptive Covering Algorithms*, which solve a particular covering problem—how to best cover a target shape using a set of simply parameterized elements. The algorithms, inspired by adaptive filtering techniques, provide a computationally simple, robust, and efficient alternative to more traditional methods such as Bayesian approaches, convex hulls, and multi-layer perceptrons. The paper develops a theoretical understanding of the adaptive covering algorithms by relating their behavior via weak convergence techniques to the evolution of a deterministic ordinary differential equation (ODE). In the process, we give new convergence results for a class of step size-dependent recursive algorithms. Stability and instability of the ODE can be interpreted in terms of local stability/instability of the algorithm. In terms of the covering problem, candidate coverings tend to improve as more data is gathered whenever the ODE is stable. Several examples are given which demonstrate the ideas and which verify that the analysis accurately predicts the true behavior of the algorithms.**

I. Introduction

MANY problems in image analysis, data compression, automatic classification, and pattern recognition can be stated succinctly in terms of the covering problem:

Given a set of parameterized shapes (such as rectangles, ellipses, polygons, half planes), how can a target region (or family of target regions) be best covered by these shapes?

Of primary interest are algorithms which automatically learn the target region. Algorithms which are easily implemented, computationally efficient, and robust to noise and misclassification errors are preferred. This paper presents a family of such algorithms whose first members were introduced in [16], and which are variants of known adaptive filtering methods [15]. We call this family of techniques adaptive covering algorithms (ACAs).

A parameter (or weight) vector $W_k \in \Re^n$ is used to concisely describe the best current guess at time k of the target region. An iterative method of the form

$$W_{k+1} = W_k + \mu \{correction\ term\} \tag{1}$$

is employed to improve this guess, where the *correction term* is some simple function of the data available at time k, and μ is a stepsize that determines the impact of the new data on the current estimate. A good choice of the correction term will cause the parameter vector sequence $\{W_k\}$ to improve with time on average. In certain cases, an analytical technique in the spirit of [2], [3], [8], and [9] can be used to provide concrete information about the behavior of the algorithm. This technique relates the stochastic behavior of the algorithm to the behavior of a deterministic ordinary differential equation (ODE). When the ODE is stable, the algorithm will tend to converge to a region about its minimum, and this convergence can be characterized in terms of a steady state error distribution. When the ODE is unstable, the algorithm is unstable.

In terms of the covering problem, these analyses demonstrate that when the ODE is stable, the covering of the target area will improve, on average, over the course of time. Moreover, if the covering is initialized "near" the optimal covering, then the algorithm is guaranteed to converge (distributionally) to a small ball about that optimal covering. When the ODE is unstable, the covering is unlikely to be useful. Many nonlinear algorithms are susceptible to local minimum problems. We have found, however, that the best of the ACA's tend to behave rather well in the sense that they converge to a good covering, in reasonable time, even if it is not optimal.

Much of the robustness of the ACA approach arises because the adaptation of the parameterized shapes is smoothed by a user-chosen stepsize. Larger stepsizes cause the algorithm to converge faster, but result in more jitter in the event of significant noise (such as errors in the sample points or in the indicators). Small stepsizes imply slower convergence, but noises are more effectively smoothed. This sort of behavior is generic in the study of adaptive algorithms.

There are, of course, many ways of attacking this problem. Algorithms which address the problem of learning decision regions typically fall into two categories, "convex hull" (and other clustering) approaches, and Bayesian (or other statistical) methods. The convex hull approaches are computationally expensive and are sensitive to noise or errors in the training set, while the Bayesian methods are sensitive to deviations from the assumptions, which typically require detailed knowledge of the statistical properties of the relevant data. We will not detail either of these methods.

The present approach is closest to [6] (which is also an outgrowth of [16]), where a class of online learning algorithms are analyzed deterministically via Lyapunov and averaging methods. These algorithms incorporate sigmoidal functions in their updates, making their implementation similar to the neural networks of [1], which solved for a class of nonlinearly parameterized regions using examples and queries. The learning algorithm of [6] and the adaptive covering approach can both solve such problems using examples only.

Manuscript received July 29, 1992; revised January 5, 1994. The associate editor coordinating the review of this paper and approving it for publication was Dr. H. Fan.

The authors are with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53706-1691 USA.

IEEE Log Number 9403738.

Another related method is that of [13], where a standard perceptron is used with an augmented input vector to identify regions of various shapes. The advantage of our explicit parameterization is that it allows one to easily incorporate prior information about target shape or shapes directly into the algorithm structure. This serves the purpose of reducing the dimensionality of the parameter space that must be searched in order to find a good estimate, and finesses a difficulty encountered in perceptron-type algorithms of requiring (for solvability) a large augmented input vector [11].

ACA's are only superficially related to the template matching morphological approach of [14] in which fixed structuring elements (or templates) are moved in a predetermined way across an image. ACA's, in contrast, use parameterized regions that change size, multiple templates operate simultaneously, and their motion across the image is determined by the image itself.

Section II presents and motivates some candidate covering algorithms. Section III presents some new theoretical results using weak convergence techniques developed for the study of adaptive algorithms. Section IV applies the theory to several specific examples, which allows us to draw conclusions regarding the suitability of certain of the algorithmic variants. Section V presents simulations which graphically illustrate the behavior of the algorithms as the primitive shapes migrate to cover the target areas. Section VI is devoted to conclusions and possible future work.

II. Candidate Algorithm Generation

The first method employed to generate algorithms was to modify some well known adaptive filtering algorithms to work on the covering problem. The most famous of all such algorithms is the least mean square (LMS) algorithm. For example, let W_k be a set of weights in a linear filter and denote the output as

$$y_k = W_k^* X_k \tag{2}$$

where X_k is an input vector and $*$ denotes transpose. The goal is to adjust the parameters W_k so that the output of the linear filter (2) matches d_k as closely as possible. The LMS algorithm [15] is

$$W_{k+1} = W_k + \mu X_k (d_k - y_k). \tag{3}$$

Numerous variants are possible. For instance, the "signed error" algorithm [5]

$$W_{k+1} = W_k + \mu X_k \text{sgn}(d_k - y_k) \tag{4}$$

and the "sign-sign" LMS algorithm [10]

$$W_{k+1} = W_k + \mu \text{sgn}(X_k) \text{sgn}(d_k - y_k) \tag{5}$$

are popular choices, where sgn of a vector is taken to be an element-by-element operation. Two of the algorithms studied in Section IV are close analogs of (4) and (5).

LMS and its variants are frequently interpreted as being modified gradient descent algorithms. Given an estimated set of parameters W_k at time k, a "cost" is defined via the function $J(W_k)$. The key idea of this technique is to make improved estimates via the iteration

$$W_{k+1} = W_k - \mu \nabla J \tag{6}$$

where the gradient ∇J is taken with respect to W_k. If the cost function is (locally) convex, and the stepsize is small, then the cost $J(W_k)$ is a nonincreasing function of time, that is, the parameter estimate W_{k+1} is never worse than the estimate W_k. The particular form of the update term depends heavily on the function J and its dependency on W. Gradient descent strategies can be exploited in the covering problem in a fairly straightforward fashion.

The stochastic approximations approach is similar, except that the stepsize μ is replaced by a time varying stepsize that gradually converges to zero; $\mu_k = 1/k$ is a common choice. The decreasing μ_k is attractive because one can often guarantee that the algorithm converges. With fixed μ algorithms, this rarely occurs. Once the algorithm has "converged" it tends to jitter around some averaged equilibrium. It has been our experience that the fixed μ case (for small μ) tends to be more useful because it does not "shut off" for large k. The fixed μ algorithms tend to zoom into an equilibrium and wiggle about near it. The decreasing μ_k algorithms tend to take orders of magnitude longer to get within range of the equilibrium. Hence we have concentrated most of our attention on fixed μ structures.

We now apply these gradient descent notions to the covering problem. For any set $A \in \Re^r$, let $I_A(\cdot): \Re^r \to \{0,1\}$ denote the indicator function of the set A. Suppose there are n parameterized "shape" or kernel functions $K_{\mathbf{a}^i}(\cdot)$: $\Re^r \to \Re^1$ $i = 1, \ldots, n$ where $\mathbf{a}^i \in \Re^m$ is the i^{th} parameter vector $\mathbf{a}^i = (a_1^i, a_2^i, \ldots, a_m^i)$. A typical example (for $r = 2$) is $K_{\mathbf{a}}(\cdot) = I_{R(s,d)}(\cdot)$ where $R(s,d)$ is the interior of a rectangle with center $s = (s_1, s_2)$ and sidelengths $d = (d_1, d_2)$. Let X denote an r-dimensional random variable distributed over a region that includes the target area. Usually X is taken to be uniform. This random variable may be thought of as the "sampling" random variable. Consider an L_2 or mean squared error objective function:

$$J(\mathbf{a}^1, \mathbf{a}^2, \ldots, \mathbf{a}^n) = E\{|I_T(X) - \sum_{i=1}^{n} K_{\mathbf{a}^i}(X)|^2\}. \tag{7}$$

For example, if $K_{\mathbf{a}}(\cdot) = I_{R(s,d)}(\cdot)$, the argument of the expectation is 0 at a point $x \in T$ exactly when a single box covers that point. In general, the argument is $(k-1)^2$ if k boxes are covering. Thus there is an impetus to cover, but also a counterbalancing tendency to "spread out" over the target area. Similar arguments usually hold for other choices of kernel function.

Using the cost function (7) and the gradient strategy (6) leads to the algorithm

$$(\mathbf{a}_{\mathbf{k+1}}^{\mathbf{1}}, \mathbf{a}_{\mathbf{k+1}}^{\mathbf{2}}, \ldots, \mathbf{a}_{\mathbf{k+1}}^{\mathbf{n}}) = (\mathbf{a}_{\mathbf{k}}^{\mathbf{1}}, \mathbf{a}_{\mathbf{k}}^{\mathbf{2}}, \ldots, \mathbf{a}_{\mathbf{k}}^{\mathbf{n}}) - \mu V(X) \tag{8}$$

where $V(x) = \nabla |I_T(x) - \sum_{i=1}^{n} K_{\mathbf{a}^i}(x)|^2$ and the gradient is taken with respect to the parameters $(\mathbf{a}^{\mathbf{1}}, \mathbf{a}^{\mathbf{2}}, \ldots, \mathbf{a}^{\mathbf{n}})$.

In some cases, as when $K_{\mathbf{a}}(\cdot) = I_{R(s,d)}(\cdot)$, the differentiation operation needed in the definition of $V(\cdot)$ is impossible.

Even though $V(\cdot)$ does not exist in functional form, $J(\cdot)$ might still be differentiable. In this case, one possible approach is to numerically differentiate $J(\cdot)$, and use the resulting calculation in (8). This is reminiscent of the Kieffer–Wolfowitz algorithm [7] of stochastic approximations. The numerical differentiation sometimes causes problems of its own, usually pertaining to poor noise immunity and resulting slow convergence rates. We investigate an algorithm that incorporates numerical differentiation in Section IV.

Another alternative is to make sure that the kernel functions are differentiable. One possibility is to choose a kernel function $K_{\mathbf{a}}$ with "smooth" edges. For example, we might choose $\mathbf{a} = (s, d)$ and

$$K_{\mathbf{a}}(x) = \exp(-[x - s]^* D[x - s]) \tag{9}$$

where D is the diagonal matrix whose nonzero entries are the d vector. $V(\cdot)$ is differentiable and (8) can be implemented directly. We investigate this algorithm in Section IV.

The Gaussian kernel of (9) is, of course, only one of many sensible choices. Butterworth approximations to boxes, and other kernels from filtering theory, quickly come to mind. We find that the convergence properties of the algorithms are heavily dependent upon the nature of the kernel functions (as well as target shape). The design of a good algorithm for a particular application must take into account all of these factors.

Another way to create more candidate algorithms is to change the functional form of $J(\cdot)$. Instead of the L^2 error, $E\{|I_T(X) - \sum_{i=1}^n K_{\mathbf{a}^i}(X)|^2\}$, one might use the L^1 error, $E\{|I_T(X) - \sum_{i=1}^n K_{\mathbf{a}^i}(X)|\}$. The former leads to algorithms which might be thought of as analogs of LMS, while the latter leads to "signed" style updates.

Given this large body of potential algorithms, how can an intelligent choice be made? The next section presents a methodology that has been successful in analyzing and comparing various adaptive filtering algorithms. We then give examples of this methodology applied to the specifics of the covering problem.

III. Local Stability and Weak Convergence Analysis

The basis of the analytical approach is to find an ordinary differential equation (ODE) that accurately mimics the behavior of the algorithm for small values of μ. Studying the ODE then gives valuable information regarding the behavior of the algorithm. For example, if the ODE is stable, then the algorithm is convergent (at least in distribution). If the ODE is unstable, then the algorithm is divergent. In addition, looking at the eigenvalues of the linearized portion of the algorithm (which can be accomplished even for highly nonlinear algorithms if there is sufficient smoothing imparted by the input and noise processes) gives a local rate of convergence for the algorithm. To be more specific, consider an ACA as a discrete time iteration process

$$W_{k+1} = W_k + \mu G(W_k, Y_k, U_{k+1}, \mu) \tag{10}$$

where W_k is the parameter vector of weights that define the primitive shapes, μ is the stepsize, U_k is an input vector that usually consists of the new sample point x_k, and Y_k represents errors in the samples x_k, in the evaluation of $I_T(x_k)$, computation errors, or other disturbances. The function $G(\cdot, \cdot, \cdot, \cdot)$ represents the update term of the algorithm, and is in general discontinuous for ACA's. In implementation, one typically considers both the sample point x_k and its indicator $I_T(x_k)$ to be inputs. It is more convenient analytically to suppose that only x_k is input, and that G then calculates $I_T(x_k)$. A related, but somewhat simpler model than (10) (without the dependence of G on μ) is considered extensively in [3] and in the book by Beneveniste, *et al.* [2]. The model

$$W_{k+1} = W_k + \mu G(W_k, Y_k, U_{k+1}) + \mu^2 H(W_k, Y_k, U_{k+1})$$

studied in [2] may be viewed as a special case of the μ-dependent algorithm. This form can be generalized to (10) when $G(\cdot, \cdot, \cdot, \cdot)$ is differentiable with respect to μ. The μ-dependent algorithms considered in this paper are usually not differentiable. Kushner [8] studies the general μ-dependent case but uses different methods and technical conditions from ours. Our theorems are different from theirs, following the techniques and methods of [3].

What is the nature of the random process $\{W_k\}$? When is this process stable? How can its convergence to equilibria be characterized? These questions can be addressed by relating the behavior of the algorithm (10) for small μ to the behavior of the associated deterministic ordinary differential equation (ODE)

$$W(t) = W_0 + \int_0^t \overline{G}(W(s))ds \tag{11}$$

where $\overline{G}(\cdot)$ is a version of $G(\cdot, \cdot, \cdot, \cdot)$ that is smoothed, or averaged, over the inputs and the disturbances. Speaking loosely, the ODE $W(t)$ in (11) represents the "averaged" behavior of the parameters W_k in (10).

Suppose that (W_k, Y_k, U_k) is adapted to the filtration $\{\mathcal{F}_k\}$. Assume also that U_{k+1} given (W_k, Y_k) is independent of U_k. (See assumption A4 and remark 3 below for discussion of this assumption.) Define

$$\hat{G}(W_k, Y_k, \mu) = E[G(W_k, Y_k, U_{k+1}, \mu)|\mathcal{F}_k]$$

to be a version of G that is smoothed by the distribution of the inputs U_{k+1}. This smoothed version is often differentiable even if G itself is discontinuous. In the following, several versions of the update term will be defined. G is the update term in the algorithm (10) and $\hat{G}$ is a version of G that is smoothed over the distribution of the inputs. $\tilde{G}$ is the limit of the $\hat{G}$'s with respect to μ, while $\overline{G}$ of (11) is a version of $\tilde{G}$ smoothed further over the distribution of the disturbances. A time scaled version of $\{W_k\}$ is defined as

$$W_\mu(t) = W_{[t/\mu]}, \ t \in [0, \infty)$$

where $[z]$ means the integer part of z. Note that W_k (with the Roman subscript) represents the discrete iteration process, while $W_\mu(t)$ (with the Greek subscript) represents a continuous time version. $W(t)$ (with no subscript) is the ODE (11) to which the others converge.

Let (E, r) denote a metric space with associated Borel field $\mathcal{B}(E)$ and let $D_E[0,\infty)$ be the space of right continuous functions with left limits mapping from the interval $[0,\infty)$ into E. Let $C_E[0,\infty)$ denote the subspace of continuous functions. We assume that $D_E[0,\infty)$ is endowed with the Skorohod topology (see Appendix for definition).

Let $\{X_\alpha\}$ (where α ranges over some index set) be a family of stochastic processes with sample paths in $D_E[0,\infty)$ and let $\{P_\alpha\} \subset \mathcal{P}(D_E[0,\infty))$ be the family of associated probability distributions (i.e., $P_\alpha(B) = P\{X_\alpha \in B\}$ for all $B \in \mathcal{B}(E)$). We say that $\{X_\alpha\}$ is relatively compact if $\{P_\alpha\}$ is relatively compact in the space of probability measures $\mathcal{P}(D_E[0,\infty))$ endowed with the topology of weak convergence (see Appendix for definition). (A set is said to be relatively compact if every sequence contained in the set has a convergent subsequence.) The symbol $\Rightarrow$ will denote weak convergence, while the arrow $\rightarrow$ will denote convergence under the appropriate metric. An excellent reference for all the mathematical terms and probabilistic constructs used in this section and in the Appendix is [4].

Consider the following technical assumptions:

A1) $\{\hat{G}(W_k, Y_k, \mu) : k \in \mathcal{Z}^+, \mu > 0\}$ is uniformly integrable.

A2)

$$\mu^2 \sum_{k=1}^{[t/\mu]} E[(G(W_k, Y_k, U_{k+1}, \mu) - \hat{G}(W_k, Y_k, \mu))^2] \rightarrow 0.$$

A3) $W_0 = W_\mu(0) \rightarrow w_0 \in \Re^r$ in probability.

A4) $\{Y_k\}$ is a stationary ergodic sequence of E valued random variables. Furthermore, there exists a measurable function $q(\cdot,\cdot,\cdot)$ such that $U_{k+1} = q(W_k, Y_k, \psi_k)$ where the $\{\psi_k\}$ are i.i.d. E_ψ-valued random variables.

A5) $\hat{G}(w, y, \mu)$ converges uniformly on $\Re^r \times E$ to a bounded continuous function $\tilde{G}(w, y)$.

Theorem 1: Under A1-A5, $\{W_\mu\}$ is relatively compact and every possible limit point is a random process taking values in $C[0,\infty)$. Furthermore, every limit point of $\{W_\mu\}$ satisfies (11).

All proofs are relegated to the Appendix, since they are technically involved.

The theorem asserts that the ACA's (10) will behave like the ODE (11) for small enough μ. If the solution to the ODE is unique, then the sequence actually converges in probability (not just has a weakly convergent subsequence). The solution of the ODE is, of course, continuous. The Skorohod topology for continuous functions corresponds exactly to uniform convergence on bounded time intervals. Hence, convergence in probability means that for every $T > 0, \epsilon > 0$, $\lim_{\mu\rightarrow 0} P(\sup_{0\le t\le T} |W_\mu(t) - W(t)| > \epsilon) = 0$. This is useful because the ODE can often be analyzed in a straightforward manner, otherwise it can be numerically integrated. The advantage of calculating the ODE over directly simulating the algorithm is that the behavior of the algorithm can vary widely in the short term, depending on the vagaries of the disturbances, the sampling methods used, the input, the target area, etc., while the ODE is fully deterministic.

Remarks

1) If desired, $W_\mu(t)$ can be "stopped" (and held constant) if it wanders outside of some predetermined set A (usually chosen to be compact). A similar theorem for the sequence of stopped processes can then be shown. If A is compact, then the assumption of boundedness in A5 can be removed (since the stopped process will always be bounded anyway). If the limiting ODE has a unique solution which does not blow up in finite time, we can then let the compact set A enlarge to the whole space and have a limit theorem for the "unstopped" process. See [3] for details. Working with stopped processes, A1 can be replaced by
A1′) $\{\sup_{w\in A} \hat{G}(w, Y_k, \mu): k \in \mathcal{Z}^+, \mu > 0\}$ is uniformly integrable.
2) The first part of assumption A4 can be changed to
A4′) $\{Y_k\}$ is asymptotically ergodic. Furthermore, there exists a measurable function $q(\cdot,\cdot,\cdot)$ such that $U_{k+1} = q(W_k, Y_k, \psi_k)$ where the $\{\psi_k\}$ are i.i.d. E_ψ-valued random variables.
When A4′ holds, we must redefine $\overline{G}(w) = \lim_{n\rightarrow\infty} \frac{1}{n} \sum_{k=1}^{n} \hat{G}(w, Y_k)$.
3) The assumed i.i.d. nature of the underlying noise process $\{\psi_k\}$ in the second part of A4 may at first glance seem stringent. It is really more of a convenience to simplify notation than any fundamental limitation. We could instead assume that $U_{k+1} = q(W_k, Y_k, \psi_k)$ and $\{(Y_k, \psi_k)\}$ is a jointly ergodic sequence of $E \times E_\psi$ valued random variables. In that case, then $\hat{G} = \hat{G}(W_k, Y_k, \psi_k, \mu)$, i.e., we have a ψ_k dependence now in the $\hat{G}$ function. This causes no difficulties and the proofs and theorems proceed identically.

A1-A2 are sometimes onerous conditions to check. They depend on the $\{W_k\}$ values themselves, which we are trying to obtain information about in the first place. Therefore, we may wish to replace A1-A2 with the following condition A6 and obtain the first corollary.

A6) $E[\sup_{w\in A, \mu} |G(w, Y_k, U_{k+1}, \mu)|] < \infty$ and $E[\sup_{w\in A, \mu} |\hat{G}(w, Y_k, \mu)|] < \infty$. In addition, suppose $U_{k+1} = q(W_k, Y_k, \psi_k)$, where (Y_k, ψ_k) is jointly ergodic.

Corollary 1: Suppose A3–A6. Then the conclusions of the previous theorem hold.

One case of particular interest is where G has no dependence on a $\{Y_k\}$ process. The following corollary asserts the results of the theorem still hold but under a milder condition. We define a new assumption:

A5′) $\hat{G}(w, \mu)$ converges to $\overline{G}(w)$, a continuous function for all $w \in A$. Furthermore, for sufficiently small μ, $\sup_{w\in A} \hat{G}(w, \mu) \le B < \infty$, for some B not dependent upon μ.

Corollary 2: Suppose the algorithm form is

$$W_{k+1} = W_k + \mu G(W_k, U_{k+1}, \mu) \tag{12}$$

where $\{\mathcal{F}_k\}$, $\hat{G}$ and W_μ are defined as before. Assume one of the two sets of assumptions ($A1'$ (or A1), A2, A3, A4), or (A3, A4, A6), along with A5′. Then the conclusions of the previous theorem hold.

IV. Examples and Applications

This section applies the theory of the previous section to investigate the behavior of several ACA's. The first example is a simple 1-D μ-dependent algorithm. Its simplicity allows a complete closed form analysis. We then analyze the performance of some more complex (and more useful) ACA's.

For simplicity, these analyses suppose that a target rectangle T in the unit square [0, 1] $\times$ [0, 1] is to be covered by two kernel functions. We fix the shape of the kernel functions and adapt only the location parameters. Thus, the parameter vector is a 4-tuple in all of the analyses. Then, by evaluating the G, $\hat{G}$, and $\overline{G}$ functions, (the update and the smoothed versions), the ODE (11) can be readily determined. The procedure is straightforward, although the calculations sometimes are tedious.

Stable ODE's correspond to well-behaved algorithms while unstable ODE's correspond to algorithms which will fail. In a practical sense, stability implies that the algorithms will be robust to noise, misclassification error, and (most importantly) to target areas that do not exactly match the shape of the primitive figures. For instance, if the target area is a circle but the weights are parameterized to represent a square, then the figure cannot exactly cover the target. Stability of the ODE suggests that the square will center itself on the circle and adjust its sidelength so as to tradeoff the target area uncovered with the nontarget area covered. This is indeed the observed behavior of the successful algorithms.

A. A 1–D Example

In order to introduce the techniques, we first present the analysis of a very simple "μ-dependent" algorithm. This example considers a unit line segment $[w - 1/2, w + 1/2]$ seeking to automatically identify the target region $[-1/2, 1/2]$. The algorithm uses a Kiefer–Wolfowitz style update which numerically approximates the derivative of the cost function (7). The input to the algorithm consists only of the sample points $\{X_k\}$. The algorithm (8) becomes

$$W_{k+1} = W_k + \gamma\mu[I_T(X_k) - 1] \cdot [K_{W_k+\sqrt{\mu}}(X_k) - K_{W_k-\sqrt{\mu}}(X_k)]/\sqrt{\mu}$$

where $K_z(\cdot) = I_{[z-1/2, z+1/2]}(\cdot)$ is the indicator of the segment $[z - 1/2, z + 1/2]$. Defining $U_{k+1} = X_k$, we may rewrite the above as

$$\begin{aligned} W_{k+1} &= W_k + \gamma\mu[I_T(U_{k+1}) - 1] \\ &\quad \times [K_{W_k+\sqrt{\mu}}(U_{k+1}) - K_{W_k-\sqrt{\mu}}(U_{k+1})]/\sqrt{\mu} \\ &= W_k + \mu G(W_k, U_{k+1}, \mu). \end{aligned}$$

This is of the form of (12) of Corollary 2 where $\gamma > 0$ is some fixed parameter $U_{k+1} = \psi_k + W_k$, and the $\{\psi_k\}$ are i.i.d. uniformly distributed $[-1/2, 1/2]$ random variables. Fix some $\delta > 0$ and assume that $W(0) = W_0 = w_0 \in (-1 + \delta, 1 - \delta)$. Let $A = (-1 + \delta, 1 - \delta)\backslash\{0\}$. We stop the algorithm at time $\tau_\mu^A = \{\inf k{:}W_k \notin A\}$. Denote

$$\begin{aligned} &\hat{G}(w, \mu) \\ &= E\{G(w, U, \mu)\} \\ &= \begin{cases} -\frac{\gamma w \ \mathrm{sgn}\ (w)}{\sqrt{\mu}} & w \in [-\sqrt{\mu}, \sqrt{\mu}] \\ -\gamma \ \mathrm{sgn}\ (w) & w \in (-1 + \delta, 1 - \delta) \cap [-\sqrt{\mu}, \sqrt{\mu}]^c \end{cases}. \end{aligned}$$

It is clear that for initial conditions well away from the target, the trajectory of the algorithm will be that of a symmetric random walk, and that there is a "centering" force once near the target.

To analyze the asymptotics of $W_\mu(t) = W_{[t/\mu]}$, check the conditions of the corollary. $\hat{G}$ bounded implies that the collection $\{\hat{G}(W_k, \mu\}$ is uniformly integrable and hence A1 holds. $E[(G(W_k, U_{k+1}, \mu) - \hat{G}(W_k, \mu))^2] \le E[G(W_k, U_{k+1}, \mu)^2] = E[(\gamma I_{(1/2-\sqrt{\mu}, 1/2)}(\psi_k))^2] = \gamma^2/\sqrt{\mu}$. Hence $\mu^2 \sum_{k=1}^{[t/\mu]} E[(G(W_k, Y_k, U_{k+1}, \mu) - \hat{G}(W_k, Y_k, \mu))^2] \le \mu^2 \frac{t}{\mu} \frac{\gamma^2}{\sqrt{\mu}} \to 0$, which verifies A2. A3 is satisfied by supposition. A4 is satisfied since the $\{\psi_k\}$ are i.i.d. $\hat{G}(w, \mu) \to \overline{G}(w) = -\gamma \mathrm{sgn}(w)$ a continuous function on K. Since $\hat{G}(w, \mu) \le 1$ for all (w, μ), A5′ is satisfied. In this case, the ODE (11) is simple enough to be solved, and the solution is $W(t) = w_0 - \gamma t \mathrm{sgn}(w_0)\ 0 \le t \le |w_0|/\gamma$.

If w_0 is positive, then $W(t)$ decreases at a rate γ until it hits zero, while if w_0 is positive, $W(t)$ increases until it gets to zero. Thus the ODE converges to the correct answer; the theorem ensures that the algorithm's tendency is to follow the ODE and cover the target area.

B. Signed Algorithm

The second example supposes that there are two squares with fixed sidelengths $d^1 = (d_1^1, d_2^1)$ and $d^2 = (d_1^2, d_2^2)$, and centers $s^1 = (s_1^1, s_2^1)$ and $s^2 = (s_1^2, s_2^2)$, which try to identify a target rectangle with sidelengths $d = (d_1, d_2)$ and center $s = (s_1, s_2)$. Let $\{X_k\}$ denote an i.i.d. sequence of uniformly distributed random variables over the unit square [0, 1] $\times$ [0, 1]. At each timestep, X_k is the input to the algorithm. Let $K_{(s^j, d^j)}(\cdot) = I_{R(s^j, d^j)}$. We propose the following algorithm: At each time step k, for each box i,

1) move towards the sample point if it is in the target area and is not in any box. ($I_T(X_k) = 1$ and $K_{(s_k^j, d^j)}(X_k) = 0$ for every j.)
2) move away from the sample point if a) the sample is in the ith box but not in the target area ($I_T(X_k) = 0$ and $K_{(s_k^i, d^i)}(X_k) = 1$)
   or if b) the sample is in multiple boxes ($K_{(s_k^i, d^i)}(X_k) = K_{(s_k^j, d^j)}(X_k) = 1$ for $i \ne j$.)
   In [16], the motion of the parameter estimates towards or away from the sample point is always in the $+/-$ $\mathrm{sgn}(x - s^i)$ direction, where $\mathrm{sgn}(\cdot)$ of a vector indicates an element by element operation. Note that step 2b) provides the conflict resolution.

This logic can be stated succinctly. For the first box, the update direction (at the k^{th} time step) $\mathrm{sgn}(X_k - s_k^1)$ is multiplied by

$$z^1(X_k) = -K_{(s_k^1, d^1)}(X_k) - K_{(s_k^2, d^2)}(X_k) I_T(X_k) + I_T(X_k)$$

while for the second box, the update in the direction $\mathrm{sgn}(X_k - s_k^2)$ is multiplied by

$$z^2(X_k) = -K_{(s_k^2, d^2)}(X_k) - K_{(s_k^1, d^1)}(X_k) I_T(X_k) + I_T(X_k).$$

Letting $W_k = (s^1_{1k}, s^1_{2k}, s^2_{1k}, s^2_{2k})^*$ (a four vector), the full algorithm is then

$$W_{k+1} = W_k + \mu \text{sgn}\begin{pmatrix}(X_k - s^1_k)z^1(X_k)\\(X_k - s^2_k)z^2(X_k)\end{pmatrix}$$
$$= W_k + \mu G(W_k, U_{k+1})$$

where $U_{k+1} = X_k$.

The $\text{sgn}(X_k - s^i_k))$ term acts like the "regressor" or "information" vector in the LMS adaptive filter, determining the direction that the update moves. The z^1 and z^2 take on values in the set $\{-1, 0, 1\}$ and act like a signed error term, determining whether the updates move in the plus or minus $\text{sgn}(X_k - s^i_k))$ direction. Thus, this algorithm is analogous to the well known sign-sign LMS (5) algorithm.

The first step in analyzing the behavior of the algorithm is to find the related ODE, which can be calculated using the fact that

$$E\{K_{(s,d)}(X)\text{sgn}(X - s^1)\}_i$$
$$= \begin{cases} -d(1)d(2) & \text{if } \tilde{s}(i) < -d(i)/2 \\ -2d(1)d(2)\tilde{s}(i)/d(i) & \text{if } |\tilde{s}(i)| < d(i)/2 \\ d(1)d(2) & \text{if } \tilde{s}(i) > -d(i)/2 \end{cases}$$

where $\tilde{s}(i) = s(i) - s^1(i)$. Letting $G(w) = E[G(w, U_{k+1})] = [G_1(w)\ G_2(w)]^*$ and $w(t) = (s^1_1(t), s^1_2(t), s^2_1(t), s^2_2(t))^*$,

$$G_1(w) = E[I_T(X)\text{sgn}(X - s^1(t))$$
$$- K_{(s^1(t),d^1(t))}(X)\text{sgn}(X - s^1(t))$$
$$- K_{(s^2(t),d^2(t))}(X)I_T(X)\text{sgn}(X - s^1(t))]$$
$$= E[I_T(X)\text{sgn}(X - s^1(t))]$$
$$- E[K_{(s^2(t),d^2(t))}(X)I_T(X)\text{sgn}(X - s^1(t))].$$

Similarly,

$$G_2(w) = E[I_T(X)\text{sgn}(X - s^2(t))]$$
$$- E[K_{(s^1(t),d^1(t))}(X)I_T(X)\text{sgn}(X - s^2(t))].$$

The resulting ODE is too complex to solve in closed form, though it can be easily solved numerically. Fig. 1(a) shows a typical trajectory of the ODE in which the two roving boxes lock onto the target area, moving to cover the desired region precisely. Actual trajectories of the algorithm, of course, do not proceed as smoothly, but the theorem assures us that they do, on average, follow the desirable trajectory of the ODE.

C. Unsigned Algorithm

Despite the popularity of the sign-sign LMS adaptive algorithm, one of its shortcomings is clear—it can only move the parameter estimates a small (stepsize dependent) distance at each iteration. In the adaptive filtering context, one can usually obtain significantly faster convergence using the sign-error algorithm or the unsigned LMS, since these allow larger motion of the parameter estimates when far from the desired answer. In the previous ACA, there is no easy way to "remove the sign" from the error terms z^1 and z^2, since they are formed from indicator functions. It is easy, however, to remove the sign from the regressor vector, and update in the directions $(x_k - s^i)$ rather than their signed versions. Using the same

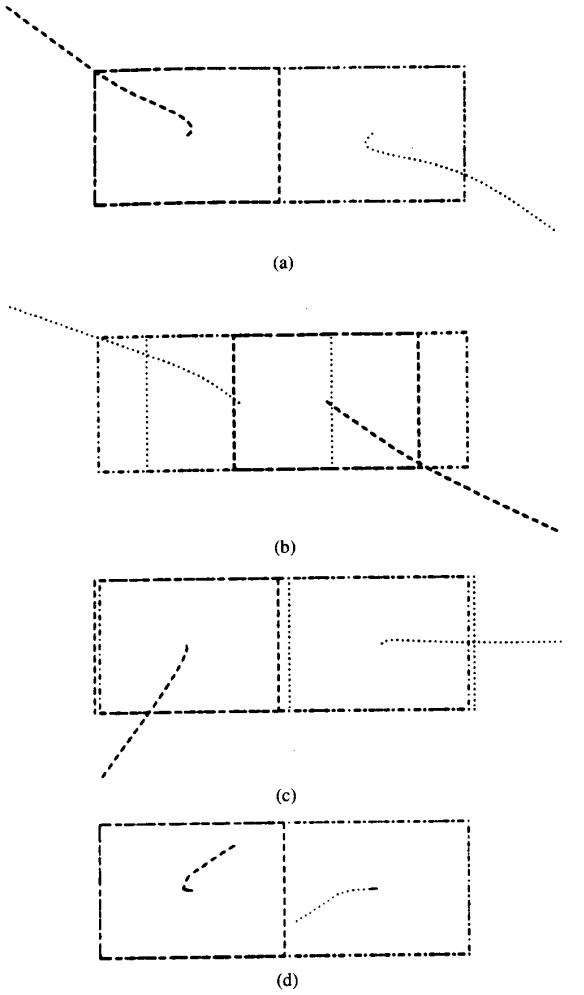


Fig. 1(a). Trajectory of ODE for signed algorithm, example 2; (b) trajectory of ODE for unsigned algorithm, example 3; (c) trajectory of ODE with Gaussian kernal function; (d) trajectory of ODE with Keifer–Wolfowitz form.

problem setup and notations as in the previous example (two squares of fixed sidelength searching for a target rectangle), this leads to the algorithm

$$W_{k+1} = W_k + \mu\begin{pmatrix}(X_k - s^1_k)z^1(X_k)\\(X_k - s^2_k)z^2(X_k)\end{pmatrix}$$
$$= W_k + \mu G(W_k, U_{k+1}) \tag{13}$$

which can be examined as before. Noting that

$$E\{K_{(s^2_k,d^2)}(X)(X - s^1_k)\}$$
$$= \int_{[0,1]\times[0,1]} K_{(s^2_k,d^2)}(x - s^1_k)dx$$
$$= \int_{K_{(s^2_k,d^2)}} (x - s^1_k)dx$$
$$= d^2(1)d^2(2)\left[\frac{1}{d^2(1)d^2(2)}\int_{K_{(s^2_k,d^2)}} (x - s^1_k)dx\right]$$
$$= d^2(1)d^2(2)[s^2_k - s^1_k]$$

the first component of $G(w) = [G_1(w) \quad G_2(w)]^*$ is

$$\begin{aligned} G_1(w) &= E[I_T(X)(X - s^1(t)) \\ &\quad - K_{(s^1(t),d^1(t))}(X)(X - s^1(t)) \\ &\quad - K_{(s^2(t),d^2(t))}(X)I_T(X)(X - s^1(t))] \\ &= d(1)d(2)(s - s^1(t)) \\ &\quad - E[K_{(s^2(t),d^2(t))}(X)I_T(X)(X - s^1(t))] \end{aligned}$$

since $I_T(X) = K_{(s(t),d(t))}(X)$. Similarly, the second component of $G(w)$ is given by

$$\begin{aligned} G_2(w) &= d(1)d(2)(s - s^2(t)) \\ &\quad - E[K_{(s^1(t),d^1(t))}(X)I_T(X)(X - s^2(t))]. \end{aligned}$$

This ODE is too complex to solve in closed form, although it too can easily be integrated numerically. Fig. 1(b) shows a trajectory of the ODE which does not converge to the desired target figure—the two boxes overlap significantly! It is easy to check that the solutions of $G(w) = 0$ include $(s^1, s^2) = (s + (\lambda\ 0)^*, s - (\lambda\ 0)^*)$, $0 \le \lambda \le d(1)/4$. ($\lambda = d(1)/4$ is the desired solution.) Such a dense set of equilibria near the desired one virtually guarantees poor performance of this algorithm. This was quite a surprise to us, since the unsigned LMS is generally considered superior to the signed versions [12], though both usually function adequately. For ACA's, however, the signed version functions well, while the unsigned version fails completely. This demonstrates the usefulness of the ODE analytical approach. We are able to state categorically that the unsigned algorithm (13) would be a poor choice for this problem.

D. Gaussian Kernel

This example uses the gradient approach with cost function (7) and the Gaussian kernel

$$K_{(s^i,d^i)}(x) = \exp\left(-\frac{(x_1 - s_1^i)^2}{d_1^i}\right)\exp\left(-\frac{(x_2 - s_2^i)^2}{d_2^i}\right).$$

The update function $V(\cdot)$ of (8) can be evaluated straightforwardly since

$$\frac{\partial}{\partial s^i}K_{(s^i,d^i}(x) = -K_{(s^i,d^i)}(x)2(x - s^i)/d^i.$$

As in the previous two examples, consider the case where two adaptive boxes with fixed sidelength seek to identify a target rectangle. The algorithm is

$$W_{k+1} = W_k + \mu \begin{pmatrix} e_k(s,d)K_{(s_k^1,d^1)}(X_k)\frac{X_{1k} - s_{1k}^1}{d_1^1} \\ e_k(s,d)K_{(s_k^1,d^1)}(X_k)\frac{X_{2k} - s_{2k}^1}{d_2^1} \\ e_k(s,d)K_{(s_k^2,d^2)}(X_k)\frac{X_{1k} - s_{1k}^2}{d_1^2} \\ e_k(s,d)K_{(s_k^2,d^2)}(X_k)\frac{X_{2k} - s_{2k}^2}{d_2^2} \end{pmatrix}$$

where $e_k(s,d) = 4(I_{R(s,d)}(X_k) - \sum_{i=1}^2 K_{(s_k^i,d^i)}(X_k))$.

With the algorithm (8), the associated differential equation can be written as $\dot{w} = G(w)$, where $G(w) = (G_{11}(w), G_{12}(w), G_{21}(w), G_{22}(w))^*$ and where

$$\begin{aligned} G_{ij}(w) &= 4E\{(I_{R(s,d)}(X) \\ &\quad - \sum_{l=1}^2 K_{(s^l(t),d^l)}(X))K_{(s^i(t),d^i)}(X)\frac{X(j) - s_j^i(t)}{d_j^i}\}. \end{aligned}$$

This last expression may be further expanded into a lengthy expression involving error functions. Fig. 1(c) shows a trajectory of the ODE converging to the desired covering. All of the starting points that we studied demonstrate this same qualitative behavior. As a Gaussian kernel is hard to plot, we plot instead a rectangle with sidelengths in a given direction given by the standard deviation in that direction. Although the kernels and target are not matched very well, the convergence behavior of the center points is very good, indicating the robustness of the approach to mismatch between the expected target shape and the shape of the kernels.

E. Kiefer–Wolfowitz Type Algorithm

Let $e_1 = (1\ 0\ 0\ 0)^*, e_2 = (0\ 1\ 0\ 0)^*, \ldots, e_4 = (0001)^*$. Then the algorithm is of the form:

$$\begin{aligned} W_{k+1} &= W_k + \frac{\gamma\sqrt{\mu}}{2}(I_T(X_k) - \sum_{i=1}^2 K_{(s^i,d^i)}(X_k)) \\ &\quad \times \begin{pmatrix} K_{(s_k^1+\sqrt{\mu}e_1,d^1)}(X_k) - K_{(s_k^1-\sqrt{\mu}e_1,d^1)}(X_k) \\ K_{(s_k^1+\sqrt{\mu}e_2,d^1)}(X_k) - K_{(s_k^1-\sqrt{\mu}e_2,d^1)}(X_k) \\ K_{(s_k^2+\sqrt{\mu}e_3,d^2)}(X_k) - K_{(s_k^2-\sqrt{\mu}e_3,d^2)}(X_k) \\ K_{(s_k^2+\sqrt{\mu}e_4,d^2)}(X_k) - K_{(s_k^2-\sqrt{\mu}e_4,d^2)}(X_k) \end{pmatrix}. \end{aligned}$$

We first need to define some notation. For a given rectangle A, with sides parallel to the coordinate axes, define its width in the x-coordinate direction as $s_u(A) = s_b(A)$, and its height in the y-coordinate direction as $s_r(A) = s_l(A)$.

Consider the rectangle formed by the intersection of the ith square S^i with another rectangle A (either the other square S^j or the target T). The intersection $A \cap S^i$ is of course another rectangle. If the right (left, top, bottom) side of S^i intersects $A \cap S^i$, set $\Delta_{Ai}^{r(l,t,b)} = s_{r(l,t,b)}(A \cap S^i)$ (otherwise set $\Delta_{Ai}^{r(l,t,b)} = 0$).

One needs to take the expectation of the term in the parentheses of the algorithm, divide by $\sqrt{\mu}$ and take the limit as $\mu \to 0$. One obtains after a little effort

$$\dot{w} = \begin{pmatrix} \Delta_{T1}^r - \Delta_{T1}^l - \Delta_{S^21}^r + \Delta_{S^21}^l \\ \Delta_{T1}^t - \Delta_{T1}^b - \Delta_{S^21}^t + \Delta_{S^21}^b \\ \Delta_{T2}^r - \Delta_{T2}^l - \Delta_{S^12}^r + \Delta_{S^12}^l \\ \Delta_{T2}^t - \Delta_{T2}^b - \Delta_{S^12}^t + \Delta_{S^12}^b \end{pmatrix}.$$

Fig. 1(d) shows a trajectory of the ODE which is converging to the desired covering. All of the starting points that we studied demonstrated the same sort of behavior.

V. Experiments and Simulations

The examples of the previous sections verify that for simple problems in which the target can be exactly covered by the moving shapes, certain of the ACA's behave well. This section presents simulated evidence that the ACA approach is actually applicable to a much larger class of problems.

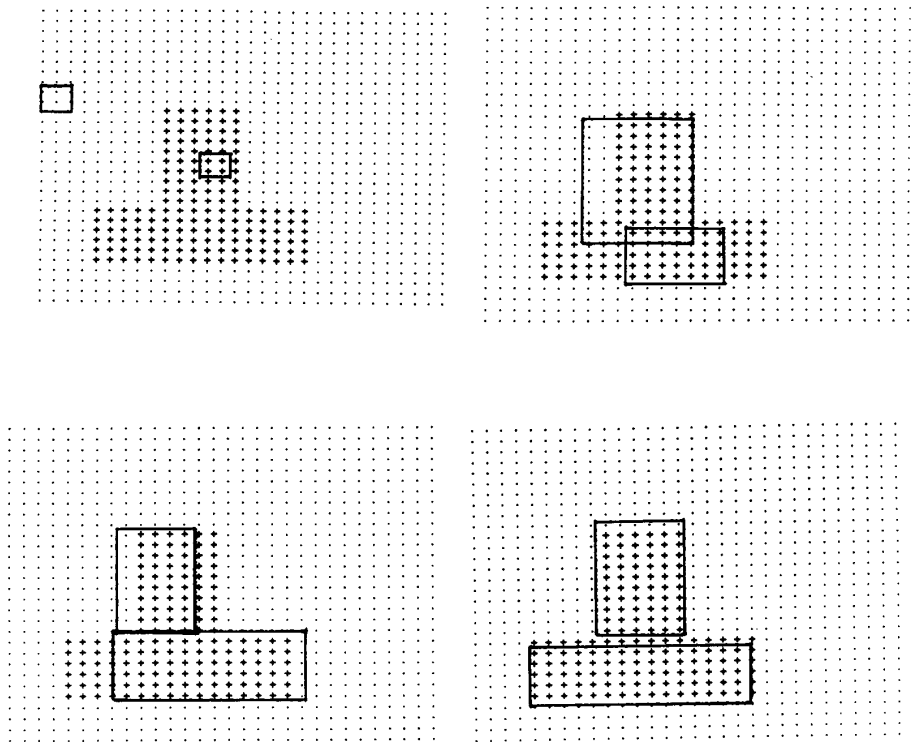


Fig. 2(a). Automated learning algorithm: Two rectangles trying to match an inverted T.

We first used the "signed" algorithm in which a varying number of rectangles are adapted to attempt to match an "unknown" target shape which is an inverted T shape. Fig. 2(a) shows two adapting rectangles which successfully converge to cover the desired figure after (about) a thousand iterations. Each iteration consists of some logic and some additions, so that the computational burden is not onerous. Fig. 2(b) shows three roving rectangles that try to match the inverted T shape, although 10% of the points are randomly misclassified. That the rectangles can do such a good job in a high noise setting argues for the robustness of the algorithm. Fig. 2(c) examines the effect of overparameterization, where five moving rectangles are adapted to cover the inverted T. There are no catastrophic effects of overparameterization, though there is somewhat more jiggling about before convergence occurs. Fig. 2(d) shows the time simulation using the Keifer–Wolfowitz style algorithm of section IV-E.

VI. Conclusion

This paper has introduced a family of algorithms for the automatic learning of complex decision regions which are inspired by adaptive filtering algorithms. These have been analyzed, and the behavior of the algorithms has been related to the evolution of a deterministic ordinary differential equation. When the ODE is stable, the algorithms converge to a region about their (possibly local) optimum. When the ODE is unstable, the algorithms will misbehave. Several simple examples were given to verify the applicability of the analysis. The work in this paper suggests several possible avenues for further investigation.

One of the major tasks is to generalize the algorithm class to more realistic problem settings. The specific algorithms presented here all operate in a binary environment; the new sample point is either inside or outside the target region. It is clearly possible to generalize this to handle grey scale (or other multilevel targets). Similarly, one could create algorithms in which the primitive shapes are adapted to textures (for instance, the Fourier transform near the sample point could be used in place of the indicator function).

A second major task is to deal with the local nature of the algorithms. One approach is to initially overparameterize the problem (use "too many" primitive shapes), and to then allow shapes to "die away" or disappear if they fail to capture an appropriate number of target points. An alternative is to use a fixed number of shapes which can die away, but to allow the possibility of resurrection. Both of these schemes work quite well (in simulation) to avoid local minima. Analysis of the schemes is tricky because of the introduction of the discontinuous switch (the alive or dead indicator) and the discontinuity of the parameter vector in the latter method.

The algorithms presented here are a start toward a solution of the covering problem, and the analysis technique gives clear evidence that theoretical results are possible.

Appendix

Let η, ψ be any two probability measures on some metric space (S, d) where d is the metric. Let $\mathcal{B}(S)$ denote the

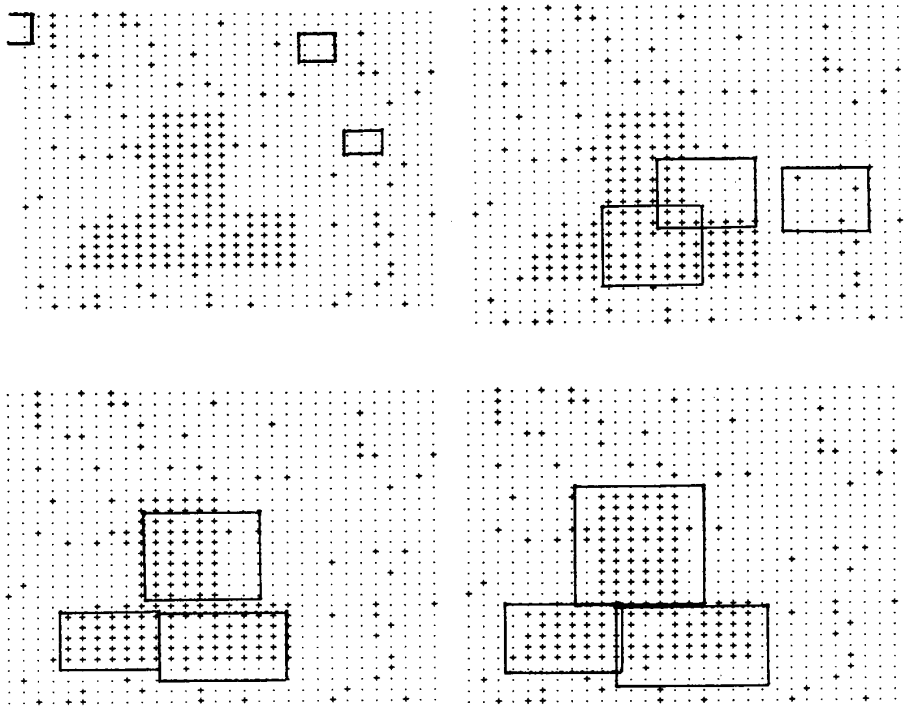


Fig. 2(b). Three rectangles trying to match an inverted T, with 10% misclassification error.

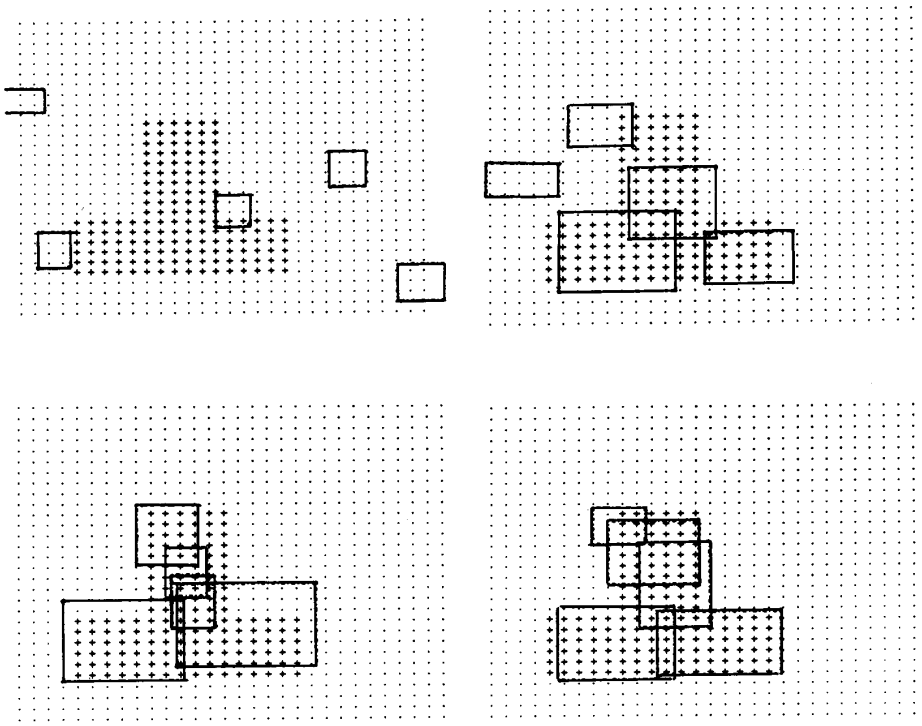


Fig. 2(c). Overparameterization: Five rectangles adapt to the inverted T.

Borel sets of S. For a set $A \in \mathcal{B}(S)$, define $A^\epsilon = \{x \in S: \inf_{y \in A} d(x,y) < \epsilon\}$. Define a distance measure between any two probability measures on (S,d) as

$$\rho(\eta,\psi) = \inf\{\epsilon > 0: \eta(F) \le \psi(F^\epsilon) + \epsilon \text{ for all } F \in \mathcal{C}\}$$

where $\mathcal{C}$ is the collection of closed sets of S. ρ can be shown

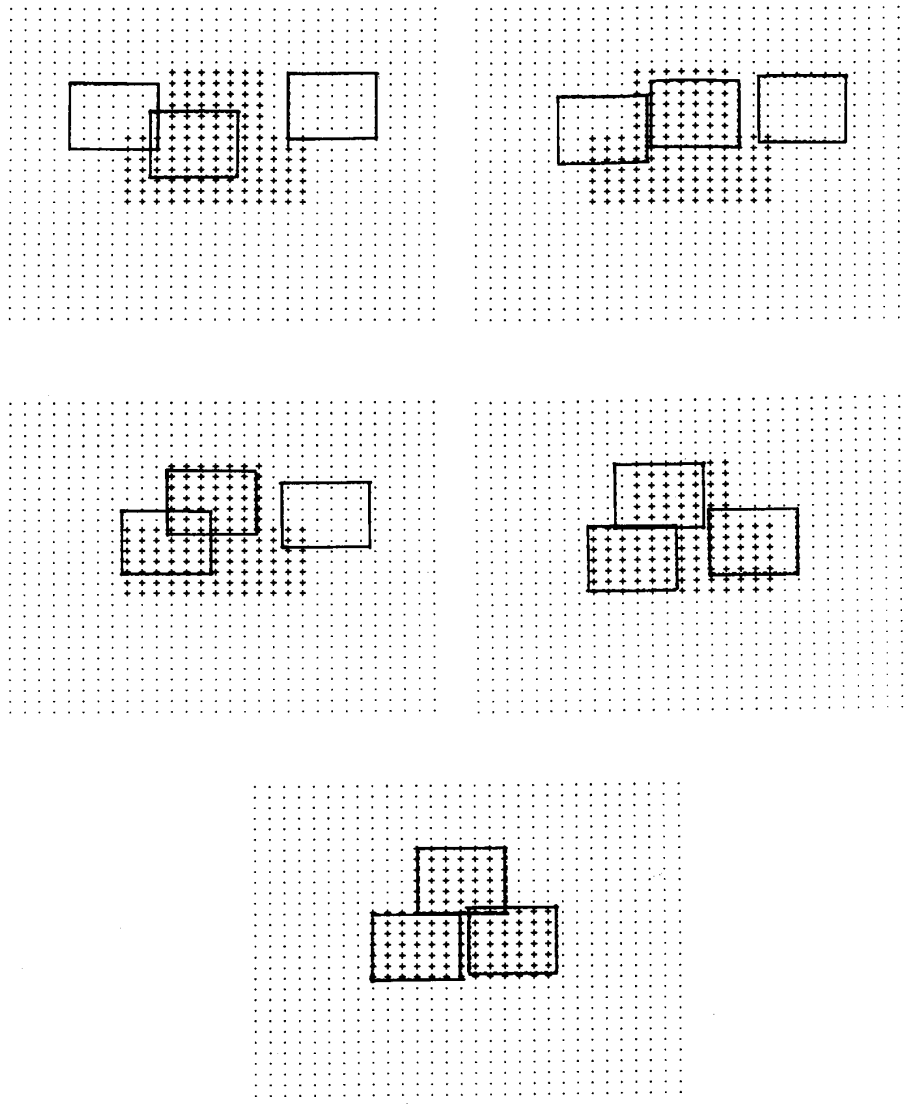


Fig. 2(d). Keifer–Wolfowitz style algorithm of Section IV-E.

to be a metric, and is called the *Prohorov metric* for the space of probability measures on S. When a sequence of probability measures η_n converges to another probability measure η under this metric (i.e., $\rho(\eta_n, \eta) \to 0$), we say that η_n *converges weakly* to η. The topology (i.e., the open sets) created by the Prohorov metric is called the *topology of weak convergence*. In the case when $S = \Re^r$, weak convergence of probability measures corresponds exactly to convergence in distribution.

Consider the space $D_{\Re^r}[0, \infty)$ of right continuous functions with left limits. The trajectories of the random processes $\{W_\mu(t)\}$ lie in this space. One would like to define a distance measure for this space for which it will be a nice (separable) metric space. The metric $d_S(\cdot, \cdot)$ that is typically used is called the *Skorohod metric*. The topology created by this metric is called the *Skorohod topology*. The actual definition of the metric is complicated. However, for the case of measuring the distance between two continuous functions $f, g \in D_{\Re^r}[0, \infty)$, it corresponds to a weighted "sup" norm, i.e.

$$d_S(f, g) = \int_0^\infty \exp(-u) 1 \wedge \sup_{0 \le t \le u} ||f(t) - g(t)|| du.$$

Let (S, d) be a complete separable metric space and let $\mathcal{M}(S)$ be the space of finite measures on S with the weak topology. Let $\mathcal{L}(S)$ be the space of measures on $[0, \infty) \times S$ such that for every $\mu \in \mathcal{L}(S)$, $\mu([0, t] \times S) < \infty$ for each $t \ge 0$. For $\mu \in \mathcal{L}(S)$, let μ^t denote the restriction of μ to $[0, t] \times S$. Let r_t denote the Prohorov metric on $\mathcal{M}([0, t] \times S)$ and define $\hat{r}$ on $\mathcal{L}(S)$ by

$$\hat{r}(\mu, \nu) = \int_0^\infty \exp(-t) 1 \wedge r_t(\mu^t, \nu^t) dt.$$

$\overline{C}(A)$ is defined as the space of all bounded continuous-functions on the metric space A. For a metric space E, let $D_E[0,\infty)$ be the space of right continuous E-valued functions with left limits endowed with the Skorohod topology. See [4] for further definitions and properties of this space.

We first state a necessary lemma.

Lemma 1 Let $\{(x_n,\mu_n)\} \subset D_E[0,\infty) \times \mathcal{L}(S)$, and $(x_n,\mu_n) \to (x,\mu)$. Let $h \in \bar{C}(E \times S)$. Suppose further that h_n converges uniformly to h on $E \times S$. Define

$$u_n(t) = \int_{[0,t]\times S} h_n(x_n(s),y)\mu_n(ds \times dy),$$

$$u(t) = \int_{[0,t]\times S} h(x(s),y)\mu(ds \times dy).$$

Let $z_n(t) = \mu_n([0,t] \times S)$ and $z(t) = \mu([0,t] \times S)$.
If x is continuous on $[0,t]$ and $\lim_{n\to\infty} z_n(t) = z(t)$, then $\lim_{n\to\infty} u_n(t) = u(t)$.

Proof: The proof is given in [3].

Proof of the theorem: Let M_μ be the martingale defined by

$$M_\mu(t) = \sum_{k=1}^{[t/\mu]} (G(W_k,Y_k,U_{k+1},\mu) - \hat{G}(W_k,Y_k,\mu))\mu.$$

Note that

$$W_\mu(t) = W_\mu(0) + M_\mu(t) + \sum_{k=1}^{[t/\mu]} \hat{G}(W_k,Y_k,\mu)\mu.$$

A2 states that the quadratic variation process of M_μ is converging to zero. Hence, it follows that $M_\mu \Rightarrow 0$. By (lemma 7 on p. 51 of [8]), A1 implies that $V_\mu = \sum_{k=1}^{[t/\mu]} \hat{G}(W_k,Y_k,\mu)\mu$ is relatively compact. This fact and A3 give us relative compactness of $\{W_\mu\}$.

Define a measure on $\Re \times E$ as

$$\Gamma_\mu([0,t] \times B) = \sum_{k=0}^{[t/\mu]-1} I_B(Y_k)\mu.$$

By ergodicity, $\Gamma_\mu \to m \times \nu_Y$. We may then write

$$W_\mu(t) = W_\mu(0) + M_\mu(t) + \int_{[0,t]\times E} \hat{G}(W_\mu(s^-),y,\mu)\Gamma_\mu(ds \times dy).$$

We may suppose by the Skorohod embedding theorem that (W_μ, M_μ) is converging (on some other probability space) almost surely to $(W,0)$, where W is some random process on $C[0,\infty)$. For each ω in that probability space, $W_\mu(t,\omega)$ is converging (as a sequence of functions) to $W(t,\omega) \in C[0,\infty)$. Also for each $\omega, \Gamma_\mu \to m \times \nu_Y$. Hence by the lemma, $W(t,\omega)$ must satisfy

$$W(t,\omega) = w_0 + \int_{[0,t]\times E} \tilde{G}(W(s,\omega),y)ds \times \nu_Y(dy) = w_0 + \int_0^t \overline{G}(W(s,\omega))ds$$

which is a deterministic ODE. Since this behavior is almost sure in this other probability space, we have the asserted relative compactness and limit point behavior in the original space. □

Proof of Corollary 1 We need only show that $M_\mu \Rightarrow 0$. Consider bounding the increments of $M_\mu(t)$as follows:

$$\sup_{s\le h} |M_\mu(t+s) - M_\mu(t)| \le \sum_{k=[t/\mu]+1}^{[(t+h)/\mu]} \sup_{w\in K} |G(w,Y_k,U_{k+1},\mu)| + \sup_{w\in K} |\hat{G}(w,Y_k,\mu)| \to h[E[\sup_{w\in K} |G(w,Y_k,U_{k+1},\mu)|] + E[\sup_{w\in K} |\hat{G}(w,Y_k\mu)|]] = Ch \text{ a.s}$$

This bound will be uniform for $t \le T$ almost surely. Therefore

$$\limsup_{\mu\to 0} \sup_{t\le T} \sup_{s\le h} |M_\mu(t+s) - M_\mu(t)| \le Ch$$

which implies that $\{M_\mu\}$ is relatively compact. The same-bounding technique shows that the total variations up to time t are bounded in L_1. The martingale property then implies that $M_\mu \Rightarrow 0$.

Proof of Corollary 2 We suppose that $K = \Re^r$. Otherwise, work with the stopped process. Note that

$$W_\mu(t) = W_\mu(0) + M_\mu(t) + \int_0^t \hat{G}(W_\mu(s^-),\mu)ds.$$

By a Skorohod embedding we may suppose that $(W_\mu, M_\mu) \to (W,0)$ almost surely. The first part of A5′ implies for each ω, $\hat{G}(W_\mu(s^-),\mu) \to \overline{G}(W(s))$. The second part of A5′ that $\int_0^t \hat{G}(W_\mu(s^-),\mu)ds \to \int_0^t \overline{G}(W(s))ds$, by the dominated convergence theorem. The remainder of the argument is identical to the theorem proof. □

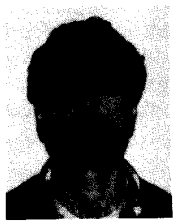
ACKNOWLEDGMENT

The authors would like to thank T.G. Kurtz of the University of Wisconsin Mathematics Department for the proof of the first part of Theorem 1.

REFERENCES

[1] E. Baum, "Neural network algorithms that learn in polynomial time from examples and queries," *IEEE Trans. Neural Networks*, vol. 2, pp. 5–19, 1991.
[2] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*. New York: Springer-Verlag, 1990.
[3] J. A. Bucklew, T. G. Kurtz, and W. A. Sethares, "Weak convergence and local stability properties of fixed step size recursive algorithms," *IEEE Trans. Inform. Theory*, vol. 39, no. 3, pp. 966–978, May 1993.
[4] S. Ethier and T. Kurtz, *Markov Processes—Characterization and Convergence*. New York: Wiley-Interscience, 1986.
[5] A. Gersho, "Adaptive filtering with binary reinforcement," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 2, pp. 191–198, Mar. 1984.
[6] K. L. Halliwell, R. C. Williamson, and I. M. Y. Mareels, "Learning nonlinearly parameterized decision regions," to appear in *J. Math. Syst., Estimat., Contr.*.
[7] J. Kiefer and J. Wolfowitz, "Stochastic estimation of the modulus of a regression function," *Ann. Math. Stat.*, vol. 23, pp. 462–466, 1952.
[8] H. J. Kushner, *Approximation and Weak Convergence Methods for Random Processes*. Cambridge, MA: MIT Press Series in Signal Processing, Optimization, Control, 1984.
[9] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Trans. Automat. Contr.*, vol. 22, no. 4, pp. 551–575, Aug. 1977.
[10] R. W. Lucky, "Techniques for adaptive equalization of digital communication systems," *Bell Syst. Tech. J.*, vol. 45, Feb. 1966.
[11] M. L. Minsky and S. A. Papert, *Perceptrons*. Cambridge, MA: MIT Press, 1988.

[12] W. A. Sethares and C. R. Johnson, Jr., "A comparison of two quantized state adaptive algorithms," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 1, pp. 138–143, Jan. 1989.
[13] J. Sklansky and G. N. Wassell, *Pattern Classifiers and Trainable Machines*. New York: Springer-Verlag, 1981.
[14] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters*. Boston: Kluwer, 1990.
[15] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
[16] R. C. Williamson and W. A. Sethares, "A provably convergent perceptron-like algorithm for learning hypercubic decision regions," in *Int. Conf. Artificial Neural Networks* (Helsinki), June 1991.



James A. Bucklew (S'75–M'79) received the Ph.D. degree from Purdue University in 1979.

He is currently a Professor at the University of Wisconsin-Madison in the Department of Electrical and Computer Engineering and in the Department of Mathematics. He has served as the Associate Editor-at-Large (1989–1992) and the Associate Editor-Detection (1992) for the IEEE TRANSACTIONS ON INFORMATION THEORY. His research interests are in the applications of probability to signal processing and communications problems. He is the author of the book *Large Deviation Techniques in Decision, Simulation, and Estimation* (Wiley-Interscience, 1990).

Dr. Bucklew is the recipient of a Presidential Young Investigator Award (1984).

William A. Sethares (S'86–M'87) received the B.A. degree in mathematics from Brandeis University, Waltham, MA, and the M.S. and Ph.D. degrees in electrical engineering from Cornell University, Ithaca, NY.

He has worked at the Raytheon Company as a Systems Engineer and is currently on the faculty of the Department of Electrical and Computer Engineering at the University of Wisconsin in Madison. His research interests include adaptive systems in signal processing, communications, control, and electronic music. He especially enjoys writing brief biographical sketches.