Fixed Step Size Recursive Algorithms for The Covering Problem

J.A. Bucklew and W.A. Sethares

Department of Electrical and Computer Engineering
University of Wisconsin-Madison, Madison, WI 53706 USA

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 to the evolution of a deterministic ordinary differential equation. 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.

1 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 [8], which are variants of known adaptive filtering methods [7]. We call this family of techniques, adaptive covering algorithms (ACAs).

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

$$W_{k+1} = W_k + \mu \{\textit{correction term}\} \qquad (1)$$

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 form will cause the parameter vector sequence $\{W_k\}$ to improve with time on average. In certain cases, an analytical technique in the spirit of [1, 2, 5, 6] 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.

2 Candidate Algorithm Generation

We apply gradient descent notions to the covering problem. For any set $A \in \Re^d$, let $I_A(\cdot) : \Re^d \rightarrow \{0,1\}$ denote the indicator function of the set A. Suppose there are n parameterized "shape" or kernel functions $K_{\mathbf{a}^{\mathbf{i}}}(\cdot) : \Re^d \rightarrow \Re^1 \;\; i = 1, \ldots, n$ where $\mathbf{a}^{\mathbf{i}} \in \Re^m$ is the i^{th} parameter vector $\mathbf{a}^{\mathbf{i}} = (a_1^i, a_2^i, \ldots, a_m^i)$. A typical example (for $d = 2$) 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 side lengths $d = (d_1, d_2)$. Let X denote a d-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 type objective function:

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

0-7803-0946-4/93 $3.00 © 1993 IEEE

One can see that if for example $K_{\mathbf{a}}(\cdot) = I_{R(s,d)}(\cdot)$, the argument of the expectation is 0 at a point $x \in T$ if and only if 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 (2), the gradient descent method leads to the algorithm

$$(\mathbf{a}^{\mathbf{1}}_{\mathbf{k+1}}, \mathbf{a}^{\mathbf{2}}_{\mathbf{k+1}}, \ldots, \mathbf{a}^{\mathbf{k}}_{\mathbf{k+1}}) = (\mathbf{a}^{\mathbf{1}}_{\mathbf{k}}, \mathbf{a}^{\mathbf{2}}_{\mathbf{k}}, \ldots, \mathbf{a}^{\mathbf{k}}_{\mathbf{k}}) - \mu V(X) \tag{3}$$

where $V(x) = \nabla |I_T(x) - \sum_{i=1}^{n} K_{\mathbf{a}^{\mathbf{i}}}(x)|^2$ 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 (3). This is reminiscent of the Kieffer-Wolfowitz [4] algorithm 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 such an algorithm in [3]

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

$$K_{\mathbf{a}}(x) = \exp(-[x-s]^* D [x-s]) \tag{4}$$

where D is the diagonal matrix whose nonzero entries are the d vector. $V(\cdot)$ is differentiable and (3) can be implemented directly. We investigate this algorithm in [3].

The Gaussian kernel of (4) 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, $E\{|I_T(X) - \sum_{i=1}^{n} K_{\mathbf{a}^{\mathbf{i}}}(X)|^2\}$, one might use the L^1 error, $E\{|I_T(X) - \sum_{i=1}^{n} K_{\mathbf{a}^{\mathbf{i}}}(X)|\}$. The former leads to algorithms which might be thought of as analogs of LMS, while the latter lead 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.

3 Local Stability and Weak Convergence Analysis

Consider an ACA as a discrete time iteration process

$$W_{k+1} = W_k + \mu G(W_k, Y_k, U_{k+1}, \mu) \tag{5}$$

where W_k is the parameter vector of weights that define the primitive shapes, μ is the stepsize, $U_{k+1} = q(W_k, Y_k, \psi_k)$, $\{\psi_k\}$ i.i.d. is an input vector that usually just 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 ACAs. 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 (5) (without the dependence of G on μ) is considered extensively in [2] and in the book by Beneveniste et al [1]. Kushner [5] studies μ-dependent algorithms but uses different methods. Our theorems are different from theirs, following the techniques and methods of [2].

We relate the behavior of the algorithm (5) for small μ to the behavior of the associated deterministic ordinary differential equation (ODE)

$$W(t) = W_0 + \int_0^t \bar{G}(W(s))ds \tag{6}$$

where $\bar{G}(\cdot)$ is a smoothed version of $G(\cdot,\cdot,\cdot,\cdot)$.

Suppose that (W_k, Y_k, U_k) is adapted to the filtration $\{\mathcal{F}_k\}$, and 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. A time scaled version of W is defined as

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

where $[z]$ means the integer part of z.

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

A4) $\{Y_k\}$ is a stationary ergodic sequence of E valued random variables.

A5) $\hat{G}(w, y, \mu)$ converges uniformly on $\Re^d \times E$ to a 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 in $C[0, \infty)$. Furthermore, every limit point of $\{W_\mu\}$ satisfies (6).*

Proofs of all theorems are presented in [3].

The theorem asserts that the ACAs (5) will behave like the ODE (6) 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). Convergence in probability means that for every $T > 0$, $\epsilon > 0$, $\lim_{\mu \to 0} P(\sup_{0 \leq t \leq T} |W_\mu(t) - W(t)| > \epsilon) = 0$. 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.

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.

Corollary 1 *Suppose the algorithm form is*

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

where $\{\mathcal{F}_k\}$, $\hat{G}$ and W_μ are defined as before. Assume A1′ (or A1), A2, and A3. Replace A5 with
A5′) $\hat{G}(w, \mu)$ converges to $\bar{G}(w)$ a continuous function for all $w \in A$. Furthermore, suppose that $\sup_{w \in A} \hat{G}(w, \mu) \leq B < \infty$. *Then the conclusions of the previous theorem hold.*

A2 is frequently an onerous condition to check. It depends on the $\{W_k\}$ values themselves which we are trying to obtain information about in the first place. Therefore we may wish to replace A2 with the following condition $A6$ and obtain the final corollary.
A6) $E[\sup_{w \in A} |G(w, Y_k, U_{k+1}, \mu)|] < \infty$ and $E[\sup_{w \in A} |\hat{G}(w, Y_k, \mu)|] < \infty$.

Corollary 2 *Suppose A1, A3, A5, A6. Then every limit point of $\{W_\mu\}$ satisfies (6).*

4 Examples and Applications

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

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 can not exactly cover the target. Stability of the ODE suggests that the square will center itself on the circle and adjust its side length so as to tradeoff the target area uncovered with the nontarget area covered. This is, indeed, the observed behavior of the successful algorithms.

In [3], we give more details of the analyzes and consider three other examples.

4.1 A One Dimensional 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 (2). The input to the algorithm consists only of the sample points $\{X_k\}$. The algorithm (3) becomes

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

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

$$W_{k+1} = W_k + \gamma\mu[I_T(U_{k+1}) - 1]$$

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

This is in the form of (1) of the corollary 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. One can show that the solution to the limiting ODE is $W(t) = w_0 - \gamma t \mathrm{sgn}(w_0) \;\; 0 \leq t \leq |w_0|/\gamma$.

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 true answer, the theorem ensures that the algorithm's tendency is to follow the ODE and cover the target area.

4.2 Signed Algorithm

The second example supposes that there are two squares with fixed side lengths $d^1 = (d_1^1, d_2^1)$ 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 side lengths $d = (d_1, d_2)$ 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 time step, 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 $K_{(s^j,d^j)}(X_k) = 0$ 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 $K_{(s^i,d^i)}(X_k) = 1$)
or if (b) the sample is in multiple boxes ($K_{(s^i,d^i)}(X_k) = K_{(s^j,d^j)}(X_k) = 1$ for $i \neq j$.)
In [8], 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 2(b) provides the conflict resolution.

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

$$z^1(X_k) = -K_{(s^1,d^1)}(X_k) - K_{(s^2,d^2)}(X_k) I_T(X_k) + I_T(X_k) \tag{7}$$

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

$$z^2(X_k) = -K_{(s^2,d^2)}(X_k) - K_{(s^1,d^1)}(X_k) I_T(X_k) + I_T(X_k). \tag{8}$$

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

$$\begin{aligned} W_{k+1} &= W_k + \mu \mathrm{sgn} \begin{pmatrix} (X_k - s^1) z^1(X_k) \\ (X_k - s^2) z^2(X_k) \end{pmatrix} \\ &= W_k + \mu G(W_k, U_{k+1}) \end{aligned} \tag{9}$$

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

We develop the ODE for this algorithm in [3] and demonstrate through it and simulations that this ACA is a reasonable choice. We then analyze an "unsigned" version of this algorithm and demonstrate that it is unstable.

References

[1] A. Benveniste, M. Metivier, P. Priouret, *Adaptive Algorithms and Stochastic Approximations*, Springer-Verlag, New York, 1990.

[2] J. Bucklew, T. Kurtz, and W. Sethares, "Weak Convergence and Local Stability Properties of Fixed Step Size Recursive Algorithms," Presented at 1992 International Conference on Acoustics, Speech, and Signal Processing, San Francisco, CA, March 1992 (to appear IEEE Trans. Info. Thy.).

[3] J. Bucklew and W. Sethares, "The Covering Problem: Learning Decision Regions via Adaptive Algorithms," Submitted to IEEE Transactions on Signal Processing.

[4] J. Kiefer and J. Wolfowitz, "Stochastic Estimation of the modulus of a regression function," *Ann. Math. Stat.*, Vol 23, pp. 462-466, 1952.

[5] H.J. Kushner, *Approximation and Weak Convergence Methods for Random Processes*, MIT Press Series in Signal Processing, Optimization, and Control, Cambridge, Massachusetts 1984.

[6] L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Trans. on Automatic Control* Vol. 22, No. 4, pp. 551-575, August 1977.

[7] B. Widrow, and S. D. Stearns, *Adaptive Signal Processing,* Prentice-Hall, Englewood Cliffs, New Jersey, 1985.

[8] R. C. Williamson and W. A. Sethares, "A provably convergent perceptron-like algorithm for learning hypercubic decision regions," Intl. Conference on Artificial Neural Networks, Helsinki, Finland, June 1991.