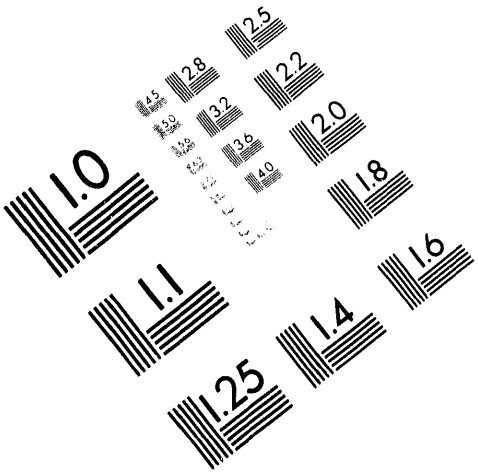
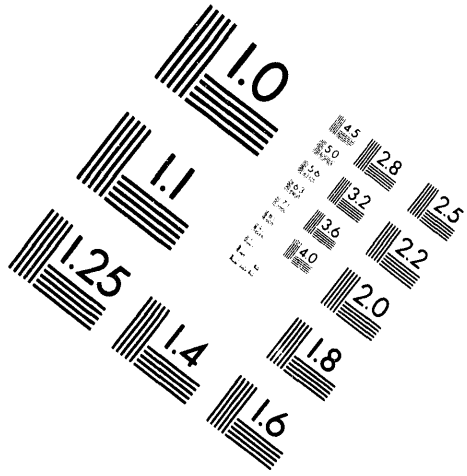
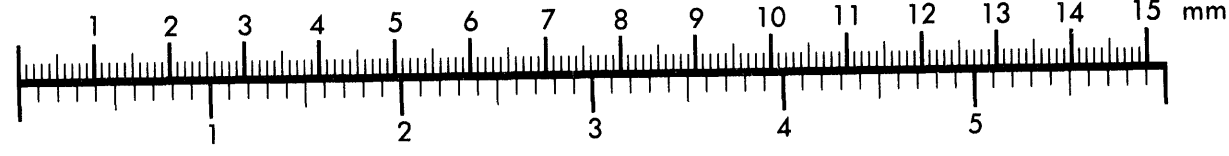


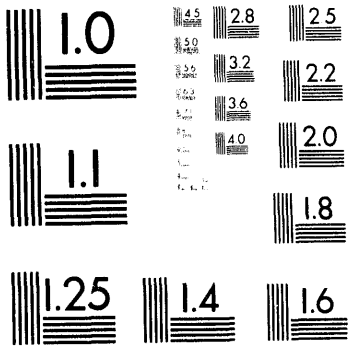
Association for Information and Image Management

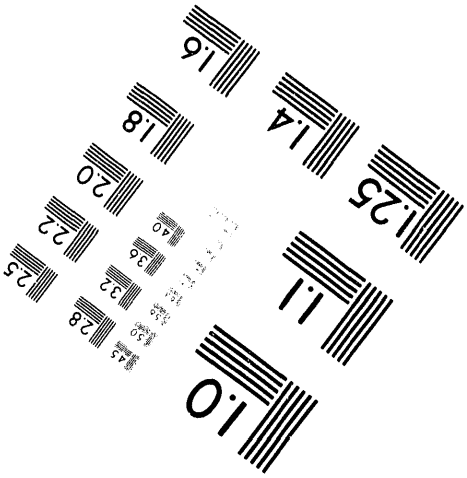
1100 Wayne Avenue, Suite 1100
Silver Spring, Maryland 20910

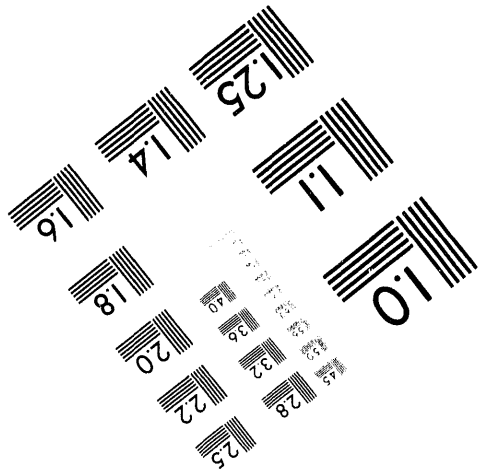
301/587-8202











MANUFACTURED TO AIIM STANDARDS
BY APPLIED IMAGE, INC.

1 of 1

CONF-9404137--4

Stochastic Approximation Methods for Fusion-Rule Estimation in Multiple Sensor Systems †

Nageswara S.V. Rao
Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6364
nrao@plato.epm.ornl.gov

To be presented at
Twelfth Symposium on Energy Engineering Sciences, Argonne National Laboratory, Argonne, IL, April 27-29, 1994.

---

†Research sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, of the U.S. Department of Energy, under Contract No. DE-AC05-840R21400 with Martin Marietta Energy Systems, Inc.

STOCHASTIC APPROXIMATION METHODS FOR FUSION RULE ESTIMATION IN MULTIPLE SENSOR SYSTEMS

Nageswara S. V. Rao
Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6364
nrao@plato.epm.ornl.gov

ABSTRACT

A system of N sensors $S_1, S_2, \ldots, S_N$ is considered: corresponding to an object with parameter $x \in \Re^d$, sensor S_i yields output $y^{(i)} \in \Re^d$ according to an unknown probability distribution $p_i(y^{(i)}|x)$. A training l-sample $(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)$ is given where $y_i = (y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(N)})$ and $y_i^{(j)}$ is the output of S_j in response to input x_i. The problem is to estimate a fusion rule $f : \Re^{Nd} \mapsto \Re^d$, based on the sample, such that the expected square error

$$I(f) = \int [x - f(y^{(1)}, y^{(2)}, \ldots, y^{(N)})]^2 p(y^{(1)}, y^{(2)}, \ldots, y^{(N)}|x) p(x) dy^{(1)} dy^{(2)} \ldots dy^{(N)} dx$$

is to be minimized over a family of fusion rules Λ based on the given l-sample. Let $f_* \in \Lambda$ minimize $I(f)$; f_* cannot be computed since the underlying probability distributions are unknown. Three stochastic approximation methods are presented to compute $\hat{f}$, such that under suitable conditions, for sufficiently large sample,

$$P[I(\hat{f}) - I(f_*) > \epsilon] < \delta$$

for arbitrarily specified $\epsilon > 0$ and δ, $0 < \delta < 1$. The three methods are based on Robbins-Monro style algorithms, empirical risk minimization, and regression estimation algorithms.

INTRODUCTION

There are fundamental limitations on the capabilities of single sensor systems in a number of application areas such as robotics (see Abidi and Gonzalez [1]). Diverse

and/or similar information from different sensors can often be used to overcome the limitations of a single sensor. In some systems, similar sensors are employed for fault tolerance, and in several others the task requires information from disparate sensors. The problem of obtaining a fusion rule to combine the outputs from a system of sensors has been the focus of extensive research over the past decades. A number of issues related to this problem have been studied under the framework of distributed sensor networks (see Iyengar et al. [2] and references therein). We consider a generic formulation of the problem of inferring a fusion rule using training examples, where the errors introduced by various individual sensors are unknown and not controllable, e.g. a robot system equipped with sensors. Here, the system is available, and the fusion rule for it has to be obtained; this aspect has to be contrasted with the general areas of team decision problems (e.g. Radner [3]) and distributed detection (e.g. Tsitsiklis and Athans [4]) where the individual elements as well as the fuser are to be designed to achieve an overall goal. If the sensor error distributions are known, several cases of this problem have been solved (Thomopoulos [5]), typically, by maximizing the a posteriori probabilities of hypotheses under a suitable probabilistic model.

Consider a system of N sensors $S_1, S_2, \ldots, S_N$ such that corresponding to an object with parameter $x \in \Re^d$, sensor S_i yields output $y^{(i)} \in \Re^d$ according to an unknown probability distribution $p_i(y^{(i)}|x)$. A training l-sample $(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l)$ is given where $y_i = (y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(N)})$ and $y_i^{(j)}$ is the output of S_j in response to input x_i. The problem is to estimate a fusion rule $f : \Re^{Nd} \mapsto \Re^d$, based on the sample, such that $f(y^{(1)}, y^{(2)}, \ldots, y^{(N)})$ "closely" approximates x. More precisely, we consider the expected square error

$$I(f) = \int [x - f(y^{(1)}, y^{(2)}, \ldots, y^{(N)})]^2 p(y^{(1)}, y^{(2)}, \ldots, y^{(N)}|x) p(x) dy^{(1)} dy^{(2)} \ldots dy^{(N)} dx \tag{1}$$

which is to be minimized over a family of fusion rules Λ based on the given l-sample. Let $f_* \in \Lambda$ minimize $I(f)$. In general, f_* cannot be computed since the underlying probability distributions are unknown. Furthermore, since no restrictions are placed on the underlying distributions, it will not be possible to infer f_* (with probability one) based on a finite sample. We obtain conditions under which an approximation $\hat{f}$ to f_* can be computed such that for a sufficiently large sample we have

$$P[I(\hat{f}) - I(f_*) > \epsilon] < \delta \tag{2}$$

for arbitrarily specified $\epsilon > 0$ and δ, $0 < \delta < 1$. Thus "error" due to $\hat{f}$ is bounded by an arbitrarily specified precision ϵ with arbitrarily specified confidence $1 - \delta$ given sufficiently large sample.

First issue in the computation of an approximation of f_* is the finite representation of elements of Λ. Based on the recent density results (e.g. Cybenko [6], Leshno et al. [7]) we identify several approximations for representing $f \in \Lambda$ as a feedforward network of sigmoid functions, radial basis functions, wavelets, and non-polynomial functions. Second issue is an algorithm to compute an approximation to f_*. We present three types of stochastic approximation algorithms (applicable under different conditions):

(a) **Robbins-Monro Style Algorithms:** The problem (1) is solved by direct stochastic approximation methods of Rao at al. [8] based on network representations of the fusion function.

(b) **Empirical Risk Minimization Methods:** Let the empirical estimate

$$I_{emp}(f) = \frac{1}{l}\sum_{i=1}^{l}[x_i - f(y_i^{(1)}, y_i^{(2)}, \ldots, y_i^{(N)})]^2 \tag{3}$$

be minimized by $f = f_{emp} \in \Lambda$. We compute an approximation to f_{emp} and then use the proximity results between f_* and f_{emp}. This method combines the empirical risk minimization results of Vapnik [9] with the finite sample results of Aizerman et al. [10].

(c) **Regression Estimation Algorithms:** The intimate relationship between the problem (1) and regression estimation is first employed; then the stochastic approximation methods of Revesz [11] based on kernels for regression estimation are employed to compute $\hat{f}$.

The convergence properties and the expressions for required sample sizes of these algorithms depend on different technical conditions which must be matched with the application scenario at hand.

The organization of this paper is as follows. In Section 2, the finite representations methods are discussed. A solution method to solve (1) based on the empirical risk minimization methods of Vapnik [9] are described in Section 3. In Section 4, stochastic approximation algorithms are described to compute $\hat{f}$ in (2).

NETWORK APPROXIMATIONS

We now briefly discuss some existing methods for finitely representable hypothesis classes, where $f \in \Lambda$ is represented by $w = (w_1, w_2, \ldots, w_M)$. The hypothesis classes

that are too numerous to be so represented are unlikely to be of use in practical implementations. A general architecture of a multilayer feedforward network consists of an input layer with Nd units and output layer with d units, and one or more hidden layers; here we consider $d = 1$ and single hidden layer. The hidden unit j has a weight vector $b_j \in \Re^{Nd}$ and a threshold $t_j \in \Re$. The output of the jth hidden unit is $\sigma(b_j^T y - t_j)$, where $y = (y^{(1)}, y^{(2)}, \ldots, y^{(d)})$ is the input vector, $b_j^T y$ denotes the scalar product, and $\sigma : \Re \mapsto \Re$ is called an *activation function.* The output of the output node is given by

$$h(w, y) = \sum_{j=1}^{m} a_j \sigma(b_j^T y - t_j)$$

where $a_i \in \Re$, $a = (a_1, a_2, \ldots, a_m)$, m is the number of units in the hidden layer, and w is the *parameter vector* of the network which consists of a, $b_1, b_2, \ldots, b_m$ and $t_1, t_1, \ldots, t_m$. Then $h(w, x)$ is called the *output* of the network with the *weight vector* w. The feedforward networks are found to be very useful in approximating fairly general classes of functions. A collection of some of the recently studied classes is as follows:

(a) **Feedforward neural networks:** Each $f : \Re^{Nd} \mapsto \Re$ is approximated by an artificial neural network with at least one hidden layer and with a finite number of nodes. As shown in Cybenko [6], such networks can approximate continuous functions with arbitrary levels of precision (see also Barron [12]). Here w of f corresponds the connection weight vector. The problem of computing f_{emp} is NP-hard in general. An approximation to f_{emp} can be computed using the well-known backpropagation algorithm (Werbos [13] and Rumelhart et al. [14]). Convergence properties of such algorithm have been studied by White [15] and Nedeljkovic [16].

(b) **Radial basis functions:** The radial basis networks with suitably chosen non-linear hidden layers (Broomhead and Lowe [17], Chen et al. [18]) can be used in the computation of f_{emp}; also there are a number of learning algorithms that can be applied in this case.

(c) **Wavelet-based expansion:** Zhang and Benveniste, [19] proposed networks of wavelets (in a manner analogous to neural networks) which can approximate arbitrary continuous maps; each network is characterized by a finite real vector that corresponds to the dilation and translation operations. They also propose an algorithm similar to the back propagation algorithm that can be used to compute an approximation to f_{emp}.

(d) **Networks of non-polynomial Units:** In a general treatment, Leshno et al. [7] showed that finite networks of non-polynomial units can be used to approximate the arbitrary continuous maps. Although no algorithms to compute the required connection weights are available, backpropagation style algorithms can be designed in several cases.

The essence of the above approximation (density) results is that fairly general classes of Λ can be represented (within a precision value) by a finite vector representing the connection weights of a feedforward network consisting of suitable non-polynomial units.

EMPIRICAL RISK MINIMIZATION METHODS

For family $\{A_\gamma\}_{\gamma\in\Gamma}$, $A_\gamma \subseteq A$, and for a finite set $\{a_1, a_2, \ldots, a_n\} \subseteq A$ we define:

$$\Pi_{\{A_\gamma\}}(\{a_1, a_2, \ldots, a_n\}) = \{\{a_1, a_2, \ldots, a_n\} \cap A_\gamma\}_{\gamma\in\Gamma},$$

$$\Pi_{\{A_\gamma\}}(n) = \max_{a_1, a_2, \ldots, a_n} |\Pi_{\{A_\gamma\}}(\{a_1, a_2, \ldots, a_n\})|.$$

The following is critical identity established in [9].

$$\Pi_{\{A_\gamma\}}(n) = \begin{cases} 2^n & \text{if } n \le h \\ < 1.5\frac{n^h}{h!} & \text{if } n > h \end{cases}$$

Notice that for a fixed h, the right hand side increases exponentially with n until it reaches h and then varies as a polynomial in n with fixed power h. This quantity h is called the *Vapnik-Chervonenkis dimension* of the family of sets A_γ; it can also be alternatively defined as the largest size h of a set $\{a_1, a_2, \ldots, a_n\} \subseteq A$ that can be subdivided in all possible ways into two classes by means of sets A_γ.

For a set of functions, the *capacity* is defined as the largest number h of pairs (x_i, y_i) that can be subdivided in all possible ways into two classes by means of rules of the form

$$\{\Theta[(x - f(y))^2 + \beta]\}_{f,\beta}$$

where

$$\Theta(z) = \begin{cases} 1 & \text{if } z \ge 0 \\ 0 & \text{f } z < 0 \end{cases}$$

Formally, the capacity of Λ is the Vapnik-Chervonenkis dimension of the set of indicator functions

$$\{\Theta[(x - f(y))^2 + \beta]\}_{(f,\beta)\in\Lambda\times\Re}.$$

To solve for a rule $f_* \in \Lambda$ that minimizes the expected error in (1), we instead minimize the empirical error in (2) (with $\hat{f}$ replaced by f_{emp}) to obtain a best empirical estimate f_{emp}. The closeness of f_{emp} to f_* is specified by the parameters *precision* ϵ and *confidence* δ in condition (2) referred to as the (ϵ, δ)-condition. In order to ensure the (ϵ, δ)-condition, two types of conditions are to be satisfied [9]: (a) the capacity of $\{f_\alpha\}_{\alpha\in\Lambda}$ must be bounded: and (b) the error $I(.)$ must be bounded.

Theorem 1 *Consider that the error is bounded as* $\sup_{x,y,f}(x - f(y))^2 \leq \tau$.

(i) Then given l examples, we have

$$P[I(f_{emp}) - I(f_*) \geq 2\tau\kappa] \leq 9\frac{(2l)^h}{h!}e^{-\kappa^2 l/4}.$$

(ii) If the hypothesis space is finite in that $\Lambda = \{f_1(y), f_2(y), \ldots f_P(y)\}$. Then given l examples, we have

$$P\left[I(f_{emp}) - I(f_*) > 2\tau\kappa\right] < 18Ple^{-\kappa^2 l/4}.$$

The Parts (i) and (ii) of this theorem directly follow from Theorem 7.1 and 7.3 of Vapnik [9] respectively. Similar results can be shown under the conditions of bounded error, and simpler solution conditions (see Rao [20]). Note that the results of Theorem 1 are mainly existential in nature in that they do not yield computational methods to obtain the required f_{emp}.

STOCHASTIC APPROXIMATION METHODS

The traditional results of stochastic approximation algorithms typically deal with convergence in an asymptotic sense; but, the problem (1) calls for finite sample results. We first discuss methods based on empirical estimation using the algorithms of Rao et al. [8] and the potential function methods developed in early sixties by Aizerman et al. [10]. Then we illustrate that the well-known kernel methods can be used for the fusion rule estimation using the methods of Revesz [11]. The convergence results of the

above algorithms are valid under different conditions. For simplicity of presentation, we assume that $d = 1$, i.e. x is a scalar.

Robbins-Monro Type Algorithms

We first develop conditions required to implement a stochastic approximation algorithm for fusion rule estimation problem using the formulations based on the results of Cybenko [6] (several generalizations of this method can be found in Rao et al. [8]).

Condition 1 *Let Y be the the Nd-dimensional unit cube. For a fixed m, the set of functions of the form $h(w, y) = \sum_{j=1}^{m} a_j \sigma(b_j^T y - t_j)$ approximate the set of fusion functions Λ, where $M = (n+2)m$ and w is the parameter vector with components a_i, t_j and b_j^T: for each $f \in \Lambda$ and $\varepsilon > 0$, there exists some $h(w, .)$ such that $|x - h(w, y)| < \varepsilon$ for all $y \in D \subseteq Y$ such that Lebesgue measure of D is at least $1 - \varepsilon$.*

The basic structure of the algorithm is as follows [8]: the function at the $(n+1)$th stage, represented by w_{n+1}, is computed from w_n using the algorithm

$$w_{n+1} = w_n + \Gamma_n[|h(w_n, y_n) - x_n|] \qquad (A.1)$$

where $w_n, w_{n+1} \in \Re^M$, each component of $\Gamma_n \in \Re^M$ consists of scalar γ_n (same in all components of Γ_n) called the *step size*, and (x_n, y_n) is the nth example.

Let $\mu(w) = \int |h(w, y) - x| P(x, y) dx dy$ denote the expected error made by the hypothesis with weight vector w. Notice that $\mu(w) \geq 0$, and $\inf_w \mu(w) = 0$ by Condition 1. Also $E[(|h(w_n, y_n) - x_n|)|w_n] = \mu(w_n)$. In order to ensure the convergence of the algorithm (A.1), we require the following conditions on the probability measure generated by $P(x, y)$. Notice that Condition 1 applies to Λ and the following Conditions 2 and 3 apply to class from which $P(x, y)$ is chosen.

Condition 2 *Let $\mu(w)$ be differentiable and its gradient satisfy the following Lipschitz condition: for all $u, v \in \Re^N$, there exists a positive constant L such that $|| \nabla \mu(u) - \nabla \mu(v)|| \leq L||u - v||$.*

Condition 3 *There exists a scalar θ such that for any w and M-dimensional vector $\Psi(w) = (\mu(w), \ldots \mu(w))^T$, we have*

$$\nabla \mu(w)^T \Psi(w) \geq \theta \mu(w),$$

where $\nabla \mu(w)^T = \left(\frac{\partial \mu(w)}{\partial w_1}, \ldots, \frac{\partial \mu(w)}{\partial w_M}\right)$. This condition implies $\nabla \mu(w)^T \underline{1} \geq \theta$, every where except at $\mu(w) = 0$, where $\underline{1}$ is a column vector of all 1's.

Theorem 2 *Under the conditions (1)-(3), for* $\hat{f}(.) = h(., w_l)$ *we have*

$$P[I(\hat{f}) - I(f_*) < \epsilon] > 1 - \delta$$

under either of the following cases:

(i) For $\gamma_n = \gamma$, $\gamma < 1/\theta$, *and sufficiently large sample of size* l *given by*

$$l = \ln\left(\frac{1 - \frac{LM\gamma}{2\theta}}{\delta\epsilon - \frac{LM\gamma}{2\theta}}\right) / \ln\left(\frac{1}{1 - \gamma\theta}\right).$$

(ii) For $\gamma_n = \gamma/(n+1)$ *for* $n \geq 0$, *and* $\gamma < 1/\theta$ *and sufficiently large sample of size* l *given by*

$$l = \left[\frac{1 - \frac{L\gamma M}{2\theta}}{\epsilon\delta - \frac{L\gamma M}{2\theta}}\right]^{\frac{2}{\theta\gamma}}.$$

This theorem follows from Corollary 1 of [8]; several general results along these lines can be found in [8].

Empirical Risk Minimization

We compute an approximation $\hat{f}_{emp}$ to f_{emp} such that

$$P[I(\hat{f}_{emp}) - I(f_{emp}) > \epsilon_{emp}] < \delta$$

which is combined with the Theorem 1 (with $\epsilon = 2\tau\kappa$, and $\delta = 9\frac{(2l)^h}{h!}e^{-\kappa^2 l/4}$) to obtain

$$P[I(\hat{f}_{emp}) - I(f_*) > \epsilon + \epsilon_{emp}] < \delta. \tag{4}$$

We now consider the algorithm based on the potential functions of [10] (see also [21]).

Condition 4 *For a fixed* M, *any function* $f \in \Lambda$ *be of the form* $f(y) = h(w, y) = \sum_{j=1}^{M} a_j\sigma_j(y)$, *where* w *is the parameter vector with components* a_i *such that*

$$\int_Y f^2(y)P(y)dy > 0.$$

Now let

$$K(y,z) = \sum_{i=1}^{M} \lambda_i^2 \sigma_i(y)\sigma_i(z).$$

We now consider an algorithm that explicitly takes into account the network form of the fusion rule:

$$f_{n+1}(y) = f_n(y) + \frac{1}{\Theta}[x_n - f_n(y_n)]K(y, y_n) \qquad (A.2)$$

such that

$$\Theta > \frac{1}{2}\max_{y\in Y} K(y,y)$$

Notice that under Condition 4. the above algorithm can be implemented by using weight vectors ($a = (a_1, a_2, \ldots, a_M)$) as in (A.1).

Theorem 3 *Under the condition (4), and* $\delta = 9\,e^{-\epsilon^2 l/16\tau^2}$ *for* $\hat{f}(.) = f_l$ *we have*

$$P[I(\hat{f}_l) - I(f_*) < \epsilon + \epsilon_{emp}] > 1 - \delta$$

for sufficiently large sample size l *such that* $\epsilon_{emp} = \frac{c}{\delta}(1 - ra)^l$, *where* c *and* r *are constants.*

This theorem follows from Theorem 3 of [10] and Chebyshev's inequality as in Theorem 2 of [8].

Regression Algorithms

Consider (x, y) distributed according to $P(x, y)$ where y is the sensor output corresponding to the feature x. We additionally assume that y is real and the multidimensional case can be handled by standard methods [11]. For a given value of y, let x_y be a random variable distributed according to $P(x|y)$. Now we show that the regression function $r(y) = E(x|y) = E(x_y)$ minimizes (1) if $r(.) \in \Lambda$ and is closest to the minima in L^2-norm otherwise. To see this, consider $\Delta f(y) = f(y) - r(y)$. Then it can be shown [9] that

$$I(f) = \int [x - r(y)]^2 P(x,y)dxdy + \int [f(y) - r(y)]^2 P(y)dy$$

which shows that the minimum of $I(.)$ is achieved at the regression function since the first term is independent of f.

The applicability of the stochastic approximation methods based on kernel functions for regression estimation problem is established by Revesz [11]. Now consider the algorithm

$$f_{n+1}(y) = f_n(y) + \frac{1}{(n+1)a_{n+1}} K\left(\frac{y - y_{n+1}}{a_{n+1}}\right)(x_{n+1} - f_n(y)) \qquad (A.3)$$

where $a_n = n^{-\alpha}$ and

$$K(y) = \begin{cases} 1 & \text{if } |y| \leq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

Theorem 4 *Suppose that (i) $r(y)$ is measurable and bounded, (ii) y has an absolutely continuous distribution with density $p(y)$ for which $1/2 \leq p(y) \leq \infty$, (iii) x is bounded with probability 1. For $\hat{f}(.) = f_l$, and $r(y) \in \Lambda$ we have*

$$P[I(\hat{f}) - I(f_*) > \epsilon] < \delta$$

for sufficiently large sample size l such that

$$l = \left(\frac{1}{C}\ln(1/\delta)\right)^{\frac{1}{1-\rho}}$$

where C is a function of only ϵ and ρ.

This theorem follows from Theorem A of [11].

CONCLUSIONS

We considered a general computational framework for a system of multiple sensors with unknown noise characteristics. Here the system is available so that readings corresponding to objects of known parameters can be obtained. In this context, we addressed the problem of computing a fusion rule based on a set of training data with only a limited assumptions on the noise. The proposed methods are applicable only if suitable samples are available. If the underlying probabilities are available, then other methods are more likely to be effective. Future research directions include (a) identification of classes of Λ based on the specific properties of the system (b) investigation of practical methods that employ problem specific information to improve the runtime of the algorithms. Presently, methods based on stochastic approximation and potential functions are being tested on practical problems of fusing data from sonar arrays. The former has been successfully applied to train a recognition system for glassware based on laser range images [22].

ACKNOWLEDGEMENTS

This research is sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, of the U.S. Department of Energy, under Contract No. DE-AC05-840R21400 with Martin Marietta Energy Systems, Inc.

References REFERENCES

[1] M. A. Abidi and R. C. Gonzalez, editors. *Data Fusion in Robotics and Machine Intelligence*. Academic Press, New York, 1992.

[2] S. S. Iyengar, R. L. Kashyap, and R. B. Madan. Special issues on distributed sensors networks. *IEEE Trans. Syst. Man Cybernetics*, 21(5):1027–1031, 1991.

[3] R. Radner. Team decision problems. *Annals of Mathematical Statistics*, 33:857–881, 1962.

[4] J. N. Tsitsiklis and M. Athans. On the complexity of decentralized decision making and detection problems. *IEEE Transactions on Automatic Control*, AC-30(5):440–446, 1985.

[5] S. C. A. Thomopoulos. Theories in distributed decision fusion. In A. H. Lewis and H. E. Stephanou, editors, *Distributed Intelligence Systems*, pages 195–200. Pergamon Press, New York, 1992.

[6] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Contols, Signals, and Systems*, 2:303–314, 1989.

[7] M. Leshno, V. Ya. Lin, A. Pinkus, and S. Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6:861–867, 1993.

[8] N. S. V. Rao, V. R. R. Uppuluri, and E. M. Oblow. Stochastic approximation algorithms for classes PAC learning problem. In *Proceedings of 1994 IEEE Conf. on Neural Networks*. 1994. to appear.

[9] V. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York, 1982.

[10] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer. *Extrapolative problems in automatic control and method of potential functions*, volume 87 of *American Mathematical Society Translations*, pages 281–303. 1970.

[11] P. Revesz. How to apply the method of stocastic approximation in the non-parametric estimation of a regression function. *Math. Operationsforsch. Statist., Ser. Statistics*, 8(1):119–126, 1977.

[12] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):931–945, 1993.

[13] P. J. Werbos. Backpropagation through time:What it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[14] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and J. L. McClelland, editors, *Parallel and Distributed Processing: Explorations in the Microstructures of Cognition*. MIT Press, Cambridge, MA, 1986.

[15] H. White. *Artificial Neural Networks*. Blackwell Publishers, Cambridge, Massachusetts, 1992.

[16] V. Nedeljkovic. A novel multilayer neural networks training algorithm that minimizes the probability of classification error. *IEEE Transactions on Neural Networks*, 4(4):650–659, 1993.

[17] D. S. Broomhead and D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, 2:321–355, 1988.

[18] S. Chen, S. A. Billings, and P. M. Grant. Recursive hybrid algorithm for non-linear system identification using radial basis function networks. *International Journal of Control*, 55(5):1051–1070, 1992.

[19] Q. Zhang and A. Benveniste. Wavelet networks. *IEEE Transactions on Neural Networks*, 3(6):889–898, 1992.

[20] N. S. V. Rao. Fusion rule estimation in multiple sensor systems with unknown noise distributions. In *Proc. Indo-US Workshop on Parallel and Distributed Signal and Image Integration Problems*. 1993. to appear.

[21] L. Fisher and S. J. Yakowitz. Uniform convergence of the potential function algorithm. *SIAM J. Control and Optimization*, 14(1):95–103, 1976.

[22] S. Toemoe, N. S. V. Rao, and R. C. Mann. A learning-based method to recognize and localize glassware using laser range images. 1994. under preparation.

DATE
FILMED
8/8/94
END