Learning a Decision Boundary from Stochastic Examples: Incremental Algorithms with and without Queries

Yoshiyuki Kabashima*
Shigeru Shinomoto
Department of Physics, Kyoto University, Kyoto 606, Japan

Even if it is not possible to reproduce a target input–output relation, a learning machine should be able to minimize the probability of making errors. A practical learning algorithm should also be simple enough to go without memorizing example data, if possible. Incremental algorithms such as error backpropagation satisfy this requirement. We propose incremental algorithms that provide fast convergence of the machine parameter \( \theta \) to its optimal choice \( \theta_0 \) with respect to the number of examples \( t \). We will consider the binary choice model whose target relation has a blurred boundary and the machine whose parameter \( \theta \) specifies a decision boundary to make the output prediction. The question we wish to address here is how fast \( \theta \) can approach \( \theta_0 \), depending upon whether in the learning stage the machine can specify inputs as queries to the target relation, or the inputs are drawn from a certain distribution. If queries are permitted, the machine can achieve the fastest convergence, \( (\theta - \theta_0)^2 \sim O(t^{-1}) \). If not, \( O(t^{-1}) \) convergence is generally not attainable. For learning without queries, we showed in a previous paper that the error minimum algorithm exhibits a slow convergence \( (\theta - \theta_0)^2 \sim O(t^{-2/3}) \). We propose here a practical algorithm that provides a rather fast convergence, \( O(t^{-4/5}) \). It is possible to further accelerate the convergence by using more elaborate algorithms. The fastest convergence turned out to be \( O((\ln t)^2 t^{-1}) \). This scaling is considered optimal among possible algorithms, and is not due to the incremental nature of our algorithm.

1 Introduction

An ideal objective of machine learning is to identify a target input–output relation. Even if all the examples can be reproduced by adjusting machine parameters, the relation acquired via examples is generally not identical to the target relation, and the central issue is then the probability of error in the prediction of a novel example (Valiant 1984; Baum and Haussler 1989; Levin et al. 1990; Amari et al. 1992; Sompolinsky and Barkai 1993).

*Present address: Department of Physics, Nara Women’s University, Nara 630, Japan.
However, in most of the practical applications of learning algorithms, it is still hard to reproduce all the examples drawn from a target relation, so that it is definite already in the learning stage that the target relation is not reproducible by the machine (Rumelhart et al. 1986; Sejnowski and Rosenberg 1987). The objective of learning in this case is not necessarily to look for the target relation, but just to obtain the best output prediction for an individual input. When considering the binary choice model whose target relation is stochastic, it is obvious that the best prediction is to choose an output that appears more often than the alternative. Thus the learning machine has to partition the input space so as to minimize the prediction error.

In a previous paper, we discussed the error minimum and the maximum-likelihood algorithms as strategies to find a decision boundary that partitions the input space (Kabashima and Shinomoto 1992). In the error minimum algorithm, a parameter or a set of parameters $\theta$ is readjusted so that the controlled decision boundary makes the minimum number of empirical errors. We found in this case that the parameter $\theta$ converges to the optimal choice $\theta_0$, rather slowly, $(\theta - \theta_0)^2 \sim O(t^{-2/3})$. Though the anomalous fractional exponent $2/3$ is theoretically interesting, the error minimum algorithm cannot be called efficient due to this exponent. We noticed that problems with a similar origin have been discussed independently in various fields of science: the time scaling of the intervals of shocks observed in the Burgers equation (Burgers 1974; Karder et al. 1986), mathematical economics (Manski 1975; Kim and Pollard 1990; Kawanabe and Amari 1993), pattern recognition (Kohonen 1989), and statistical decision theory (Haussler 1991). Apart from the asymptotic scaling, Barkai et al. (1993) studied the problem specific to the high dimensional error minimum algorithm. They estimated the number of examples needed to attain a significant inference using a machine with a large number of parameters.

In the maximum-likelihood algorithm, a probability distribution function is selected out of a family of hypothetical functions so as to maximize the (log) likelihood for given data. This can also be utilized for finding a decision boundary. The decision boundary is determined as a hypersurface on which the hypothetical probabilities for alternative classes balance each other. If the true distribution happens to be included in a family of hypothetical distribution functions, the decision boundary $\theta$ eventually converges to the optimal choice $\theta_0$. In this case, we obtain rapid convergence, $(\theta - \theta_0)^2 \sim O(t^{-1})$. If the true distribution is not available, however, the decision boundary does not converge to the optimal choice. The naive maximum-likelihood algorithm is thus not efficient either.

To reproduce an arbitrary probability distribution function, we have to prepare a family of functions with an infinite number of parameters. When the number of parameters is finite, the determination of the boundary must have some error. If on the one hand we prepare a number of parameters to approximate any distribution function and to obtain the
asymptotic dependence of $O(t^{-1})$ to a certain precision, then the prefactor of $t^{-1}$ will become large and the asymptotic regime of $O(t^{-1})$ will not be reached within a practical number of examples. For a given number of examples, there must be an optimal number of parameters. Using the strategy to change the number of parameters depending on the number of examples, we can obtain a fairly rapid convergence of the decision boundary $\theta$ to the optimal choice $\theta_o$. The question we wish to address here is how fast $(\theta - \theta_o)^2$ converges to zero. The $O(t^{-1})$ convergence is not attainable. We will propose a practical algorithm that provides a rather fast convergence, $(\theta - \theta_o)^2 \sim O(t^{-4/5})$. A more elaborate algorithm can provide more rapid convergence, $O(t^{-2p(2p+1)})$, $p = 2.3, \ldots$. Although larger $p$ appears preferable, it has to pay the price of a larger prefactor. The best choice of $p$ depending upon the number of examples gives the fastest convergence, which turned out to be $O((\ln t)^2 t^{-1})$.

In comparison with passive learning, which leaves the inputs to be drawn from a certain distribution, there must be an advantage of introducing the freedom of queries into learning; the machine can specify inputs to inquire about respective outputs. Seung et al. (1992) and Freund et al. (1992) showed that the information gain per example remains finite even in the limit $t \to \infty$ if queries are allowed, while it decays as $t^{-1}$ if not. We propose here a practical algorithm that gives the fastest convergence $(\theta - \theta_o)^2 \sim O(t^{-1})$. The use of queries in neural networks was discussed by Baum (1991), in which for a realizable target relation an exponentially fast convergence is proved under some conditions. This exponential convergence is due to the deterministic nature of the target relation, and is in principle not attainable for our stochastic relation (Cramér 1946).

For the algorithm to be practical, there is another requirement in addition to the problems of convergence. The learning algorithm should be simple enough to work without storing the example data, if possible. We will introduce incremental algorithms that do not require memory of the previous examples. This incremental nature is extremely important for a practical algorithm, as it greatly reduces the computational burden. The popularity of the error backpropagation algorithm is due to its being of this nature. The backpropagation algorithm, however, can be considered as a kind of maximum-likelihood algorithm. Instead, what we introduce here is not a naive maximum-likelihood algorithm that can give a wrong decision boundary, but an efficient and practical algorithm for finding the optimal decision boundary for output prediction.

First, we are going to discuss the problem of dividing a one-dimensional space that has a blurred boundary (see Fig. 1). Every example consists of the real input $x \in [a, b]$ and the binary output $s = \pm 1$. In learning without queries, $t$ inputs $x$ are drawn independently from some nonsingular distribution $p(x)$. On the other hand, the machine can specify inputs $x$ in learning with queries. The target relation is stochastic, namely for every input $x$, the output $s$ is drawn from a certain conditional probability distribution $p(s | x)$. We will assume that $p(s = +1 | x) = 1 - p(s = -1 | x)$.
as a function of $x$ is infinitely differentiable and monotonically increasing. The learning machine knows nothing other than that. Even if the perfect knowledge of $p(s \mid x)$ is acquired, it is impossible to give a perfect prediction of the individual output for every input. The best prediction for the individual examples is attained if we separate the input space into positive and negative regions depending on $p(s = +1 \mid x) > p(s = -1 \mid x)$ and $p(s = +1 \mid x) < p(s = -1 \mid x)$. Learning a decision boundary does not necessarily require complete knowledge of the conditional probability distribution $p(s \mid x)$, but it suffices to find a (directed) boundary at which alternative probabilities balance. As $p(s = +1 \mid x)$ is a monotonically increasing function of $x$ in this one-dimensional case, the optimal decision boundary $x = \theta_o$ is a single point that satisfies $p(s = +1 \mid x) = p(s = -1 \mid x) \mid_{x=\theta_o}$. We will first show the incremental algorithm that enables the fastest convergence $(\theta - \theta_o)^2 \sim O(t^{-1})$ with queries allowed, and then discuss the efficient algorithms for learning without queries.

Second, we are going to discuss the higher dimensional case to see whether these scaling forms obtained in the one-dimensional case are critically dependent on the dimension of the parameter space. The higher dimensionality affects the convergence by a factor, but presumably does not deteriorate the scaling form, though we have not succeeded in finding a concrete algorithm to accelerate the convergence up to $O((\ln t)^2 t^{-1})$.

2 Learning with Queries

We consider first an active learning process in which the learning machine can specify inputs as queries to a target relation. For every input $x \in [a, b]$, ...
an output $s = \pm 1$ is drawn from the conditional probability $p(s \mid x)$. The nature of getting $s = \pm 1$ consists of two factors: the mean and the fluctuation. If for instance $p(s = +1 \mid x) > p(s = -1 \mid x)$ at some point $x$, then there is a tendency to find $s = +1$ more often than $s = -1$ at this point. However, if the alternative probabilities do not have a significant difference, a number of examples are needed before we are sure of which is larger.

An appropriate series of queries brings about a fast convergence of the decision boundary $x = \theta$ to its optimal choice $\theta_0$. Provided that $p(s = +1 \mid x)$ is a monotonically increasing function of $x$, it will be preferable on average to push back the hypothetical boundary $x = \theta$ if one gets $s = +1$ at the boundary, and push it forward otherwise. We propose the learning algorithm as follows. Let $\theta_t$ denote the hypothetical boundary determined via $t$ examples, and assume that the query given at this point is $x = \theta_t$. Given the output $s$ for this input $x = \theta_t$, the machine moves the parameter $\theta_t$ to

$$\theta_{t+1} = \theta_t - s\alpha_t, \quad (2.1)$$

where $\alpha_t (> 0)$ is a step size that can depend on $t$.

We assume that the conditional probability $p(s = +1 \mid x)$ can be expanded around $x = \theta_0$ as

$$p(s = +1 \mid x) = 1/2 + k_1(x - \theta_0) + k_2(x - \theta_0)^2 + \cdots \quad (2.2)$$

In the vicinity of $\theta_0$, the mean and the variance of $s = \pm 1$ are approximated as

$$\langle s \rangle_x \sim 2k_1(x - \theta_0), \quad \langle s^2 \rangle_x - \langle s \rangle_x^2 \sim 1 \quad (2.3)$$

where $\langle \cdots \rangle_x = \sum_{s=\pm 1} \cdots p(s \mid x)$. As the hypothetical boundary $x = \theta_t$ comes close to the optimal position $\theta_0$, the mean drift force toward the optimal boundary becomes weak, while its fluctuation remains large. If we keep $\alpha_t$ constant, $\theta_t$ is subject to both the drift force and fluctuation and will not converge to $\theta_0$.

It was proven (Robbins and Monro 1951; Kushner and Clark 1978) that $\theta_t$ strongly converges to its optimal value $\theta_0$ provided that

$$\sum_t \alpha_t = \infty, \quad \sum_t \alpha_t^2 < \infty \quad (2.4)$$

The $\alpha_t$ dependence of the way in which $\theta_t$ converges has not been studied in detail. We are going to investigate the convergence of $\theta_t$ by means of a physical interpretation of its dynamics. We found that 2.4 is just a sufficient condition for the convergence of $\theta_t$, and it is even possible to give a successful schedule $\{\alpha_t\}_t$, which is outside of the condition 2.4.

Owing to the coexistence of the drift force (the mean of $s$) and the fluctuation (the variance of $s$), equation 2.1 with equations 2.3 can be
interpreted as Brownian motion in a quadratic drift potential. These dynamics can be approximated by the Langevin equation,

$$dz/dt = \alpha[-2k_1z + \eta(t)]$$  \tag{2.5}

where $z = \theta_t - \theta_o$, $\alpha = \alpha_t$ is generally dependent on $t$, and $\eta(t)$ is white noise characterized by the statistical properties, $\langle \eta(t) \rangle = 0$, $\langle \eta(t)\eta(t') \rangle = \delta(t-t')$. The Fokker-Planck equation is an alternative description of stochastic dynamics,

$$\frac{\partial P(z,t)}{\partial t} = 2k_1\alpha \frac{\partial [zP(z,t)]}{\partial z} + \frac{\alpha^2}{2} \frac{\partial^2 P(z,t)}{\partial z^2}$$ \tag{2.6}

where $P(z,t)$ is the ensemble distribution of learning machines with parameters $z = \theta_t - \theta_o$ at the moment $t$. From equation 2.5 or 2.6 we are able to obtain the evolution equation of the mean square deviation $u = \langle z^2 \rangle = \langle (\theta_t - \theta_o)^2 \rangle$,

$$du/dt = -4k_1\alpha u + \alpha^2$$ \tag{2.7}

From this equation, we can find the optimal series of $\alpha_t$ for obtaining the fastest convergence of $u$. This is performed by minimizing the right-hand side of equation 2.7, which is obtained by adjusting $\alpha = 2k_1u$. This gives the solution $u(t) = 1/[4k_1(t + \text{const.})] \sim 1/(4k_1^2 t)$, and hence $\alpha \sim 1/(2k_1 t)$. This strategy is similar to what we employed to discuss the finite time scaling of energy in simulated annealing (Shinomoto and Kabashima 1991). The present model is in some sense similar to a thermodynamic system. The mean square deviation $u$ is proportional to $\alpha$ in equilibrium, so $\alpha$ corresponds to the “temperature” of the thermodynamic system. The feature specific to this model is that the drift potential is also proportionally dependent on $\alpha$.

To obtain the optimal learning schedule $\{\alpha_t\}_t$, one has to have a knowledge of $k_1$ and the mean square deviation $u(t)$. In a practical learning situation, a rough estimate of $k_1$ might be available, but the deviation $u(t)$ at the moment $t$ is unknown. Instead, we can fix a reasonable schedule $\{\alpha_t\}_t$ first that would give a fast convergence of the (unknown) deviation. By substituting the learning schedule $\alpha_t = A/t$, we can solve equation 2.7. The asymptotic form of the solution turns out to be

$$u(t) \sim A^2[(4k_1A - 1)t]^{-1}, \quad \text{for } A > 1/4k_1$$
$$\sim \ln t/t, \quad \text{for } A = 1/4k_1$$
$$\sim Ct^{-4k_1A}, \quad \text{for } 0 < A < 1/4k_1$$ \tag{2.8}

This result shows that the learning schedule $\alpha_t = 1/(2k_1 t)$ is optimal (see Fig. 2a), which is in agreement with the solution of the optimal strategy. All the learning schedules here satisfy the conventional condition for convergence 2.4. Though convergent, it is intriguing to see that the asymptotic form exhibits a qualitative deterioration in its exponent for $A < 1/4k_1$ (see Fig. 2b).
Figure 2: Asymptotic behavior of the mean square deviation $u(t) = \langle (\theta_t - \theta_0)^2 \rangle$ obtained with various values of $A$ for learning schedule $\alpha_t = A/t$. (a) Prefactor of the $u(t) \sim O(t^{-1})$ for the case $A > 1/4k_1$; (b) qualitative deterioration seen in the exponent of the asymptotic decay $u(t) \sim O(t^{1-\gamma})$, seen for $A < 1/4k_1$. The closed circles are the results of numerical experiments.

Next, we wish to see what happens if we change the learning schedule from $A/t$ to $A/t^\beta$. It is still easy to integrate equation 2.7, and the asymptotic form is

$$u(t) \sim \frac{A}{4k_1 t^\beta} + C \exp \left[ -\frac{4k_1 A}{1-\beta} (t^{1-\beta} - 1) \right]$$

(2.9)

The mean square deviation converges to zero, if $0 < \beta \leq 1$. Note that the learning schedules with $0 < \beta < 1/2$ are out of the conventional condition for convergence 2.4. The convergence for $0 < \beta < 1/2$ is not so surprising,
Learning a Decision Boundary

Figure 3: The mean square deviation \( u(t) = \langle (\theta_t - \theta_0)^2 \rangle \) vs. \( t \) for various learning schedules \( \alpha_t = A/t^\beta \), \( \beta = 0.0, 0.125, 0.25, \) and 0.5. The average is taken over 1000 sets of \( t \) examples for the target relation \( p(s = +1 | x) = x \). The mean square fit for \( u(t) \sim O(t^-\gamma) \), respectively, gives \( \gamma = -0.0081 \pm 0.0048, 0.1226 \pm 0.0045, 0.2534 \pm 0.0051 \), and 0.5271 \pm 0.0043. These are in good agreement with the theoretical asymptotes \( u(t) = A/4k_1t^{-2\beta} \) shown as lines in the figure.

and can be reasonably understood from a physical point of view. As described before, the parameter \( \alpha \) works as a kind of "temperature" in this system, and in equilibrium \( u(t) \) is proportional to \( \alpha \). If we reduce the temperature too rapidly, the ensemble of the systems does not attain the equilibrium distribution, and will be partially "frozen." This happens for \( \beta > 1 \), in which the second term of the right-hand side of equation 2.9 does not vanish even in the limit \( t \to \infty \). On the other hand, if one reduces the temperature slowly, the ensemble equilibrates almost every time, which makes the mean square deviation \( u \) proportional to \( \alpha_t \). This happens for \( 0 < \beta < 1 \), in which the first term of the right-hand side of equation 2.9 is dominant, implying \( u(t) \propto \alpha_t \). The result of numerical simulation is shown in Figure 3.
Smaller $\beta$ gives slower convergence, which is not preferable. On the other hand, the second term of equation 2.9, which represents a memory effect with respect to the initial condition, exhibits more rapid decay for smaller $\beta$. We are often faced with a learning situation in which the target relation itself depends on time. In such a case, the machine has to be sufficiently adaptive to the temporal change. Amari (1967) illustrated that learning with a fixed step size, which is similar to the case $\beta = 0$ in the present framework, is adaptive to a time varying target. As it is easy for small $\beta$ to erase the memory, we are then able to choose sufficiently small $A$ in order to obtain smaller mean square deviation. The number of examples required to obtain a certain precision for parameter estimation is not critically dependent on the choice of $\beta$. This argument with respect to the sample complexity was discussed by Kabashima and Shinomoto (1993).

3 Learning without Queries

If queries are not allowed, the machine has to make most of the information available from examples drawn from the distribution $p(s, x) = p(s | x)p(x)$. If there is a symmetry with respect to the inversion, $p(s, \theta_o + x) = p(-s, \theta_o - x)$ and we can assume this symmetry in advance, then most of the data from this joint probability can be utilized in determining the hypothetical boundary, and the machine can attain the fastest convergence of $O(t^{-1})$. There is, however, no symmetry in general. Due to the absence of symmetry, we have to prepare more in the inference, and this makes the convergence slower.

An easy way of utilizing an incremental algorithm similar to the preceding one is to prepare a window for accepting inputs. Let us assume a window of an interval $2\sigma_i$ centered at the hypothetical boundary $\theta_i$. The parameter $\theta_i$ is updated when the input $x$ falls in the window,

$$\theta_{i+1} = \theta_i - s\sigma_i / 2\tau_i. \text{ if } x \in [\theta_i - \tau_i, \theta_i + \tau_i]$$

and does not change otherwise. This algorithm is similar to the vector quantization procedure LVQ2, proposed by Kohonen (1989). In the original LVQ2, the window size is fixed, but we are going to control the window size so as to obtain the exact convergence.

We will hereafter assume that the probability distribution $p(x)$ is infinitely differentiable and is expanded around $x = \theta_o$ as

$$p(x) = p_o + h_1(x - \theta_o) + h_2(x - \theta_o)^2 + \cdots$$

The probability that the machine receives $s = +1$ for an input that falls in the window is obtained via

$$\int_{\theta_i - \tau_i}^{\theta_i + \tau_i} p(s = +1 | x)p(x) \, dx / \int_{\theta_i - \tau_i}^{\theta_i + \tau_i} p(x) \, dx$$

(3.3)
The Langevin equation for the corresponding dynamics 3.1 is given by

$$\frac{dz}{dt} = \alpha \left[ -2k_1z - \frac{2}{3} \left( k_2 + \frac{k_1h_1}{p_e} \right) \tau^2 + \sqrt{\frac{1}{2\tau\eta(t)}} \right]$$

(3.4)

where $z = \theta_i - \theta_o$, and $\eta(t)$ is the white noise characterized by $\langle \eta(t) \rangle = 0$ and $\langle \eta(t)\eta(t') \rangle = \delta(t - t')$. The second term on the right-hand side of equation 3.4 does not vanish even in the limit $z = 0$. This term is the origin of the systematic error due to the asymmetry in the joint distribution $p(s,x)$. To eliminate this systematic error, one has to shorten the interval $\tau_i$ itself. On the other hand, updates become infrequent as one narrows the window, and then the relative intensity of fluctuations increases by $(2\tau)^{-1/2}$. The trade-off between these tendencies determines the optimal choice of the window size.

We assume the learning schedule to be $\alpha_t = A/t$, and seek the optimal window schedule from among $\tau_i \sim O(t^{-\xi})$. From the Langevin equation 3.4 we obtain the evolution of the mean square deviation $u = \langle z^2 \rangle$,

$$u(t) \sim O(t^{-6/5}) + O(t^{-4/5})$$

(3.5)

Minimizing this with respect to $\xi$, we obtain the mean square deviation $u(t) \sim O(t^{-4/5})$ and the optimal window schedule $\tau_i \sim O(t^{-1/5})$. It is interesting to see that the same exponent $4/5$ for the convergence was obtained by Seung et al. (1992b) with respect to the learning process of the perceptron with binary weights. This is presumably a mere coincidence. Their scaling is exact in the thermodynamic limit, where the number of binary weight $N \rightarrow \infty$, the number of examples $t \rightarrow \infty$, and $N/t$ is finite, while our scaling is exact in the asymptotic limit $t \rightarrow \infty$ for a finite-dimensional system.

We had to throw away many examples that do not fall in the window in order to suppress the systematic error of $O(\tau^2)$ due to the possible asymmetry in the probability distribution $p(s,x)$. This is because the original algorithm is not elaborate enough to manage the convexity and higher order asymmetry of the probability distribution. A more elaborate algorithm must be able to utilize more data available. In the preceding model, we adopted a constant learning rate over the window. This is unnatural because an example from the central region of the window should be more influential than that from the surrounding region.

We are actually able to accelerate the convergence by using a more elaborate dynamics. The learning rule 3.1 has to be modified so that

$$\theta_{i+1} = \theta_i - \frac{s}{2\tau_i} w_p \left( \frac{x - \theta_i}{\tau_i} \right), \quad \text{if } x \in [\theta_i - \tau_i, \theta_i + \tau_i]$$

(3.6)

where $w_p(\cdot)$ is a kernel function to control the amount of movement depending on the position of an input $x$ relative to the present hypothetical boundary. The function $w_p(y)$ for $p \geq 2$ is defined over the interval
Figure 4: Kernel functions $w_p(y)$ for $p = 2, 4, \text{ and } 6$.

$y \in [-1, 1]$, and is assumed to satisfy the following conditions,

\[
\begin{align*}
\int w_p(y) \, dy &= 1 \\
\int w_p(y) y^k \, dy &= 0, \quad \text{for } k = 1, \ldots, p - 1 \\
\int w_p(y) y^p \, dy &\neq 0
\end{align*}
\]  

(3.7)

This set of conditions for the kernel function was introduced by Kawanabe and Amari (1993) in the context of a semiparametric estimation for the binary choice model. It is easy to obtain reasonable kernel functions by means of the Legendre polynomials. The kernel function is not determined uniquely by these conditions. In Figure 4, we plotted the simplest examples of $w_p(y)$ for even $p; 2, 4, \text{ and } 6$. The kernel function for odd $p$ has an additional arbitrary constant. By choosing a suitable constant, $w_p(y)$ for odd $p (= 2j + 1)$ can be the same as the one for even $p (= 2j)$. In agreement with our preceding discussion, the influence of every example depends on the position of the input $x$ relative to the present position of the hypothetical boundary. It is more interesting to see that the kernel can be negative for $p > 2$ due to these orthogonality conditions. Thus the resultant move for the positive example is not necessarily to the left, and vice versa.

Using this kernel function, we found that the optimal window size scales as $\tilde{\tau} \sim O(t^{-1/(2p+1)})$, which is slower than that of the original model.
for $p > 2$. The resultant asymptotic scaling of the mean square deviation turns out to be

$$u(t) \sim O(p^2 t^{-2p/(2p+1)}) \quad (3.8)$$

The same exponent $2p/(2p+1)$ was obtained by Barron and Cover (1991) although they did not give an estimate of the prefactor $p^2$ here. The scaling 3.8 implies that the machine can attain an asymptotic scaling arbitrarily close to $O(t^{-1})$. For a finite number of examples, however, it is not necessarily advantageous to use a kernel with larger $p$, as the prefactor $p^2$ increases rapidly with $p$. The optimal $p$ depends on the number of examples $t$ in such a way that

$$p \sim (\ln t)/4 \quad (3.9)$$

By substituting this into equation 3.8 we obtain the fastest convergence,

$$u(t) \sim O((\ln t)^2 t^{-1}) \quad (3.10)$$

This scaling form would presumably be optimal, although we have not succeeded in proving this.

### 4 Higher Dimensional Case

In this section, we wish to discuss whether the asymptotic scaling forms we obtained for the one-dimensional model are critically dependent on the dimension of the parameter space as well as the dimension of the input vector space. In higher dimensional problems, there are two main causes for the impossibility of reproducing the input–output relation. First, a target input–output relation is originally stochastic as is seen in our one-dimensional paradigm. In this case, it is impossible to reproduce individual examples even if the machine can produce an arbitrary decision boundary surface. Second, a target relation is deterministic and has a clear separation boundary in the input space, but the machine cannot reproduce the separation boundary due to limitations in its adaptability. For example, consider the combination of the target dichotomy with a round boundary and the machine that can produce a decision boundary by using only a finite number of hyperplanes. In practical applications, these two causes are presumably mixed; the target relation is more or less stochastic, and, moreover, the learning machine cannot produce an optimal decision boundary for this stochastic relation.

To shed light on the effect of the dimensionality, we will consider here the first case. Namely, the target relation is stochastic and the machine can produce the best decision boundary for this stochastic relation. Our model is as follows. For the $(D+1)$-dimensional unit vector $x$, output $s = \pm 1$ is drawn from the stochastic relation,

$$p(s = +1 | x) = 1/2 + k_1(\theta_0 \cdot x) + k_2(\theta_0 \cdot x)^2 + \cdots \quad (4.1)$$
where \( \theta_o \) is a \((D+1)\)-dimensional unit vector and \((a \cdot b)\) is the inner product. The optimal decision boundary is a hyperplane (containing the origin) normal to \( \theta_o \). The machine is able to choose any hyperplane containing the origin, or equivalently, a (unit) normal vector \( \theta \). The machine is actually capable of producing an optimal decision boundary by choosing \( \theta = \theta_o \). The parameter space is a \( D \)-dimensional manifold.

In learning with queries, the machine chooses inputs \( x \) randomly from the hypothetical boundary. The rule for updates can be of the form,

\[
\theta_{t+1} = (\theta_t + s \alpha_t x) / \sqrt{1 + \alpha_t^2}
\]  

(4.2)

where the factor \( 1/\sqrt{1 + \alpha_t^2} \) is added to keep the norm unity.

The dynamics is again similar to Brownian motion in a \((D\text{-dimensional})\) quadratic potential, if the input vector \( x \) is drawn uniformly from the hypothetical boundary. On a \( D \)-dimensional locally flat coordinate \( z \propto \theta_i - \theta_o \) normal to \( \theta_o \), we can estimate the average restoring force. The dynamics is then expressed by the Langevin equation,

\[
dz/dt = \alpha [-2k_1 z/D + \eta(t)],
\]

(4.3)

where \( \eta(t) \) is \( D \)-dimensional white noise whose statistical characteristics are given by \( \langle \eta(t) \rangle = 0 \) and \( \langle \eta(t) \eta(t') \rangle = (1/D) \delta_{ij} \delta(t - t') \). From this, we can obtain the evolution equation of the mean square deviation \( u = \langle |z|^2 \rangle \),

\[
du/dt = -(4k_1/D) \alpha u + \alpha^2
\]

(4.4)

In comparison with the one-dimensional case, the average restoring force is effectively weakened by a factor \( 1/D \) due to the higher dimensionality of the problem. The optimal convergence is then \( u(t) \sim D^2/(4k_1 t) \) and there is no qualitative difference.

As we pointed out, however, the preceding learning condition is unnatural because the learning machine can produce the optimal hypothetical boundary so that the boundary exactly lies on the surface \( p(s = +1 \mid x) = p(s = -1 \mid x) \). By this optimal choice the machine actually attains the best prediction for individual examples, irrespective of the probability of drawing inputs \( p(x) \). In practical circumstances, the learning machine cannot even produce the optimal surface at \( p(s = +1 \mid x) = p(s = -1 \mid x) \). In such a case, the choice of the optimal boundary from among the ones the machine can produce must depend on the probability of drawing inputs \( p(x) \). In the queries, the machine will draw inputs from the hypothetical boundary with a certain probability \( q(x) \), which is generally different from the natural probability \( p(x) \). In such a case, the hypothetical boundary does not approach the optimal choice.

In learning without queries, inputs \( x \) are drawn from a distribution \( p(x) \). A strategy similar to the one-dimensional case is also effective for this higher dimensional case. If we define a band of width \( 2\pi \) centered at the hypothetical boundary, with a similar update rule and appropriate
learning schedule \( \{ \alpha_t \} \), and also shorten the band width with a schedule \( \tau_t \sim O(t^{-1/5}) \), the machine attains the convergence \( u(t) \sim O(t^{-4/5}) \). The convergence of \( O[(\ln t)^2t^{-1}] \) may presumably be attainable by using a more sophisticated algorithm. We have not succeeded in making up kernel functions to further accelerate the convergence from \( O(t^{-4/5}) \). In the one-dimensional model, we could see that the strategy for obtaining \( O[(\ln t)^2t^{-1}] \) convergence is rather complicated to carry out practically. For practical application, the present algorithm providing \( O(t^{-4/5}) \) convergence is not only algorithmically simple but also sufficiently efficient.

Acknowledgments

We would like to acknowledge useful discussions with Shun-ichi Amari, Hideki Asoh, Motoaki Kawanabe, Makoto Fushiki, and Michael Crair. One of the authors (YK) would like to thank Naoki Abe, Jun-ichi Takeuchi, and Kazuhiro Iida at NEC C&C Information Technology Research Laboratories for helpful comments and encouragement. This research is partly supported by a Grant-in-Aid for Scientific Research on Priority Areas by the Ministry of Education, Science and Culture, Japan.

References


Received January 4, 1993; accepted May 16, 1994.