On the Effectiveness of Memory-Based Methods in Machine Learning[5]

George Cybenko[1]
Sirpa Saarinen[2]
Robert Gray[1]
Yunxin Wu[3]
Alexy Khrabrov[4]

Abstract

Many memory-based methods for learning use some form of nearest neighbor inference. By memory-based, we mean methods that localize data in the training sample to make inferences about novel feature values. The conventional wisdom about nearest neighbor methods is that they are subject to various *curses of dimensionality* and so become infeasible in high dimensional feature spaces. However, recent results such as those by Barron and Jones suggest that these dimensionality problems can be overcome in the case of parametric models such as sigmoidal neural networks which are patently nonlocal. This creates a paradox because memory-based methods have been shown to perform well in a number of applications. They are often competative with parametric methods in terms of prediction error and actually superior in terms of training time. In this paper, we study the unreasonable effectiveness of memory-based methods. We analyze their performance in terms of new metrics that take into consideration the interaction between the function being estimated and the underlying probability distribution generating the samples. Extensions of this analysis method might serve as the basis for a new foundation for more general memory-based methods that could explain their observed performance on real problems.

[1] Present Address: Thayer School of Engineering, Dartmouth College, Hanover, NH 03755 USA.

[2] Present Address: Wolfram Research Inc., Champaign, IL 61820 USA

[3] Present Address: Microsoft Corporation, Richmond, WA

[4] Present Address: Department of Computer Science, University of Pennnsylvania, Philadelphia, PA

[5] Partially supported by AFOSR grant F49620-95-1-0305 and Air Force Multidisciplinary University Research Initiative grant F49620-97-1-0382.

1. Introduction

A significant body of current literature and research is devoted to learning techniques that use direct, explicit representation of training data for learning, recognition and classification. Among the different terms used for variations of memory-based learning are: memory-based reasoning, case-based reasoning, lazy learning, radial basis functions, nearest neighbors methods, exemplar-based, instance-based, and analogical. Moreover, a number of other methods commonly used in machine learning such as adaptive resonance theory (ART), self-organizing feature maps and vector quantization are also explicitly memory based. These ideas are conceptually simple to understand and implement because they depend on simple locality arguments, clustering and interpolation algorithms. The theoretical underpinnings of this class of some memory-based approaches are solid in the asymptotic limit – namely, they will perform at or close to the Bayes limit [1] for a large class of problems.

However, there continue to be serious difficulties with memory-based methods in the non-asymptotic case. First of all, it is easy to see that for problems with high dimensional features or keys, even extremely large training sets will be sparse in the full space. Secondly, algorithms and data structures for efficiently dealing with high dimensional keys are primitive and reduce to linear searching algorithms quite often. That is, even without the theoretical density issues, the implementation problems of faster searching and updating remain as obstacles.

The first difficulty described above is paradoxical because, in practice, memory-based methods perform quite acurately when implemented properly and on appropriate applications [2, 3, 4]. This suggests that the conventional theory is somehow not incorporating relevant properties of many real learning and classification problems. It is undeniable that, in spite of theoretical density issues, experimental results demonstrate the equal or superior power of memory-based methods on many problems. As described below, we believe that existing theory fails to adequately model the interactions between the process generating the data and the performance criterion.

In particular, arguments against memory-based methods typically involve uniform distributions of data, the inclusion of many irrelevant features and/or uniform error estimates. These factors rarely appear to play a significant role in real problems. Data is often clustered and the performance criterion typically involves weighting by the data distribution. As for the inclusion of irrelevant features, proper modeling and selection of the feature space in an application should preclude this difficulty. Moreover, many real applications involve estimating functions that change slowly in regions of high probability and make abrupt transitions only in regions of low probability.

This kind of relationship between the object being estimated and the underlying probability distribution is not directly modeled by current theory. For example, in the PAC framework, the function classes and probability distributions are constrained independently. Our analysis of memory-based methods involves conditions on both simultaneously so that these classes are not independently constrained. Two extreme cases demonstrate this interdepence. On the one hand, it is trivial to "learn" a constant function under any distributional law, even the uniform one. One the other hand, it is also "easy" to learn *any* function if the distribution is concentrated at a few points even in very high dimensional spaces.

Another major stumbling block for memory-based methods has been efficiency – namely the performance of table lookup and associative addressing procedures. While data structures such as *k-d* trees [5] allow efficient retrieval of neighborhoods for fixed dimensions, the performance as a function of key dimension increases exponentially in the size of the key. This results in linear searches for situations with high dimensional keys or features. These linear searches are inefficient when large training sets are available.

Virtually all existing techniques are explicitly deterministic and seek exact neighborhoods. We will show that some memory-based learning techniques can be formulated as quadrature problems for which Monte Carlo methods work at a fraction of the cost of deterministic methods. That is, instead of finding exact neighborhoods as is currently being done, it should be possible to use approximate neighborhoods and stochastic algorithms to get significant speedups in searching without sacrificing too much performance in accuracy. Initial work in this direction has already been done [6].

Much of the recent theory about machine learning has focused on parametric methods: that is, methods that use some explicit family of functions parameterized in some natural way. Sigmoidal neural networks fall into this category and are perhaps the best examples. Memory-based approaches to learning are different from neural network methods in that there is no single global parametric model of the system being learned or modeled. Feedforward neural networks with sigmoidal activation functions are patently nonlocal – the functions and therefore the network response depend on behavior over a large portion of the feature space. Attempts to localize the response can lead to memory-based methods of one form or another.

Some of the attempts at localizing response has led to hierarchical networks advocated, for example, by Jordan and others [7]. These approaches partition the space adaptively and allow different subnets to optimize themselves to those subregions. Other approaches implicitly use lookup tables with some form of local smoothing, for example the radial basis function methods first developed by Powell in approximation theory [8] and developed by Poggio, Girosi and Moody for learning and recognition [9, 10, 11]. Statistical methods such as CART and MARS also partition the input space and attempt to construct estimates based on local information primarily.

A number of authors have made similar observations. Lee has performed a number of experiments comparing various learning methods and has commented on the strikingly good performance of memory-based methods [2]. Lin and Vitter have developed models of memory-based learning problems in the PAC [12] framework [13]. However, to our knowledge the present work is the first effort to combine the target function's behavior with the underlying probability distribution to arrive at models and analyses that capture the tight couplings that often appear to exist in real problems.

Section 2 develops some background. Section 3 develops an example of the curse of dimensionality that is used in later sections. Section 4 reviews the aforementioned work by Baron and Jones while Section 5 briefly presents the results of some simultations by other authors. Section 6 contains the main technical result of this paper which is a PAC type learning result for functions whose variations with respect to a probability distribution are bounded. Section 7 is a discussion of the results with some dieas for future work.

2. Background

We now introduce some basic notation and definitions. Sample input values $x_i \in \mathbf{R}^d$ are generated by an unknown probability law, μ, and for each x_i we have a deterministic (for simplicity) class or function value, $f(x_i) = y_i$. The aggregate sample data is $S = \{(x_i, y_i) | i = 1, ..., N\}$. The x_i are independent and identically distributed according to the probability distribution, μ. We normally think of the x's as features or system inputs. The goal is to estimate f over the whole region of support of μ. Given the training sample S, the hypothesized function, f_S, is constructed by a *learning algorithm*. The estimation error criterion typically involves the underlying probability distribution function according to

$$E_\mu(||f_S - f||)$$

where $||\cdot||$ denotes some distance, such as squared error and expectation is with respect to the underlying distribution, μ. Since this depends on the training sample, S, which is itself a random variable, the error is also a random variable. Quantification of the variation in error over the training set is typically accomplished by introducing the following probability:

$$\text{Prob}\{S | E_\mu(||f_S - f||) > \epsilon\} < \delta.$$

This describes the performance of a particular learning method in terms of two parameters: δ and ϵ.

In the widely accepted PAC model of learning [12], the functions f are constrained to belong to some class, $\mathcal{F}$ and the distributions μ may or may not be constrained as well. Loosely speaking, a class of functions $\mathcal{F}$ are PAC learnable if for every $\epsilon > 0$ and $\delta > 0$, there is some N and an efficient algorithm for constructing f_S so that

$$\text{Prob}\{S | E_\mu(||f_S - f||) > \epsilon\} < \delta.$$

whenever $|S| > N$ where $S = \{(x_i, f(x_i)\}$ and $f \in \mathcal{F}$.

Learning and estimation techniques are generally successful in an application because for most regions of interest, that is, regions with highest probability, the value of the target function, $f(x)$, changes slowly with x. This allows generalization in the sense that output values for inputs are close to outputs for sample inputs in the training data. There may be discontinuities in f but they occur in regions of lower probability and so contribute less to the overall error. Such continuity properties play a role, implicitly or explicitly, in virtually all learning and estimation problems involving real valued or finely quantized features.

Given a new value, $\hat{x}$, memory-based methods estimate $f(\hat{x})$ by retrieving data, $(x_i, f(x_i))$, with x_i near $\hat{x}$. Some sort of interpolation is used on the resulting x_i thus selected. In the case of classical nearest neighbor methods, one can estimate $f(\hat{x})$ by an average

$$f(\hat{x}) \approx f_S(\hat{x}) = \frac{1}{n} \sum f(x_{i'})$$

where the $x_{i'}$ are k close feature values. Closeness is with respect to a general, possibly spatially varying, metric. We stress that many inference techniques use similar ideas although it may not always be explicit. (We previously mentioned for example: memory-based reasoning, case-based reasoning, lazy learning, radial basis functions, exemplar-based, instance-based, and analogical.) In radial basis function methods with rapidly decaying kernels, the weightings are not uniform but the basic local averaging property still holds.

3. The Curse of Dimensionality

The curse of dimensionality arises in machine learning settings when one of the above errors, δ or ϵ, are related to the training set size exponentially such as

$$N \approx \frac{C}{\epsilon^d}$$

where C is a constant and d is the dimension of the feature space. To construct a simple example using smooth functions, consider the class of real-valued functions

$$\mathcal{F} = \{f \in C^1([-1,1]^d)| \ \ |\nabla f(x)| \leq 1\}.$$

For $v \in \mathbf{R}^d$, let

$$g_v(x) = (1 - |x - v|^2)^2$$

for $|v - x| \leq 1$ and $g_v(x) = 0$ otherwise. Let $\{v_j\}$ be an enumeration of the 2^d vertices of $[-1,1]^d$ and note that $0 \leq g_{v_j}(x) \leq 1$ for $x \in [-1,1]^d$ with $g_{v_j}(v_j) = 1$. Moreover,

$$|\nabla g_{v_j}(x)| \leq \frac{4}{3\sqrt{3}} < 1$$

for $x \in [-1,1]^d$.

Now let

$$f(x) = \sum_j \alpha_j g_{v_j}(x)$$

where $\alpha_j = \pm 1$ equally probably. Then $f \in \mathcal{F}$ and $f(v_j) = \pm 1$. This f has values ± 1 at each of the vertices of $[-1,1]^d$ and yet has gradient bounded by 1. Clearly, any estimate of f based on samples has probability of 0.5 of estimating the value of f incorrectly in a quadrant where no data samples have been drawn. If we assume the uniform distribution on $x \in [-1,1]^d$ for sampling, then for any sample of size N, that is $|S| \geq N$, we have

$$\text{Prob}\{x \text{ such that } |f_S(x) - f(x)| > 1\} > \frac{1}{2} - \frac{N}{2^{d+1}}$$

This derivation was not made in the PAC framework but it can easily be extended. The reader is invited to check the following details.

Assuming a uniform distribution, μ, on $[-1,1]^d$, we have both

$$||f||_2 = \int_{[-1,1]^d} |f(x)|^2 dx \approx \frac{8}{n^3}$$

and

$$||\nabla f||_2 = \int_{[-1,1]^d} |\nabla f(x)|^2 dx \approx \frac{8}{n^3}.$$

It can be shown that the expected error

$$\int_{[-1,1]^d} |f(x) - f_S(x)|^2 dx$$

is at least

$$\frac{4}{n^3}(1 - \frac{N}{2^d})$$

for any sample of size N. If we normalize f so that $||f||_2 = 1$ then $||\nabla f||_2 \approx 1$ also and this error would be about $1 - \frac{N}{2^d}$ so that to achieve an error of no more than ϵ we would need at least $(1-\epsilon)2^d$ samples which grows exponentially in d for fixed ϵ.

This example is of interest because it involves a function class with bounded norm and bounded averaged gradient as well and we will return to it in the next section.

4. The Barron-Jones Theory

Barron and Jones have introduced a powerful new analysis technique into machine learning that oversomes the curse of dimensionality in a large class of problems of machine learning problems involving feedforward type neural networks. We refer the reader to the original articles [14, 15] for details and only give a sketch of the main ideas here.

The basic results derived by Jones, Barron and Girosi show dimension independent convergence rates for feedforward neural networks and radial basis function methods when those methods are applied to specific constrained classes of functions to be learned.

The following result is taken from [14]. Let ϕ be a sigmoidal function on $\mathbf{R}^1$ (see [16]) such as is commonly used in feedforward neural networks. A superposition of such sigmoidal functions has the form

$$f_n(x) = \sum_{j=1}^{n} c_j \phi(a_k \cdot x + b_k) + c_0.$$

which is the output of a feedforward neural network with a single hidden layer and one output node.

Theorem [14] – Let

$$\int_{\mathbf{R}^d} |\omega||F(\omega)| d\omega \leq C < \infty$$

where f and F are a Fourier transform pair of functions on $\mathbf{R}^d$. Let B_r be the ball of radius r centered at 0 and μ be a probability measure on $\mathbf{R}^d$. Then for every $n \geq 1$, there is a superposition of sigmoidals involving n terms so that

$$\int_{B_r} (f(x) - f_n(x)^2 d\mu(x) \leq \frac{(2rC)^2}{n}.$$

This says that sigmoidal networks with n nodes can approximate smooth functions with an error rate of O($\frac{1}{n}$). This is a major breakthrough considering that earlier approximation results gave either exponential convergence rates or no rates at all, merely existence proofs [16, 17]. A number of extensions of this result can be found in the original article [14]. This result has been used widely to justify the use of feedforward neural networks in machine learning problems. Earlier work by Jones derived a similar result for projection pursuit methods [15]. All of those results rest on a powerful general theory stated below.

Theorem (Pisier [18]) – Suppose that G is a set in a Hilbert space H with $||g||^2 < C$ for all $g \in G$. Let f be in the closure of the convex hull of G. Then for every n, there are $g_i \in G, i = 1...n$ and coefficients $\lambda_i, i = 1...n$ so that

$$||f - \sum_{i=1}^{n} \lambda_i g_i||^2 \leq C/n.$$

.

While Pissier's theorem is powerful and general, its actual applicability in a specific case must be carefully examined. To illustrate the possible difficulties, consider the following. Take as G a set of m orthonormal vectors, g_i, in H (an infinite dimensional space). Let $f = 1/m \sum g_i$. The norm of f satisfies

$$||f||^2 = 1/m^2 \sum ||g_i||^2 = 1/m$$

and $||g_i|| = 1 = C$ is the bounding constant. Note that the conclusions of the theorem are satisfied by the zero vector since

$$||f||^2 = 1/m \leq 1/n$$

for any $1 \leq n \leq m$. The result is vacuously true in this situation because the norm of f is so small.

It is important to understand the relevance of this observation. Barron [14] has shown a linear convergence rate for feedforward neural networks. The same technique has recently been used by Girosi [11] to establish a linear convergence rate for radial basis function methods. While Barron's and Girosi's results are technically correct, they must be interpreted and used carefully. In particular, we have show that in a simple case, the bounds obtained by the Pissier theorem are vacuous and shed no real light on convergence rates. The problem has to do with convexity and its relationship to orthogonality in Hilbert space norms.

Another example builds on the functions f introduced in the previous section. Recall that when normalized, f has norm approximately one which is also about the size of the norm of ∇f. It is important to note that these are the norms restricted to the hypercube $[-1, 1]^d$ and not on all of $\mathbf{R}^d$. Noting that f is a convex combination of the generators g_{v_j} which are orthogonal, the same vacuous statement about convergence rates is made by the Pissier theorem. At the same time, if we note that ∇f is bounded by 1 also (when normalized), the Barron theory suggests that we can get linear convergence rates using sigmoidal network approximations. However, this contradicts the exponential rate we demonstrated in the previous section.

This seeming contradiction is resolved by recalling that the Barron result requires a bound on the gradient over all of $\mathbf{R}^d$ and not just on a subset. A smooth extension of this f will lead to a significantly larger bound on ∇f which will be exponential in d. Moreover, the sensitivity of the bound to scaling of the coordinate space are already noted by Barron [14].

5. Experimental Results

A number of empirical comparisons of methods for solving classification problems have been conducted. In this section, we briefly summarize some of those findings, refering the reader to original sources for complete details [19, 4, 2].

Lee and Lippmann report on a handwritten character recognition problem using backpropagation networks, k-nearest neighbors and radial basis functions [19, 2]. They quantized handwritten characters into 360 pixels, each with 10 gray-scale levels. The training set consisted of 30,600 samples and the test set had 5,060 patterns. They used $k = 9$ neighbors which was determined empirically. The radial basis function method used 1,000 basis elements while the feedforward network had 540 and 102 nodes in the two hidden layers. Timings are reported for a DECstation 3100 rated at 3.7 Megaflops. Results of their experiments are shown in Table 1. Table 1 is at the end of the article.

Ripley [4] surveys a number of classification techniques and reports on experiments comparing them. The following error rates are reported with 0% rejection rate (as was done above). The computations were done on a SparcStation IPC (about 2 Megaflops rating). The problem involves learning the decision regions for Tsetse flies in Zimbabwe based on 12 environmental variables. The feedforward networks used had 6 and 12 nodes on one hidden layer using the *quickprop* algorithm for training. Learning vector quantization used 200 codebook vectors. The training set is based on 500 samples which is also the size of the test set. Timings include training and evaluation on the test set. Table 2 is at the end of the article.

These empirical results are but two examples of the effectiveness of nearest neighbor methods. There are numerous other simulations that support the conclusion that memory-based methods can perform competatively on real problems.

6. Analysis of Memory-Based Methods

In this section, we explore a new approach to analysing memory-based methods in terms of the interaction between the underlying probability distribution and the target function. Let $D \subset \mathbf{R}^d$ be the support of a probability distribution μ. If μ is continuous with respect to Lesbegue measure then $d\mu(x) = g(x)dx$ for $x \in D$ and $g(x) > 0,\ x \in D$.

A basic measure of the variation of a target function, f, with respect to μ is

$$\int |\nabla f(x)| g(x) dx = \int |\nabla f(x)| d\mu(x)$$

Later we also uses the slightly modified measure

$$V(f, g) = \int |\nabla f(x)| g(x)^{\frac{n-1}{n}} dx$$

when μ is continuous with respect to Lesbegue measure.

For $\rho > 0$, let $B(x,\rho)$ be the ball centered at x of sufficient radius, ϵ, so that

$$\int_{B(x,\rho)} d\mu(x) = \rho.$$

Note that when g exists as above and is continuous, asymptotically ϵ is related to ρ via the relationship

$$g(x)C_d\epsilon^d \approx \rho$$

where $C_d = \pi^{d/2}/\Gamma(\frac{n}{2}+1)$ is the volume of the ball of radius 1 in $\mathbf{R}^d$. Then $\epsilon \approx \rho^{\frac{1}{n}} g(x)^{\frac{-1}{n}} C_d^{\frac{-1}{n}} \approx \rho^{\frac{1}{n}} g(x)^{\frac{-1}{n}} \pi^{\frac{-1}{2}} (n/2e)^{\frac{1}{2}}$ by Stirling's formula.

Introduce the average variation in f over balls of probability ρ with respect to μ as

$$V(f,\mu,\rho) = \int_D \frac{1}{\rho} \int_{B(x,\rho)} |f(x) - f(y)| d\mu(y) d\mu(x).$$

Compare this with *uniformly Lipshitz on average* functions introduced by Haussler [20]. Also define

$$\begin{aligned} W(f,\mu,\rho) &= \int_D \frac{1}{\rho} \int_{B(x,\rho)} |f(y) \\ &- \frac{1}{\rho} \int_{B(x,\rho)} f(z) d\mu(z)|^2 d\mu(y) d\mu(x). \end{aligned}$$

as the variance of f over balls of probability ρ averaged over D. For smooth f we know that $V(f,\mu,\rho) \to 0$ and $W(f,\mu,\rho) \to 0$ as $\rho \to 0$ (by dominated convergence for example).

To get a feeling for these measures of variation, it is useful to apply them to the previously mentioned extreme cases than can arise. In the case of constant f, the measures are 0 for all ρ. In the case of an arbitrary f but with a distribution that is concentrated at a finite number of point masses, the measures are 0 when ρ is smaller than the smallest point mass weight.

Theorem – Let $\alpha, \delta, k > 0$. Pick ρ so that $V(f,\mu,\rho) < \alpha\delta/8$ and $W(f,\mu,\rho) < \sqrt{k}\alpha\delta^{3/2}/16$. Then for a sample of size N for which $N\rho - 2\sqrt{(N/\delta)}\sqrt{\rho(1-\rho)} > k$ we will have

$$|f(x) - \frac{1}{k}\sum_{j=1}^{k} f(x_j)| < \alpha$$

with probability at least $1-\delta$. Here x_j are the k nearest neighbors of x from the sample of size N.

Outline of Proof – The basic idea is to break the problem down into four events, each one of whose probability can be made arbitrarily close to 1. Three of the events have to do with the local variations in f and ultimately measure the rate at which a Monte Carlo quadrature method should work for estimating f locally. The fourth event arises from purely sampling considerations, namely, how many samples are needed to guarantee enough local values on which to base a Monte Carlo estimate with high enough probability. The basic tool used is a Tchebyshev type inequality which arises repeatedly.

Proof – By the above definitions, we have

$$\int_D |f(x) - \frac{1}{\rho}\int_{B(x,\rho)} f(y)d\mu(y)|d\mu(x) \le V(f,\mu,\rho).$$

Now,

$$\begin{aligned} \text{Prob}\{ \quad x & \quad \text{such that } |f(x) \\ & - \frac{1}{\rho}\int_{B(x,\rho)} f(y)d\mu(y)| \ge \alpha/2\} \\ & \le 2V(f,\mu,\rho)/\alpha \\ & \le 2\rho^{\frac{1}{n}}\pi^{\frac{-1}{2}}(n/2e)^{\frac{1}{2}}V(f,g)/\alpha \end{aligned}$$

so that

$$\begin{aligned} \text{Prob}\{ \quad x & \quad \text{such that } |f(x) \\ & - \frac{1}{\rho}\int_{B(x,\rho)} f(y)d\mu(y)| \le \alpha/2\} \\ & \ge 1 - 2\rho^{\frac{1}{n}}\pi^{\frac{-1}{2}}(n/2e)^{\frac{1}{2}}V(f,g)/\alpha \\ & \ge 1 - 2V(f,\mu,\rho)/\alpha \ge 1 - \delta/4 \end{aligned}$$

by the choice of ρ as stated in the theorem. Similarly,

$$\begin{aligned} \text{Prob}\{ \quad x & \quad \text{such that } \frac{1}{\rho}\int_{B(x,\rho)} |f(y) \\ & - \frac{1}{\rho}\int_{B(x,\rho)} f(z)d\mu(z)|^2 d\mu(y) \le \sqrt{\delta k}\alpha/4\} \\ & \ge 1 - 4W(f,\mu,\rho)/(\alpha\sqrt{\delta k}) \\ & \ge 1 - \delta/4 \end{aligned}$$

by the choice of ρ again.

Thus the set of x for which both

$$\frac{1}{\rho}\int_{B(x,\rho)} |f(y) - \frac{1}{\rho}\int_{B(x,\rho)} f(z)d\mu(z)|^2 d\mu(y) \le \alpha\sqrt{\delta k}/4$$

and

$$|f(x) - \frac{1}{\rho}\int_{B(x,\rho)} f(y)d\mu(y)| \leq \alpha/2$$

has probability at least $1-\delta/2$.

For a sample of size N where

$$N\rho - 2\sqrt{(N/\delta)}\sqrt{\rho(1-\rho)} > k,$$

the number of samples in the ball $B(x,\rho)$ is at least k with probability at least $1-\delta/4$. To see this, we use Tchebyshev's inequality. Let $X_i = 1$ if the ith sample among the N drawn is in $B(x,\rho)$ and $X_i = 0$ otherwise. Then the sequence X_i is Bernoulli with probabilities ρ and $1-\rho$ of being 1 and 0 respectively. We have

$$\text{Prob}\{|\frac{1}{N}\sum X_i - \rho| < S\sigma'\} \geq 1 - \frac{1}{S^2} \geq 1-\delta/4$$

for $S = 2/\sqrt{\delta}$ where $\sigma' = \sqrt{\rho(1-\rho)/N}$ is the variance of $\frac{1}{N}\sum X_i$. Thus, with probability at least $1-\delta/4$, we have

$$\sum_i X_i \geq N(\rho - S\sigma') = N(\rho - 2\sqrt{\rho(1-\rho)}/\sqrt{N\delta} > k.$$

These k samples, say $x_j, j = 1, ..., k$ can be used for a Monte Carlo estimate of $\int_{B(x,\rho)} f(y)d\mu(y)$ according to

$$\int_{B(x,\rho)} f(y)d\mu(y) \approx \frac{1}{k}\sum_j f(x_j)$$

which has variance

$$\begin{aligned}\sigma_x &= \frac{1}{\rho\sqrt{k}}\int_{B(x,\rho)} |f(y) - \frac{1}{\rho}\int_{B(x,\rho)} f(z)d\mu(z)|^2 d\mu(y)\\ &\leq \sqrt{\delta}\alpha/4\end{aligned}$$

when x is in the previously specified set.

By Tchebyshev's inequality again,

$$\begin{aligned}\text{Prob}\{|\frac{1}{k}\sum_j f(x_j) &- \frac{1}{\rho}\int_{B(x\rho)} f(y)d\mu(y)| < R\sigma\}\\ &\geq 1 - 1/R^2\end{aligned}$$

With $R = 2/\sqrt{\delta}$ and $\sigma \leq \sqrt{\delta}\alpha/4$, we have

$$\begin{aligned}\text{Prob}\{|\frac{1}{k'}\sum_j f(x_j) &- \frac{1}{\rho}\int_{B(x\rho)} f(y)d\mu(y)| < \alpha/2\}\\ &\geq 1-\delta/4.\end{aligned}$$

Combining all of the above, we have with probability at least $1-\delta$, that both

$$|\frac{1}{k}\sum_j f(x_j) - \frac{1}{\rho}\int_{B(x\rho)} f(y)d\mu(y)| < \alpha/2$$

and

$$|f(x) - \frac{1}{\rho}\int_{B(x,\rho)} f(y)d\mu(y)| \leq \alpha/2$$

from which

$$|\frac{1}{k}\sum_j f(x_j) - f(x)| < \alpha$$

follows by the triangle inequality. □

7. Discussion

The main result of the previous section does not, nor cannot, defeat the curse of dimensionality in all cases. To get a sense of this note that

$$\begin{aligned}
V(f,\mu,\rho) &\leq \int_D \frac{1}{\rho}\int_{B(x,\rho)} |\nabla f(x)|\cdot|x-y|d\mu(y)d\mu(x) \\
&\leq \int_D |\nabla f(x)|\frac{1}{\rho}\int_{B(x,\rho)} \epsilon d\mu(y)d\mu(x) \\
&\leq \int_D |\nabla f(x)|\rho^{\frac{1}{n}} g(x)^{\frac{-1}{n}} \pi^{\frac{-1}{2}}(n/2e)^{\frac{1}{2}} d\mu(x) \\
&\leq \rho^{\frac{1}{n}}\pi^{\frac{-1}{2}}(n/2e)^{\frac{1}{2}}\int_D |\nabla f(x)| g(x)^{\frac{n-1}{n}} d(x) \\
&= \rho^{\frac{1}{n}}\pi^{\frac{-1}{2}}(n/2e)^{\frac{1}{2}} V(f,g)
\end{aligned}$$

to the first order in $\rho^{1/n}$. The same bound can be derived for $W(f,\mu,\rho)$ (this is left to the reader). This suggests that the convergence of $V(f,\mu,\rho)$ to zero is going to be slow in most cases. It is governed by both $\rho^{1/d}$ and $V(f,g)$ when g exists. Now $\rho^{1/d}$ approaches 0 very slowly for large d but $V(f,g)$ can be small for a problem and herein lies at least one explanation for the good observed performance of many memory-based learning methods.

As previously noted, this analysis can deal with both extreme cases: that of a trivial function and uniform probability distribution; and that of a complex function with a simple point mass distribution. We know of no other analysis demonstrating that both cases are "learnable."

It would be interesting to see whether the proof technique we use can be extended to other memory-based methods. We suspect that it can and this should form the basis for further work. The question of efficiently estimating $V(f,\mu,\rho)$, $W(f,\mu,\rho)$ and $V(f,g)$ in a specific case is interesting of course and should be attempted for some learning problems where memory-based methods are both successful and a failure.

The Monte Carlo interpretation of memory-based methods suggests that approximate nearest neighbor searches should be acceptable for some problems but with improved efficiency. This has been observed by Saarinen [6].

References

[1] T. Cover, "Estimation by the nearest neighbor rule," *IEEE Transactions on Information Theory*, vol. 14, pp. 50–55, 1968.

[2] Y. Lee, "Handwritten digit recognition using k-nearest neighbor, radial-basis function, and backpropagation neural networks," *Neural Computation*, vol. 3, pp. 440–449, 1991.

[3] W. Huang and R. Lippmann, "Comparisons between neural net and conventional classifiers," tech. rep., MIT Lincoln Laboratory, 1987.

[4] B. Ripley, "Statistical aspects of neural networks," tech. rep., Oxford University, Department of Statistics, 1992.

[5] F. Preperata and A. M. Shamos, *Computational Geometry.* New York: Springer-Verlag, 1985.

[6] S. Saarinen, "Ph.D. Thesis, Department of Computer Science, University of Illinois at Urbana," 1994.

[7] M. Jordan, *Hierarchies of adaptive experts.* San Mateo, CA: Morgan Kaufmann, 1992.

[8] M. Powell, "Radial basis functions for multivariable interpolation: a review," in *IMA Conference on Algorithms for the Approximation of Functions and Data*, Oxford University Press, 1987.

[9] J. Moody and C. Darken, "Learning with localized receptive fields," Tech. Rep. DCS/RR-649, Yale University, Department of Computer Science, September 1988.

[10] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proceedings of IEEE*, vol. 78, pp. 1481–1497, 1990.

[11] F.Girosi and G. Anzellotti, "Rates of convergenceof approximation by translates," Tech. Rep. 1288, MIT AI Laboratory, 1992.

[12] L. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27:11, pp. 1134–1142, 1984.

[13] J.-H. Lin and J. Vitter, "A theory for memory-based learning," in *Proceedings of COLT '92*, pp. 103–115, ACM, 1992.

[14] A. Barron, "Universal approximation bounds for superpositions of a sigmoidal function," *IEEE Transactions on Information Theory*, vol. 39, pp. 930–946, 1993.

[15] L. K. Jones, "Constructive approximations for neural networks by sigmoidal functions." preprint, 1988.

[16] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, pp. 303–314, 1989.

[17] K. Hornik, M. Stinchcombe, and H. White, "Multi-layer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[18] G. Pisier, "Remarques sur un resultat non publie de B. Maurey," *Seminaire d'analyse functionelle*, vol. 1-12, 1980-1981.

[19] S. Lippmann, "An introduction to computing with neural nets," *Computer Architecture News*, vol. 16, pp. 7–25, 1988.

[20] D. Haussler, "Generalizing the PAC model for neural net and other applications," Tech. Rep. UCSC-CRL-89-30, Computer Research Laboratory, UC-Santa Cruz, 1989.

	Backprop net	*k-neighbors*	*radial basis*
Error rate	5.15%	5.14%	4.77%
Parameters	5,472	11,016,000	371,000
Training time (hours)	67.68	0.00	16.54
Classification time (sec/char)	0.14	6.22	0.24

Table 1: Handwritten Character Recognition (from [2]).

Method	*Error (%)*	*Time*
1-NN	4.4	4 secs
3-NN	5.4	4 secs
Neural net (6 nodes)	4.2	3 hours
Neural net (12 nodes)	5.0	3 hours
LVQ	5.4	44 secs
Projection Pursuit	5.2	50 secs

Table 2: Tsetse Fly Distribution (from [4]).