Quasi-regression

Jian An

Stanford University
E-mail: ja@stat.stanford.edu

and

Art Owen

Stanford University
E-mail: owen@stat.stanford.edu

Quasi-regression is introduced for approximation of functions on the unit cube in s dimensions. It is computationally efficient, compared to kriging, for problems requiring a large number of function evaluations. This paper describes how to implement quasi-regression and shows how to estimate the approximation error using the same data used to build the approximation. Four example functions are investigated numerically.

Key Words: computer experiments, function mining, kriging, numerical noise, quasi-interpolation

1. INTRODUCTION

We consider the problem of approximating a function $f : [0,1]^s \to \mathrm{R}^q$ by another function $\hat{f} : [0,1]^s \to \mathrm{R}^q$. A good approximation $\hat{f}$ should be close to f in some norm (such as L^2), and it must possess at least one advantage over f: it may be faster to compute, it may be smoother and hence more amenable to optimization, or it may have a form, such as an anova decomposition, which yields insight into f. We suppose that the function f can be evaluated at any point $x \in [0,1]^s$, and that $\hat{f}$ is to be based on n function values $f(x_1), \ldots, f(x_n)$.

We are motivated here by problems arising in computer experiments [4, 8, 16]. In such applications, a function f describes the performance of a product such as an aircraft or semiconductor as a function of s variables $x = (x^1, x^2, \ldots, x^s)$ chosen to describe how it is manufactured. In semiconductor applications f may describe how fast and how stably a transistor

will switch, while in aerospace, f may describe lift and drag of a plane. In both industries extensive simulation and experimentation are carried out on computer models, before moving on to physical experimentation. While it is common to have two or more responses, we will approximate them separately, and so we take $q = 1$.

The time to compute f may range from fractions of a second to several hours. The dimension s can vary significantly. The authors know of examples with $s = 3$ and others with $s \geq 80$. The chore of extracting information from a computer model may be likened to that of extracting information from a large data base, though such function mining differs from data mining in that one has more control over the variables.

To fix ideas, we consider the borehole function of Morris, Mitchell and Ylvisaker [9] defined by

$$\frac{2\pi T_u\,[H_u - H_l]}{\log\left(\frac{r}{r_w}\right)\left[1 + \frac{2LT_u}{\log\left(\frac{r}{r_w}\right)r_w^2 K_w} + \frac{T_u}{T_l}\right]}. \tag{1}$$

This function is a model for the flow rate of water from an upper to a lower aquifer. The aquifers are separated by an impermeable rock layer but there is a borehole through that layer connecting them. The inputs r and r_w are radii of the borehole and the surrounding basin respectively, T_u and T_l are transmissivities of the aquifers, H_u and H_l are their potentiometric heads, L is the length of the borehole and K_w is a conductivity. Thus there are 8 input variables, that after appropriate scaling, yield $x \in [0,1]^8$.

As Diaconis [5] points out, knowing a formula for a function does not mean that we fully understand it. For example, looking at equation (1) does not easily let us know which are the most important input variables, or whether the function is nearly additive, or even linear, in the input variables. In fact, the answer must clearly depend on the ranges over which the raw input variables vary. But given those ranges it may still require numerical investigation to answer questions about the input variables.

The functions that motivate us may be similarly smooth to the borehole function, because they model physical phenomena. They are not ordinarily as fast to evaluate as the borehole function, as their computation may have numerical optimizations or solutions of partial differential equations embedded in them. Sometimes the functions are only piecewise continuous, even though they model a continuous physical process. The reason is that a small change in x could result in an optimization taking a different number of steps, or in a different finite element grid being generated, or a different number of terms in a series approximation being used. Such effects, called "numerical noise", are common in computer experiments, and can raise difficulties for methods that assume very smooth functions.

For the borehole function, we might seek an approximation that gives insight into the relative effects of the input variables. For functions computed by PDE's a fast function $\hat{f}$ might be desired so as to allow a numerical exploration of the tradeoff between two quantities such as lift and drag. For functions with numerical noise, a smooth approximation may be desired for optimization. Once a potential optimum x^* is located for $\hat{f}$, the original function f can be investigated in the neighborhood of x^*.

Statistical methods have something to offer in approximation problems, especially for larger s. The reason is that any feasible sample $x_1, \ldots, x_n$ is necessarily very sparse when s is large. The error in approximation depends on the value of f at points not sampled, and the language of probability is very well suited to describing how the function might behave where it was not sampled.

Section 2 provides the notation underlying statistically motivated approaches to approximation. The present state of the art consists primarily of kriging methods. They originated in geostatistics; see for example, Journel and Huijbregts [7]. The value and elegance of kriging for computer experiments was shown by Currin, Mitchell, Morris and Ylvisaker [4] and by Sacks, Welch, Mitchell and Wynn [16]. Kriging allows one to incorporate derivative information on the function, and the mathematical framework supports a notion of optimal designs. Section 2 also presents regression and quasi-regression methods.

Kriging becomes awkward numerically when n increases, eventually becoming infeasible, as shown in Section 3. For large s, it is reasonable to expect that large n will be required. Section 3 also presents regression and quasi-regression methods for approximation. Quasi-regression requires less time and space than regression. Section 4 describes some issues in implementing quasi-regression. Section 5 describes how we select out the low order elements in a tensor product of univariate bases. Section 6 presents 4 example functions, purposely split into two where quasi-regression is successful and two where it fails. The method can still provide useful information regarding functions for which it fails to generate a good approximation. Section 7 presents our conclusions, makes a brief qualitative comparison of our approach to some more standard ones, and outlines some plans for future work.

Regression methods were described only briefly, and not implemented, by Koehler and Owen [8]. Owen [12] describes quasi-regression for Latin hypercube samples and Efromovich [6] proposes a version using orthogonal series of functions on $[0, 1]$. Owen [13] uses quasi-regression to assess how nearly linear some high dimensional functions are.

2. NOTATION

The statistical approaches to approximation begin with an equation

$$f(x) = \sum_{j=1}^{p} z^j(x)\beta_j + \eta(x). \tag{2}$$

Here $z^j(x)$ are basis functions chosen to satisfy:

$$z^1(x) = 1, \quad \forall x \in [0,1]^s \tag{3}$$

$$\int z^j(x)dx = 0, \quad j \geq 1 \tag{4}$$

$$\int z^j(x)z^k(x)dx = 1, \quad \text{if } j = k \tag{5}$$

$$\int z^j(x)z^k(x)dx = 0, \quad \text{if } j \neq k, \tag{6}$$

where all integrals are understood to be over $x \in [0,1]^s$. The β_j are scalar coefficients described below, and $\eta(x)$ is an error function defined by subtraction in (2). In our examples we take the s dimensional basis functions to be tensor products of univariate basis functions, and we use low order orthogonal polynomials for the latter. The theoretical presentation does not assume that these particular basis functions have been chosen. Alternatives such as sinusoids, wavelets, and orthogonalized B-splines may be more appropriate for some settings.

We write $z_i = (z^1(x_i), \ldots, z^p(x_i))$ for the row vector of all p basis functions evaluated at the i'th input point, and Z for the n by p matrix with i'th row z_i. Similarly $Y_i = f(x_i)$ and Y denotes the column vector with i'th entry Y_i.

The kriging approach typically begins with a model in which $\eta(x)$ is the realization of a stationary Gaussian process under which $E(\eta(x)) = 0$ for all x, and $E(\eta(x)\eta(x')) = \sigma^2\Gamma(x - x')$, for a correlation function Γ. The coefficients β_j are also jointly normally distributed independently of η. Now suppose that $x_0 \in [0,1]^s$ and we wish to predict a value for $f(x_0)$. Under the kriging model, the function values $f(x_0), f(x_1), \ldots, f(x_n)$ have a $n+1$ dimensional multivariate normal distribution. The natural way to predict $f(x_0)$ is by the conditional expectation $\hat{f}(x_0) = E(f(x_0) \mid f(x_1), \ldots, f(x_n))$. Under mild continuity conditions on Γ (to eliminate the "nugget effect"), the function $\hat{f}(x)$ smoothly interpolates the given data.

In the limit as the prior variance of every β_j tends to infinity, the kriging estimator yields the interpolator

$$\hat{f}(x_0) = z_0\hat{\beta} + v_0^T V^{-1}(Y - Z\hat{\beta}), \tag{7}$$

where V is the n by n matrix with i, j element $\Gamma(x_i - x_j)$, v_0 is the column vector with i'th element $\Gamma(x_i - x_0)$, and $\hat{\beta} = (Z^T V^{-1} Z)^{-1} Z^T V^{-1} Y$.

The usual practice in computer experiments is to take $p = 1$. The correlation function Γ is typically taken to be a tensor product of univariate correlation functions. The function Γ is commonly a member of a parametric family $\{\Gamma_\theta \mid \theta \in \Theta \subseteq \mathrm{R}^t\}$ with $t = O(s)$. The parameter θ is then chosen on the basis of the sample function values.

The regression based approaches to computer experiments described here, are defined through the least squares values for β,

$$\beta^* = \arg\min_\beta \int (f(x) - z(x)\beta)^2 dx. \tag{8}$$

Elementary manipulations give

$$\begin{aligned} \beta^* &= \left[\int z(x)^T z(x) dx\right]^{-1} \int z(x)^T f(x) dx & (9) \\ &= \int z(x)^T f(x) dx, & (10) \end{aligned}$$

by orthogonality of the basis functions. Notice in particular that

$$\beta^{*1} = \int f(x) dx$$

is simply the integral of f over $[0, 1]^s$.

The regression approach is to take a simple independent Monte Carlo sample $x_1, \ldots, x_n \sim U[0, 1]^s$, and estimate the integrals in (9) by their sample values. This results in

$$\hat{\beta} = (Z^T Z)^{-1} Z^T Y \tag{11}$$

and the approximation is $\hat{f}(x) = z(x)^T \hat{\beta}$.

The quasi-regression approach exploits the known value $\int z(x)^T z(x) dx = I$, estimating β^* by a sample version of equation (10),

$$\widetilde{\beta} = \frac{1}{n} Z^T Y \tag{12}$$

and approximating by $\widetilde{f}(x) = z(x)^T\widetilde{\beta}$. The name quasi-regression is adopted for this because a similar "ignore the denominator" rule leads to quasi-interpolation. See Chui and Diamond [3].

In both regression and quasi-regression, small estimated coefficients β_i might be set to zero in order to speed up evaluation.

Regression and quasi-regression estimate β^* and use 0 for $\eta(x)$. Kriging, by contrast, uses a very flexible approximation to $\eta(x)$ and a minimal model for $z\beta$.

In the regression approaches, β^* is estimated via numerical integration. The global accuracy of such an approximation may also be expressed in terms of numerical integration as $\int(f(x) - z(x)\beta)^2 dx$.

For a fixed value of n and p, Owen [13] gives an analysis that suggests regression with n observations should ordinarily have approximately the accuracy of quasi-regression with λn observations with λ ordinarily larger than 1. But regression requires $O(p)$ times as much time and $O(p)$ times as much space as quasi-regression. For a given computational budget, quasi-regression can use either a larger value of n or a larger value of p than regression. This is why we choose to focus on quasi-regression. We also expect that improvements in quasi-regression described in Section 7 will reduce the advantages regression might enjoy at fixed sample sizes n.

3. COMPLEXITY

The time to fit the kriging model includes components proportional to n^3 and to p^3, arising from the need to solve systems of n and p equations respectively. This is the general rule, though there are special settings and approximations that can reduce the effort. See Ritter [15] for references.

In computer experiments, it is typical that $p = 1$, and the $O(n^3)$ portion of the cost dominates the fitting. This cost grows much more quickly than the cost of obtaining $f(x_1), \ldots, f(x_n)$. Koehler and Owen [8] present the following example. Suppose that a computer experiment takes one hour to compute $f(x_1), \ldots, f(x_n)$, and then one minute is spent on the computer algebra to construct the kriging approximation. The minute might be spent evaluating candidates for the covariance function $\sigma^2\Gamma$. If it emerges that more data is required, then the user might decide to run the experiment for 24 hours. The algebra would then scale to 24^3 minutes, or 9.6 days. The result is that for large n, the algebra takes over the computations. Kriging also faces numerical problems in that the matrix V becomes badly conditioned with increasing n.

Kriging is well established in applications with functions f that are slow to evaluate and are defined over small to moderate dimensions. In such cases n must be small, and a small n has a chance of being effective. We are motivated by problems with faster functions f not necessarily defined

on small dimensions. Faster functions allow sample sizes in the range $10^5 \leq n \leq 10^7$ (or larger) and such large sample sizes may be required when s is not small. In such cases kriging becomes infeasible.

The costs for fitting regression are only $O(np^2)$ and those for fitting quasi-regression are $O(np)$. Problems with large p will ordinarily require large n, so it is natural to consider p and n increasing together. But as long as $p = o(n)$, the rate favors regression and quasi-regression over kriging, for large problems.

While in most cases estimation time is the dominant cost, regression has more favorable space complexity than kriging, and quasi-regression is more favorable still. Regression and quasi-regression also have an advantage in prediction complexity.

TABLE 1.
Time and space complexity of kriging, regression, and quasi-regression, assuming n data points, p basis functions and r nonzero coefficients.

Estimation Complexity	Time T	Space S	Footprint $T \times S$
Kriging	$O(n^3 + p^3)$	$O(n^2 + p^2)$	$O(n^5 + p^5)$
Regression	$O(np^2 + p^3)$	$O(p^2)$	$O(np^4 + p^5)$
Quasi-Regression	$O(np)$	$O(p)$	$O(np^2)$
Prediction Complexity	**Time T**	**Space S**	**Footprint $T \times S$**
Kriging	$O(n + r)$	$O(n + r)$	$O(n^2 + r^2)$
Regression	$O(r)$	$O(r)$	$O(r^2)$
Quasi-Regression	$O(r)$	$O(r)$	$O(r^2)$

Table 1 shows the time and space complexity for estimation and prediction of these three statistical methods. In specialized settings, the cost may be proportional to how long an amount of memory is held. This is described by the "footprint" column in Table 1.

4. IMPLEMENTATION

This section describes some implementation details in quasi-regression. Let

$$\widetilde{\beta}_j^{(n)} = \frac{1}{n} \sum_{i=1}^{n} z^j(x_i) f(x_i) \tag{13}$$

be the quasi-regression estimate of β_j^* based on $x_1, \ldots, x_n$, and let $\widetilde{\beta}^{(n)}$ be the row vector with j'th element $\widetilde{\beta}_j^{(n)}$. The quantity

$$S_j^{(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(z^j(x_i)f(x_i) - \widetilde{\beta}_j^{(n)}\right)^2 \tag{14}$$

can be used to estimate the sampling uncertainty in $\widetilde{\beta}_j^{(n)}$. If $\int f(x)z^j(x)^2dx < \infty$, then the expected value of $S_j^{(n)}/(n-1)$ is equal to the variance of $\widetilde{\beta}_j^{(n)}$ under ordinary Monte Carlo sampling.

Both of these quantities can be updated simultaneously via

$$\widetilde{\beta}_j^{(n)} \equiv \widetilde{\beta}_j^{(n-1)} + \frac{1}{n}\left[z^j(x_i)f(x_i) - \widetilde{\beta}_j^{(n-1)}\right] \tag{15}$$

$$S_j^{(n)} \equiv S_j^{(n-1)} + \frac{n-1}{n}\left[z^j(x_i)f(x_i) - \widetilde{\beta}_j^{(n-1)}\right]^2. \tag{16}$$

The significance of updating formulas (15) and (16) is that they require only a single pass over the data, and are numerically stable, as described by Chan, Golub, and Leveque [2].

Given a vector β not necessarily equal to β^*, the accuracy of $z(x)\beta$ as an approximation to $f(x)$ may be judged through the integral $\int(f(x) - z(x)\beta)^2dx$. We estimate the accuracy of our approximation via

$$\mathrm{Err}_{n,B} = \frac{1}{B}\sum_{i=n-B+1}^{n-B}(f(x_i) - z(x_i)\widetilde{\beta}^{(i-1)})^2. \tag{17}$$

Because x_i is independent of $\widetilde{\beta}^{(i-1)}$,

$$E\left((f(x_i) - z(x_i)\widetilde{\beta}^{(i-1)})^2 \mid \widetilde{\beta}^{(i-1)}\right) = \int(f(x) - z(x)\widetilde{\beta}^{(i-1)})^2dx$$

providing an unbiased estimate of the accuracy of $z(x)\widetilde{\beta}^{(i-1)}$. The accuracy of $z(x)\widetilde{\beta}^{(i-1)}$ changes as i increases, and the quantity $\mathrm{Err}_{n,B}$ estimates the average accuracy over the most recent block of B observations.

It is natural to normalize $\mathrm{Err}_{n,B}$ by an estimate of $\int(f(x) - \beta^{*1})^2dx$. Letting $\bar{f} = \widetilde{\beta}_1^{(n)} = (1/n)\sum_{i=1}^{n} f(x_i)$, the quantity

$$\mathrm{Lof}_{n,B} = \frac{\mathrm{Err}_{n,B}}{(1/n)\sum_{i=1}^{n}(f(x_i) - \bar{f})^2} \tag{18}$$

describes the fraction of the variance in $f(x)$ not explained by the quasi-regression model. This fraction can, in unfavorable cases, exceed 1.0. When

this happens, the interpretation is that a simple model predicting the function by its global average is more accurate than $\widetilde{f}(x) = z(x)\widetilde{\beta}$.

5. TENSOR PRODUCT BASES

We construct our basis functions over $[0,1]^s$ by taking tensor products of univariate basis functions. Let $\phi_0(z) = 1$ for all $z \in [0,1]$. For integers $j \geq 1$, let $\phi_j(z)$ satisfy $\int_0^1 \phi_j(z)dz = 0$, $\int_0^1 \phi_j^2(z)dz = 1$, and $\int_0^1 \phi_j(z)\phi_k(z)dz = 0$, for $j \neq k$. An s dimensional tensor product basis function over $x = (x^1, \ldots, x^s) \in [0,1]^s$ is then

$$\Phi_{r_1,\ldots,r_s}(x) = \prod_{j=1}^{s} \phi_{r_j}(x^j). \tag{19}$$

It is easy to see that $\Phi_{0,0,\ldots,0}(x) = z^1(x) = 1$, and that any finite set of functions $\Phi_{r_1,\ldots,r_s}$ including $\Phi_{0,0,\ldots,0}$ can serve to define the functions $z^j(x)$ described in Section 2. The sets that we choose to work with are defined by vectors $r = (r_1, \ldots, r_s)$ of nonnegative integers satisfying all of

$$\sum_{j=1}^{s} r_j \;\leq\; d \tag{20}$$

$$\sum_{j=1}^{s} 1_{r_j \neq 0} \;\leq\; w \tag{21}$$

$$\max_{1 \leq j \leq s} r_j \;\leq\; m. \tag{22}$$

We refer to these as the degree, rank, and order of $(r^1, \ldots, r^s)$ respectively. The bounds d, w, and m can be varied to suit the problem at hand.

The univariate basis functions we have chosen to work with are orthogonal polynomials on $z \in [0,1]$. Using the shorthand $u = u(z) \equiv z - 1/2$,

the first few polynomials are:

$$\begin{aligned}
\phi_0(z) &= 1 \\
\phi_1(z) &= \sqrt{12}u \\
\phi_2(z) &= \sqrt{180}\left[u^2 - \frac{1}{12}\right] \\
\phi_3(z) &= \sqrt{2800}\left[u^3 - \frac{3}{20}u\right] \\
\phi_4(z) &= 210\left[u^4 - \frac{3}{14}u^2 + \frac{3}{560}\right] \\
\phi_5(z) &= 252\sqrt{11}\left[u^5 - \frac{5}{18}u^3 + \frac{5}{336}u\right] \\
\phi_6(z) &= 924\sqrt{13}\left[u^6 - \frac{15}{44}u^4 + \frac{5}{176}u^2 - \frac{5}{14784}\right].
\end{aligned}$$

These are essentially the Legendre polynomials, except that the latter are defined over $[-1, 1]$ instead of $[0, 1]$.

6. EXAMPLES

This section considers 4 example functions: The borehole function of equation (1), a robot arm function widely used in neural network papers, a 9 dimensional function with 2 spikes, and a function from Chemical Vapor Deposition (CVD).

6.1. Borehole function

The borehole function of equation (1) was investigated over the following ranges:

$$\begin{aligned}
r_w &\in [0.05, 0.15]\ m \\
r &\in [100, 50000]\ m \\
T_u &\in [63070, 115600]\ m^3/\text{yr} \\
T_l &\in [63.1, 116]\ m^3/\text{yr} \\
H_u &\in [990, 1110]\ m \\
H_l &\in [700, 820]\ m \\
L &\in [1120, 1680]\ m \\
K_w &\in [9855, 12045]\ m/\text{yr}.
\end{aligned}$$

The first set of basis functions we considered for this model have degree, rank and order $d = 4$, $w = 2$, and $m = 4$. For $s = 8$ this results in $p = 201$

basis functions. Figure 1 shows $\mathrm{Lof}_{B,n}$ versus n, for $B = 100$. A simple model using the input variables one or two at a time explains roughly 99% of the variance of this function. Because the $\mathrm{Lof}_{B,n}$ still appears to be decreasing at $n = 10000$ it is possible that even more than 99% of the variance is explained by this model.

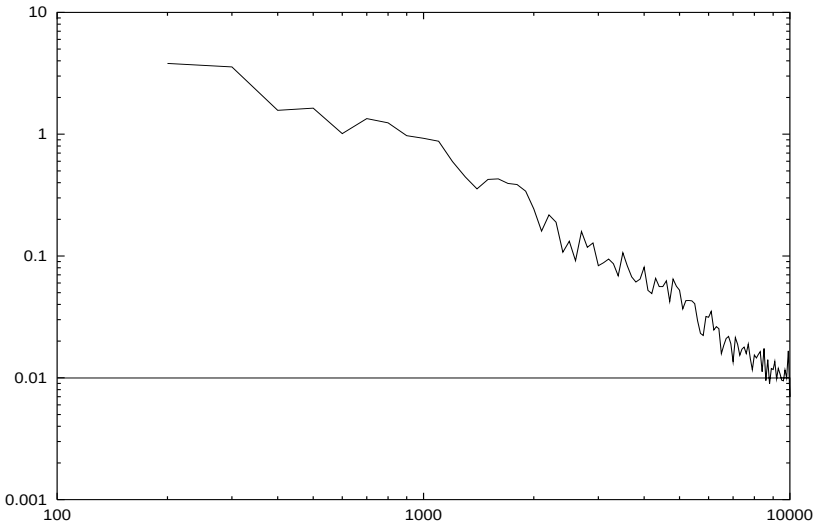

FIG. 1. Lack of fit is plotted versus sample size for the borehole function. The model used degree $d = 4$, rank $w = 2$ and order $m = 4$. There are $p = 201$ basis functions over $s = 8$ dimensions.

Figure 2 shows the same information as Figure 1, except that w has been increased from 2 to 3. This increases p from 201 to 425. The lack of fit has increased from about 1% to about 3.4%, but is still decreasing by $n = 10000$. The eventual lack of fit has to be smaller for this basis than for the one with rank 2, though for finite n, sampling fluctuations in $\widetilde{\beta}$ will increase the lack of fit, and the effect is worse for this example because p is larger.

Figure 3 shows the same information, except that now the degree is increased to $d = 6$. This basis has $p = 1517$ basis functions. With this many basis functions the lack of fit is still decreasing at $n = 100000$.

Using quasi-regression, we can infer that the borehole function is very nearly a sum of its input variables one or two at at time. Each of the three example runs gives a usable model that approximates the borehole function with small errors. The gain from using 1517 basis functions and 100000 observations to fit them, instead of using the smaller model from Figure 1

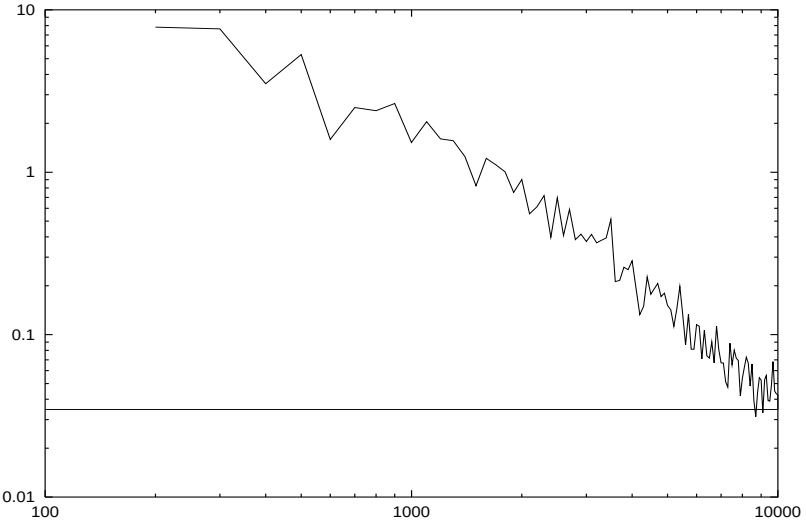

FIG. 2. Lack of fit is plotted versus sample size for the borehole function. The model used degree $d = 4$, rank $w = 3$ and order $m = 4$. There are $p = 425$ basis functions over $s = 8$ dimensions.

is small enough, that one might prefer the original approximation, or an even smaller one, in practice.

6.2. Robot arm function

A function commonly used in the neural network literature is the robot arm function. Consider a robot arm with 4 segments. The shoulder of the arm is fixed at the origin in the (u, v)-plane. The segments of this arm have lengths L_1, L_2, L_3, and L_4. The first segment is at angle θ_1 with respect to the horizontal coordinate axis of the plane. Angles describe the rotation of For $k = 2, 3, 4$, segment k makes angle θ_k with segment $k - 1$. The end of the robot arm is at

$$\begin{aligned} u &= \sum_{j=1}^{4} L_j \cos\left(\sum_{k=1}^{j} \theta_k\right) \\ v &= \sum_{j=1}^{4} L_j \sin\left(\sum_{k=1}^{j} \theta_k\right) \end{aligned}$$

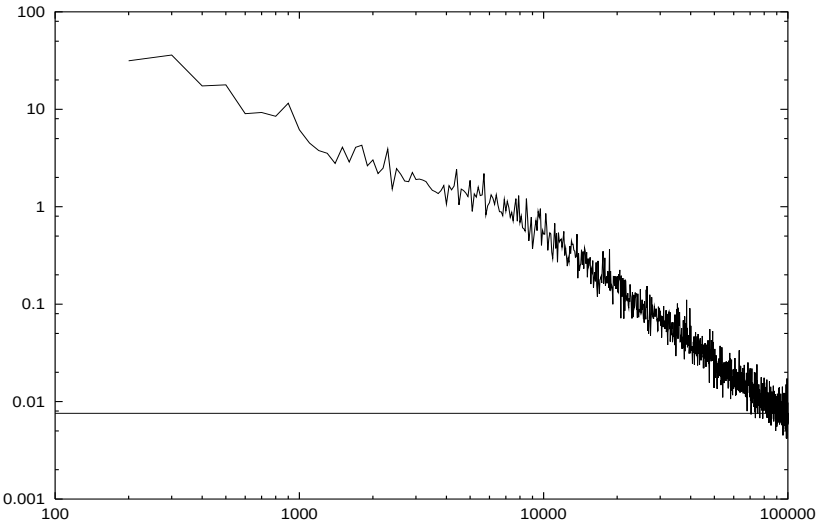

FIG. 3. Lack of fit is plotted versus sample size for the borehole function. The model used degree $d = 4$, rank $w = 2$ and order $m = 4$. There are $p = 1517$ basis functions over $s = 8$ dimensions.

and the response f is the distance $(u^2 + v^2)^{1/2}$ from the end of the arm to the origin expressed as a function of 8 variables θ_j ranging over $[0, 2\pi]$ and L_j ranging over $[0, 1]$.

Figure 4 shows the lack of fit for this function, using $d = 4$, $w = 3$, and $m = 4$, which for $s = 8$ gives $p = 425$ basis functions. The lack of fit decreases to about 29.5% by $n = 10000$ and does not decrease much further as n increases to 100000. Unlike the borehole function, the robot arm function is not well approximated by a low order polynomial. We know by Taylor's theorem that over a small domain the robot arm function would be well approximated by a low order polynomial, so this result may also be interpreted as a statement that the chosen domain is too large for such a local approximation.

Figure 5 shows lack of fit versus n for a larger basis with $d = 12$, $w = 3$, and $m = 4$, which for $s = 8$ gives $p = 4065$ basis functions. While the lack of fit is still decreasing by $n = 100000$, it is still as large as 19.2%, suggesting that simply adding basis functions has not helped much.

Polynomial basis functions do not seem to be well suited for the robot arm function, over such a large range. Some failures of this type are inevitable for a high dimensional approximation method, but at least the quasi-regression method gives a clear indication of such a failure having

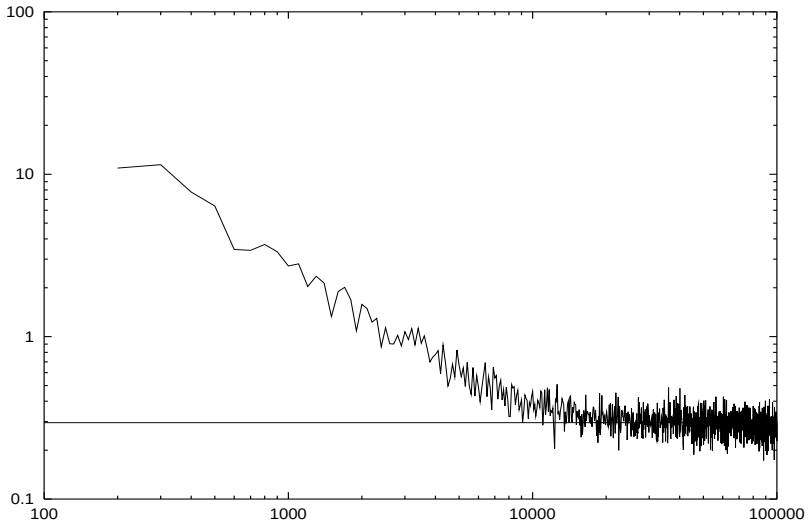

FIG. 4. Lack of fit is plotted versus sample size for the robot arm function. The model used degree $d = 4$, rank $w = 3$ and order $m = 4$. There are $p = 425$ basis functions over $s = 8$ dimensions.

happened. This could lead an investigator to try a different basis. Perhaps one based on trigonometric polynomials (at least for the θ_j, if not the L_j) would work better.

6.3. Chemical vapor deposition

Our next example is for a problem in Chemical Vapor Deposition (CVD) brought to our attention by Juan Meza and Charles Tong of Sandia National Laboratory. CVD is used to deposit a chemical on the surface of a silicon wafer for use in making integrated circuits. The wafers are heated in an oven, and the vapor is allowed to pass over them. The rate of deposition depends on the temperature of the wafers. Other things being equal it is best to have nearly uniform wafer temperature, in order to get a chemical layer of nearly uniform thickness. A computer code implements a model for the temperature field within the oven as a function of the locations and settings of the heating elements. The response function f is a measure of the uniformity of the surface temperatures of the wafers. The code is available in versions ranging from $s = 3$ to $s = 24$ depending on how much detail is used. The $s = 3$ dimensional version takes about 1 second to execute on a modern workstation.

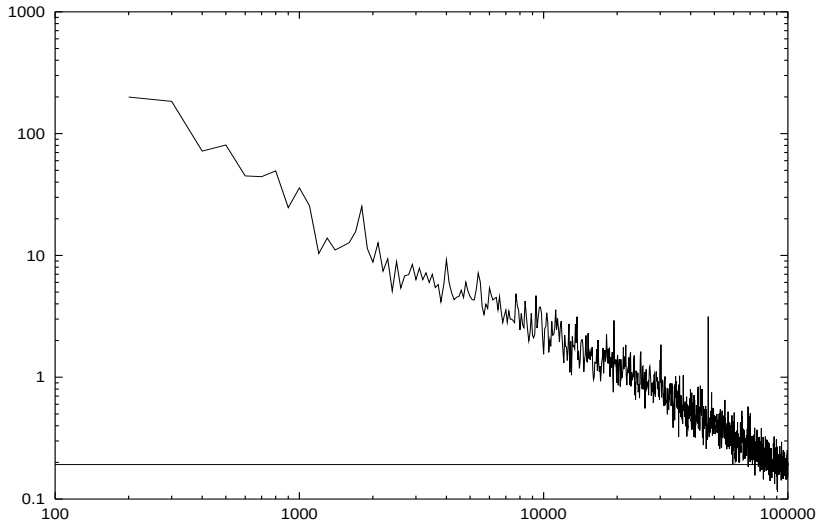

FIG. 5. Lack of fit is plotted versus sample size for the robot arm function. The model used degree $d = 12$, rank $w = 3$ and order $m = 4$. There are $p = 4065$ basis functions over $s = 8$ dimensions.

Figure 6 plots lack of fit versus n for a basis with $d = 4$, $w = 2$, and $m = 4$, leading to $p = 31$ basis functions. The lack of fit decreases to about 0.06% by $n = 50000$ and does not appear to be decreasing much at that point.

Figure 7 shows the results for a larger model having $d = 12$, $w = 3$, and $m = 4$, leading to $p = 125$. The result is only a small improvement in the lack of fit. It is possible to save the 50000 function evaluations and simply regenerate the random inputs x_i, so that evaluating a second model need not take another 50000 seconds (almost 14 hours). For many purposes the simple approximation using only 31 basis functions is a sufficiently accurate approximation to the original function. This represents a substantial speed-up of the function, and may be fast enough to support interactive visualization. The original function, while fast, would not be fast enough to have 100 evaluations take place at the click of a mouse.

6.4. Spiky function

Our final example is another negative one. The function is taken from the dissertation of Zhou [17] who considers numerical integration of spiky

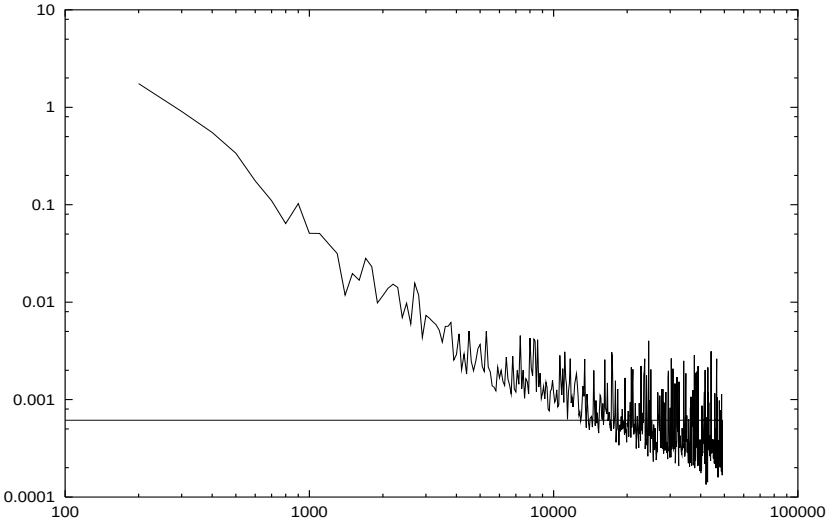

FIG. 6. Lack of fit is plotted versus sample size for the CVD function. The model used degree $d = 4$, rank $w = 2$ and order $m = 4$. There are $p = 31$ basis functions over $s = 3$ dimensions.

functions. This spike function is

$$f(x) = \frac{10^s}{2}\Big(\varphi(10(x - 1/3)) + \varphi(10(x - 2/3))\Big), \tag{23}$$

where $\varphi(x) = (2\pi)^{-s/2}\exp(-.5\|x\|^2)$ with the operation $10(x - 1/3)$ interpreted component-wise on x, and $\|\cdot\|$ denoting the Euclidean norm. This function is a sum of two narrow Gaussian probability densities centered at $(1/3, \ldots, 1/3)$ and at $(2/3, \ldots, 2/3)$. We chose to investigate it in $s = 9$ dimensions. Truncating the function to the unit cube makes its integral slightly smaller than 1.

There is no reason to expect this function to be approximately a low order polynomial. Figure 8 shows the lack of fit using $d = 4$, $w = 2$ and $m = 4$ (with $p = 253$), and Figure 9 shows the lack of fit using $d = 6$, $w = 3$ and $m = 4$ (with $p = 2185$). In both cases the lack of fit fails to become small, and is in fact larger than 1. The spikes in $f(x)$ show themselves as spikes in the lack of fit curve. The reason is that each point in the lack of fit curve is an average of $B = 100$ squared error estimates, normalized by the function variance. Most of the blocks of function values include no spikes, and produce small error values. But many of the blocks do in fact

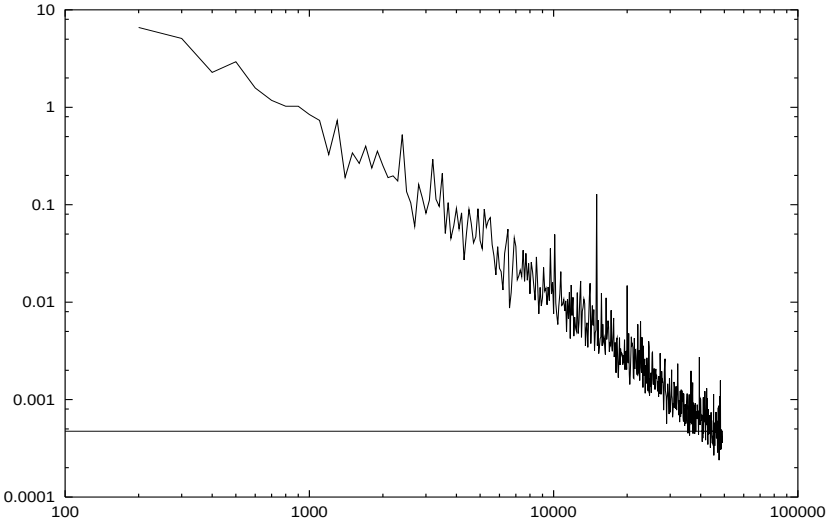

FIG. 7. Lack of fit is plotted versus sample size for the CVD function. The model used degree $d = 12$, rank $w = 2$ and order $m = 4$. There are $p = 125$ basis functions over $s = 3$ dimensions.

contain points x in a spike, and these ones produce very large values of Lof.

7. DISCUSSION

We have found that quasi-regression is workable on some realistic problems, and for sample sizes n that would make kriging infeasible. Our view is that this makes quasi-regression a worthwhile addition to the computer experimenter's toolbox. We have not compared quasi-regression with kriging on problems where both are feasible. We do not expect quasi-regression to perform well in settings where only a few dozen observations can be obtained. In such settings regression is more suitable, and kriging may be more effective still.

Quasi-regression also provides a direct measure of its accuracy, helping the user to decide whether the approximation is good enough. By watching the trajectory of the lack of fit, one can infer whether increasing n is likely to be worthwhile. The trajectory can also give an indication of whether the target function is spiky, and hence likely to be require quite different techniques.

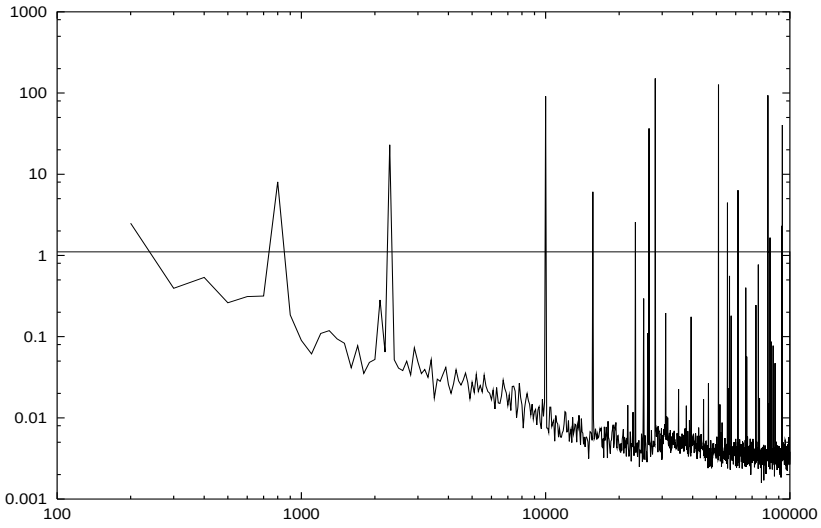

FIG. 8. Lack of fit is plotted versus sample size for the spiky function. The model used degree $d = 4$, rank $w = 2$ and order $m = 4$. There are $p = 253$ basis functions over $s = 9$ dimensions.

Our approach has been very different from the usual one in approximation theory. We have chosen to focus on example target functions individually, instead of on function classes, such as balls in Hilbert spaces. The study of high dimensional numerical integration gained greatly from just such a study of specific example functions, as in Paskov and Traub [14] and Caflisch, Morokoff and Owen [1] and others, and we hope the same will happen for approximation.

Asymptotic theory for function classes suggests that the smoother the class containing f, the better the rate of convergence attainable for it. The constant in front of this rate is usually determined by the radius of the ball of functions. Generally, the functions considered in this paper are very smooth. The CVD function might be an exception; it is not available in closed form and it may have numerical noise. The robot arm function is an exception, only near points where $L_1 = L_2 = L_3 = L_4 = 0$, but there is no reason to expect that raising the minimum value of the L_j slightly would make quasi-regression perform well. The performance differences seen on these functions seem to be more a matter of the leading constants than of the rates.

Our work continues on quasi-regression. The method was designed with the idea that simple Monte Carlo points could be replaced by quasi-Monte

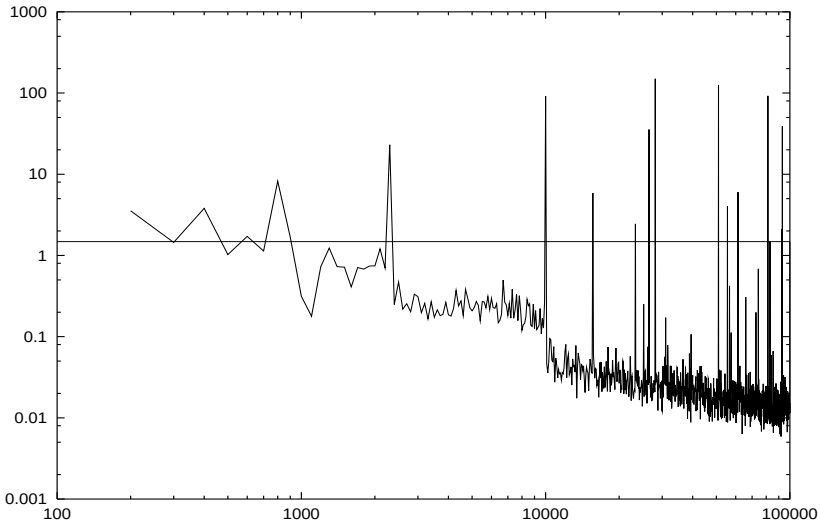

FIG. 9. Lack of fit is plotted versus sample size for the spiky function. The model used degree $d = 6$, rank $w = 3$ and order $m = 4$. There are $p = 2185$ basis functions over $s = 9$ dimensions.

Carlo (Niederreiter [10]) or by randomized quasi-Monte Carlo (Owen [11]) points. There is room for more sophisticated statistical estimation of β_j, such as shrinking $\widetilde{\beta}_j$ towards zero if S_j is large, and using ordinary regression for some but not all of the β_j. The dissertation of An (in progress) considers extended quasi-regression which replaces $f(x_i)$ by $f(x_i) - \sum_{k \in \mathcal{K}_j} \beta^{(i-1)} z_i^k$ in (13) where $j \notin \mathcal{K}_j$. Extended quasi-regression can reduce the variance of the estimated coefficients.

ACKNOWLEDGMENT

A talk based on this material was presented at Complexity'99. We thank the organizers Fred Hickernell and Henryk Wozniakowski, the sponsors, and the hosts at Hong Kong Baptist University for their efforts. We also thank Juan Meza and Charles Tong of Sandia Laboratories for providing the CVD example function. Finally, we thank the National Science Foundation of the U.S., for supporting this work under grant DMS-9704495.

REFERENCES

1. Russel E. Caflisch, William Morokoff, and Art B. Owen. Valuation of mortgage backed securities using brownian bridges to reduce effective dimension. *Journal of Computational Finance*, 1:27–46, 1997.

2. Tony F. Chan, Gene H. Golub, and Randall J. LeVeque. Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician*, 37:242–247, 1983.

3. Charles K. Chui and H. Diamond. A natural formulation of quasi-interpolation by multivariate splines. *Proceedings of the American Mathematical Society*, 99(4):643–646, 1987.

4. Carla Currin, Toby Mitchell, Max Morris, and Don Ylvisaker. Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments. *Journal of the American Statistical Association*, 86:953–963, 1991.

5. Persi Diaconis. Bayesian numerical analysis. In *Statistical Decision Theory and Related Topics IV, in two volumes*, volume 1, pages 163– 176, 1988.

6. Sam Efromovich. On orthogonal series estimators for random design nonparametric regression. In *Computing Science and Statistics. Proceedings of the 24rd Symposium on the Interface*, pages 375– 379, 1992.

7. Andre G. Journel and Charles J. Huijbregts. *Mining Geostatistics*. Academic, 1979.

8. James Koehler and Art Owen. Computer experiments. In S. Ghosh and C.R. Rao, editors, *Handbook of Statistics, 13: Design and Analysis of Experiments*, pages 261–308. North-Holland, 1996.

9. Max D. Morris, Toby J. Mitchell, and Donald Ylvisaker. Bayesian design and analysis of computer experiments: Use of derivatives in surface prediction. *Technometrics*, 35:243–255, 1993.

10. Harald Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. S.I.A.M., Philadelphia, PA, 1992.

11. A. B. Owen. Scrambled net variance for integrals of smooth functions. *Annals of Statistics*, 25(4):1541–1562, 1997.

12. Art B. Owen. A central limit theorem for Latin hypercube sampling. *Journal of the Royal Statistical Society, Series B, Methodological*, 54:541–551, 1992.

13. Art B. Owen. Detecting near linearity in high dimensions. Technical report, Stanford University, Statistics Department, 1998.

14. Spassimir Paskov and Joseph Traub. Faster valuation of financial derivatives. *The Journal of Portfolio Management*, 22:113–120, 1995.

15. Klaus Ritter. *Average Case Analysis of Numerical Problems*. PhD thesis, University of Erlangen.

16. Jerome Sacks, William J. Welch, Toby J. Mitchell, and Henry P. Wynn. Design and analysis of computer experiments (c/r: P423-435). *Statistical Science*, 4:409–423, 1989.

17. Yi Zhou. *Adaptive Importance Sampling for Integration*. PhD thesis, Stanford University, 1998.