# Quasi-regression with shrinkage ⋆

Tao Jiang [1], Art B. Owen [*,1]

*Department of Statistics, Stanford University, Stanford CA 94305, USA*

**Abstract**

Quasi-regression is a method of Monte Carlo approximation useful for global sensitivity analysis. This paper presents a new version, incorporating shrinkage parameters of the type used in wavelet approximation. As an example application, a black box function from machine learning is analyzed. That function is nearly a sum of functions of one and two variables and the first variable acting alone accounts for more than half of the variance.

*Key words:* computer experiments, global sensitivity analysis, machine learning, wavelets
*PACS:* 02.60.Gf, 02.60.-x, 02.70.Lq

## 1 Introduction

Quasi-regression is a Monte Carlo method for constructing approximations to functions. The approximations can serve many purposes. They may be used as rapidly computable surrogates for visualization or for optimization. In this article we use them in a kind of global sensitivity analysis to get qualitative information about a "black box" function of interest.

⋆ The talk by Owen at MCM 2001 in Salzburg Austria surveyed quasi-regression as developed in papers [1–3]. We thank the organizers for permitting us to write here about some new directions for quasi-regression.

* Corresponding author.
*URLs:* `http:www-stat.stanford.edu/~jiang` (Tao Jiang), `http:www-stat.stanford.edu/~owen` (Art B. Owen).

[1] Supported by NSF grant DMS-0072445

Quasi-regression is based on observations sampled from the domain of the function $f$ to be approximated. The function $f$ is expanded in an infinite sum of coefficients times basis functions. A finite subset of the coefficients are then estimated using sample values of $f$. A natural approach is least squares regression of $f$ on $p$ of the basis functions using $n$ observations. The computational cost of regression is typically $O(np^2)$ due to the need to solve a system of $p$ equations in $p$ unknowns. In quasi-regression the cost can be held to $O(np)$ by exploiting the known integrals of products of basis functions. As a result, larger values of $n$ and much larger values of $p$ become computationally feasible.

An and Owen [1] present quasi-regression, and give some illustrative examples. Owen [2] uses quasi-regression to assess the degree of linearity in high dimensional functions. In one example the degree of linearity of a $1,000,000$ dimensional function is estimated to within a few percent, with only $100,000$ function values. Lemieux and Owen [3] apply quasi-regression to study the effective dimension of some integrands. The name is in honor of quasi-interpolation [4] where a similar idea is applied to avoid solving a large system of equations.

The main innovation we present here is the use of shrinkage coefficients in quasi-regression. Shrinkage coefficients are widely used when fitting wavelet models to noisy data. See for example Donoho and Johnstone [5]. Efromovich [6] mixes quasi-regression with regression and some shrinkage for orthogonal series on the unit interval in one dimension. The idea underlying quasi-regression was proposed for computer experiments in [7,8], as a means of avoiding some $O(n^3)$ computations that can arise in the kriging approach of [9–11] and others.

Saltelli, Chan and Scott [12] provide an up to date survey of sensitivity analysis methods including global sensitivity analysis which goes well beyond the usual methods based on partial derivatives. In a series of papers, Sobol' and co-authors [13–16] develop unbiased Monte Carlo methods of estimating global sensitivity measures expressed as variances of certain analysis of variance (ANOVA) component functions. Quasi-regression methods have the advantage of yielding computable approximations to the ANOVA component functions themselves, with the disadvantage that the corresponding ANOVA variances have a truncation bias.

Section 2 presents the notation for quasi-regression with shrinkage (QRS). The tensor product bases we study are described in Section 3. Some elementary properties of the method are recorded in Section 4. Section 5 presents a numerical example. A black box function representing the output of a neural network is approximated. From the approximation we learn that the first input variable is most important with an effect that is monotone decreasing and nearly quadratic. An additive approximation captures roughly 80% of the variance of the function, and a sum of functions of two or fewer of the inputs

captures roughly 98.2% of the variance. Section 6 presents our conclusions.

## 2 Notation

We suppose that interest centers on a real valued function $f \in L^2(0,1)^d$. We suppose further that we have selected a complete orthonormal basis $\{\psi_r \mid r \in \mathsf{U}\}$ for $L^2(0,1)^d$. The index set $\mathsf{U}$ is countably infinite and we assume it contains a special element 0 with $\psi_0(x) = 1$ for all $x$.

In terms of the basis we may write

$$f(x) = \sum_{r \in \mathsf{U}} \beta_r \psi_r(x)$$

where

$$\beta_r = \int_{(0,1)^d} f(x)\psi_r(x)dx.$$

Integrals without an explicit domain are assumed to be over $(0,1)^d$, and index sums without an explicit range are over the whole of $\mathsf{U}$.

The basis functions satisfy $\int \psi_r(x)dx = 1_{r=0}$ and $\int \psi_r(x)\psi_s(x)dx = 1_{r=s}$ by their definition. We assume also that $\int \psi_r(x)^4 dx < \infty$. We will use tensor products of univariate basis functions as described in Section 3.

The quasi-regression of [1] approximates $f(x)$ by $\widetilde{f}(x) = \sum_{r \in \mathsf{R}} \widetilde{\beta}_{r,n}\psi_r(x)$ for a finite set $\mathsf{R}$ containing $p$ indices, using $\widetilde{\beta}_{r,n} = (1/n)\sum_{i=1}^n f(x_i)\psi_r(x_i)$ where $x_i$ are independent $U(0,1)^d$ vectors. The cost is only $O(np)$ to compute $\widetilde{\beta}_{r,i}$ for $r \in \mathsf{R}$ and $1 \le i \le n$.

Uniformity of $x_i$ implies that $E(\widetilde{\beta}_{r,n}) = \beta_r$ and can be used to derive an unbiased estimate $\widehat{\mathrm{Var}}(\widetilde{\beta}_{r,n}) = [n(n-1)]^{-1}\sum_{i=1}^n(\psi_r(x_i)f(x_i) - \widetilde{\beta}_{r,n})^2$ of $\mathrm{Var}(\widetilde{\beta}_{r,n})$. The error $\widetilde{f}(x) - f(x)$ depends on the variance of $\widetilde{\beta}_r$ for $r \in \mathsf{R}$ and also on the values of $\beta_r$ for $r \notin \mathsf{R}$. An and Owen [1] employ statistical methods to estimate $\int(\widetilde{f}(x) - f(x))^2 dx$ from the same function values used to compute $\widetilde{f}$.

As noted in [1], the variance of $\widetilde{\beta}_r$ for $r \neq 0$ is affected by adding a constant $c$ to $f$, even though the value of $\beta_r$ itself is not affected. It is reasonable to expect that $c = -E(f) = -\beta_0$ would be a good choice, but this value must ordinarily be estimated from the same function values used to estimate the quasi-regression. Improvements from estimating $c$ are reported in [2] and [3]. Sobol' [13,14] recommends a similar adjustment when estimating mean squared ANOVA components for global sensitivity analysis.

In this paper we adjust the estimate of $\beta_r$ using estimated values for $\beta_s$ with $s$ not necessarily zero. We also introduce some multiplicative shrinkage factors in $[0,1]$. The estimated coefficients $\widetilde{\beta}_{r,n}$ are shrunk towards zero in order to dampen the effects of their sampling variance.

Our approximations to $f$ take the form

$$\widetilde{f}_n(x) = \widetilde{f}_{n,\gamma}(x) = \sum_r \gamma_{r,n}\widetilde{\beta}_{r,n}\psi_r(x) \tag{1}$$

for $n \geq 1$, for shrinkage coefficients $\gamma_{r,n} \in [0,1]$ and coefficient estimates $\widetilde{\beta}_{r,n} \in \mathbb{R}$. For each $n$, only finitely many of the $\gamma_{r,n}$ are nonzero. The number of $r$ with $\gamma_{r,n} \neq 0$ is denoted by $p_{n,\gamma}$. Whenever $\gamma_{r,n} \neq 0$, the quantity $\widetilde{\beta}_{r,n}$ is an estimate of $\beta_r$ computed from $f(x_i)$ for $1 \leq i \leq n$. The values $\gamma_{r,n}$ can be computed from $x_1, \ldots, x_n$, and do not depend on $x_\ell$ for $\ell > n$.

The estimate $\widetilde{\beta}_{r,n}$ is defined recursively by

$$\widetilde{\beta}_{r,n} = \frac{1}{n}\sum_{i=1}^{n} \psi_r(x_i)\Big(f(x_i) - \sum_{s\neq r} \lambda_{s,i-1}\widetilde{\beta}_{s,i-1}\psi_r(x_i)\Big) \tag{2}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \psi_r(x_i)\Big(f(x_i) - \widetilde{f}_{i-1,\lambda}(x_i) + \lambda_{r,i-1}\widetilde{\beta}_{r,i-1}\psi_r(x_i)\Big), \tag{3}$$

where $\lambda_{s,i-1} \in [0,1]$ may depend on $x_1, \ldots, x_{i-1}$, and $\lambda_{s,0}$ is specified before any $x_i$ are sampled. For any $n \geq 0$, only $p_{n,\lambda} < \infty$ of the $\lambda_{s,n}$ are nonzero. The function $\widetilde{f}_{n,\lambda}$ is defined similarly to $\widetilde{f}_{n,\gamma}$ except that $\lambda_{r,n}$ replaces $\gamma_{r,n}$. We will take $\lambda_{s,0} = \widetilde{\beta}_{s,0} = 0$.

Our algorithms use two sets of shrinkage coefficients, $\lambda_{r,n}$ for use in the estimates $\widetilde{\beta}_{r,n}$, and $\gamma_{r,n}$ for the approximation $\widetilde{f}_n$. In practice one might use $\lambda_{r,n} = \gamma_{r,n}$. Quasi-regression of [1] has $\lambda_{r,n} = 0$ and $\gamma_{r,n} = 1_{r\in\mathsf{R}}$ for a fixed set $\mathsf{R}$ of $p$ basis functions. Taking $\lambda_{r,n} = 1_{r=0}$ makes an estimated adjustment to $\widetilde{\beta}_{r,n}$ for the constant value $\beta_0$. Some unpublished computations by Jian An show that taking $\lambda_{r,n} = 1_{r\in\mathsf{R}}$ for small $n$ can produce very unstable results. Highly variable values of $\widetilde{\beta}_{r,n-1}$ with $\lambda_{r,n-1} = 1_{r\in\mathsf{R}}$ can lead to even more highly variable values of $\widetilde{\beta}_{r,n}$, so some shrinkage is essential.

## 3 Tensor product bases

The basis functions we use are built as products of the univariate basis functions described here. For $j = 1, \ldots, d$ let $\{\phi_{jk}, k \geq 0\}$ be a complete orthonormal basis of $L^2(0,1)$. Then $\int_0^1 \phi_{jk}(x)\phi_{j\ell}(x)dx = 1_{k=\ell}$, and we also suppose that $\phi_{j0}(x) = 1$ for $0 < x < 1$ and $j = 1, \ldots, d$. The index $j$ allows us to use a

different basis for each input variable, when that is helpful, though usually a common basis $\phi_{jk} = \phi_k$ is used for all input variables.

Let $r = (r_1, \ldots, r_d)$ be a vector of $d$ nonnegative integers, and denote the $d$ components of $x$ by $x^j$ for $j = 1, \ldots, d$. To each such $r$ there corresponds a unique tensor product function

$$\psi_r(x) = \prod_{j=1}^{d} \phi_{jr_j}(x^j). \tag{4}$$

Our index set is $\mathsf{U} = \{(r_1, \ldots, r_d) \mid r_j \in \mathbb{Z}, r_j \geq 0\}$. The function $\psi_0$ refers to $\psi_r$ with $r = (0, \ldots, 0)$.

Sets $\mathsf{R}$ composed of the smaller index values $r$ are useful in approximation. There are several ways to measure the size of $r$, including

$$\|r\|_0 = \sum_{j=1}^{d} 1_{r_j > 0}, \quad \|r\|_1 = \sum_{j=1}^{d} r_j, \quad \text{and} \quad \|r\|_\infty = \max_{1 \leq j \leq d} r_j$$

called the rank, degree, and order, respectively, in [1]. We let

$$\mathsf{R}_{B_0, B_1, B_\infty} = \Big\{ \psi_r \mid \|r\|_0 \leq B_0,\ \|r\|_1 \leq B_1,\ \|r\|_\infty \leq B_\infty \Big\}.$$

An expansion in a tensor product basis provides some interpretable quantities that can be estimated by quasi-regression. The integral of $f$ is $\int f(x)dx = \beta_0$. The variance of $f$ is $\int [f(x) - \beta_0]^2 dx = \sum_{r \neq 0} \beta_r^2$. For $u \subseteq \{1, 2, \ldots, d\}$, let $\mathsf{R}_u = \{r \mid r_j > 0 \iff j \in u\}$. Then the ANOVA component corresponding to $u$ is $f_u = \sum_{r \in \mathsf{R}_u} \beta_r \psi_u$. The global sensitivity indices of [16] are $\sum_{r \in \mathsf{R}_u} \beta_r^2 / \sigma^2$. The dimension distribution used in [17] to describe effective dimension is $\nu(k) = \sum_{\|r\|_0 = k} \beta_r^2 / \sigma^2$. We may also define a "degree distribution" via $\sum_{\|r\|_1 = k} \beta_r^2 / \sigma^2$ to describe the smoothness of $f$.

A univariate basis can be constructed through orthogonal polynomials. The Legendre polynomials are orthogonal polynomials on $[-1, 1]$. The first four of them are

$$\begin{aligned} L_0(x) &= 1 & L_1(x) &= x \\ L_2(x) &= (3x^2 - 1)/2 & L_3(x) &= (5x^3 - 3x)/2, \end{aligned}$$

and for $k \geq 1$, they satisfy the three term recurrence

$$(k+1)L_{k+1}(x) - (2k+1)xL_k(x) + kL_{k-1}(x) = 0.$$

Legendre polynomials satisfy $\int_{-1}^{1} L_j(x)L_k(x)dx = 1_{j=k} 2/(2k+1)$. For our pur-

poses, orthonormal polynomials on $(0,1)$ can be obtained via

$$\phi_k(x) = \sqrt{2k+1}\,L_k(2x-1), \quad k \geq 0.$$

The Legendre polynomials are defined with respect to a uniform weighting of $[-1,1]$. Hermite and Chebychev polynomials are defined through scaled versions of the normal and Beta(1/2,1/2) distributions respectively. We may construct functions $\phi_k$ orthonormal on $(0,1)$ from a distribution function $F(z) = P(Z \leq z)$, by constructing monomials $(F^{-1}(x))^k$ for $k \geq 0$, orthogonalizing them by Gram-Schmidt, and then normalizing them to have unit variance. The Legendre polynomials can be obtained this way by taking $F$ to be the $U[-1,1]$ distribution. Fourier functions and Haar wavelets also provide convenient univariate basis functions.

## 4 Properties

The computational and distributional properties of the quasi-regression presented here are most simply described through terms

$$\begin{aligned} T_{r,i} &= \psi_r(x_i)\Big(f(x_i) - \sum_s 1_{s\neq r}\lambda_{s,i-1}\widetilde{\beta}_{s,i-1}\psi_s(x_i)\Big) \\ &= \psi_r(x_i)\sum_s\Big(\beta_s - 1_{s\neq r}\lambda_{s,i-1}\widetilde{\beta}_{s,i-1}\Big)\psi_s(x_i). \end{aligned}$$

Then $n\widetilde{\beta}_{r,n} = (n-1)\widetilde{\beta}_{r,n-1} + T_{r,n}$, through which a simple update of $\widetilde{\beta}_{r,n}$ may be defined.

First we show that $\widetilde{\beta}_{r,n}$ is an unbiased estimate of $\beta_r$ for all $r \in \mathsf{U}$. The proof allows considerable flexibility in choosing the shrinkage coefficients $\lambda_{r,n}$.

**Proposition 1** *For $i \geq 1$ let $x_i$ be independent $U(0,1)^d$ random vectors. Let $\lambda_{r,0} = \widetilde{\beta}_{r,0} = 0$ and for $i \geq 1$ let $\lambda_{r,i}$ be real values computed from $x_1, \ldots, x_i$. Let $\widetilde{\beta}_{r,n}$ be defined by (2). Then $E(\widetilde{\beta}_{r,n}) = \beta_r$ for $n \geq 1$ and $r \in \mathsf{U}$.*

**Proof:** Conditionally on $x_1, \ldots, x_{n-1}$ the values $\lambda_{s,n-1}$ and $\widetilde{\beta}_{s,n-1}$ are not random, and so averaging over the uniform distribution of $x_n$,

$$\begin{aligned} E(T_{r,n} \mid x_1, \ldots, x_{n-1}) &= \int \psi_r(x)\Big(f(x) - \sum_s 1_{s\neq r}\lambda_{s,n-1}\widetilde{\beta}_{s,n-1}\psi_r(x)\Big)dx \\ &= \beta_r. \end{aligned}$$

Therefore $E(T_{r,n}) = \beta_r$ and since $\widetilde{\beta}_{r,n}$ is the average of $T_{r,1}$ through $T_{r,n}$ we have $E(\widetilde{\beta}_{r,n}) = \beta_r$. $\square$

Two convenient measures of the accuracy of $\widetilde{f}_n$ are the integrated squared error, and the integrated mean squared error,

$$\text{ISE} = \int [\widetilde{f}_n(x) - f(x)]^2 dx, \quad \text{and,} \quad \text{IMSE} = \int E\Big([\widetilde{f}_n(x) - f(x)]^2\Big) dx,$$

respectively. ISE describes the accuracy of the approximation $\widetilde{f}$ we obtain while IMSE is an average over functions we might have obtained in sampling from the distribution of $x_1, \ldots, x_n$. These errors have simple expressions for quasi-regression over orthonormal bases. We analyze IMSE for the special case of prespecified nonrandom $\gamma_{r,n}$ values.

**Proposition 2** *Let $f \in L^2(0,1)^d$ and let $\widetilde{f}_n$ be defined by (1) where $\gamma_{r,n}$ are computed from $x_1, \ldots, x_n$. Then ISE $= \sum_r (\gamma_{r,n}\widetilde{\beta}_{r,n} - \beta_r)^2$. If $\gamma_{r,n}$ are not random, then IMSE $= \sum_r Var(\widetilde{\beta}_{r,n})\gamma_{r,n}^2 + (1-\gamma_{r,n})^2\beta_r^2$, and the IMSE minimizing value of $\gamma_{r,n}$ is*

$$\gamma_{r,n}^{opt} = \frac{\beta_r^2}{\beta_r^2 + Var(\widetilde{\beta}_{r,n})}.$$

**Proof:** First,

$$\begin{aligned}
\text{ISE} &= \int \Big(\sum_r (\gamma_{r,n}\widetilde{\beta}_{r,n} - \beta_r)\psi_r(x)\Big)^2 dx \\
&= \sum_r \sum_s (\gamma_{r,n}\widetilde{\beta}_{r,n} - \beta_r)(\gamma_{s,n}\widetilde{\beta}_{r,n} - \beta_s) \int \psi_r(x)\psi_s(x)dx \\
&= \sum_r (\gamma_{r,n}\widetilde{\beta}_{r,n} - \beta_r)^2.
\end{aligned}$$

Next, for nonrandom $\gamma_{r,n}$ we have $E((\gamma_{r,n}\widetilde{\beta}_{r,n} - \beta_r))^2 = \gamma_{r,n}^2 \text{Var}(\widetilde{\beta}_{r,n}) + (1 - \gamma_{r,n})^2\beta_r^2$ which applied term by term to ISE provides the result for IMSE $= E(\text{ISE})$. Because the IMSE is a sum of quadratic functions of $\gamma_{r,n}$, the optimal $\gamma_{r,n}$ are easily seen to be as above. □

The optimal $\gamma$ shrinkage factors depend on $\text{Var}(\widetilde{\beta}_{r,n})$. If an oracle were to provide the values of $\gamma_{r,n}^{\text{opt}}$ then using them would result in an IMSE of

$$\text{IMSE}^{\text{opt}} = \sum_r \frac{\text{Var}(\widetilde{\beta}_{r,n})\beta_r^2}{\text{Var}(\widetilde{\beta}_{r,n}) + \beta_r^2}.$$

Our strategy is to estimate the optimal non-random $\gamma_{r,n}$ using the sampled values of $f(x_i)$. We use $\gamma_{r,n} = \widetilde{\beta}_{r,n}^2/[\widehat{\text{Var}}(\widetilde{\beta}_{r,n}) + \widetilde{\beta}_{r,n}^2]$ where $\widehat{\text{Var}}(\widetilde{\beta}_{r,n})$ is a variance estimate computed from $x_1, \ldots, x_n$ only.

If $\gamma_{r,n}$ is random, then IMSE does not have the form from Proposition 2. We

can still estimate ISE for the approximation $\widetilde{f}_n$ because

$$E\Big([\widetilde{f}_n(x_\ell) - f(x_\ell)]^2 \mid x_1, \ldots, x_n\Big) = \mathrm{ISE}, \tag{5}$$

for $\ell > n$. We can average squared errors between $f$ and $\widetilde{f}_n$ over many values of $x_\ell$ not used in the construction of $\widetilde{f}_n$, to estimate the accuracy of $\widetilde{f}_n$. Assuming that ISE changes only slowly with $n$, it is more efficient to average the squared errors $[f(x_i) - \widetilde{f}_{i-1}(x_i)]^2$ over a block of indices $m(n) \le i \le n$.

The accuracy of approximation is improved if the values of $\mathrm{Var}(\widetilde{\beta}_{r,n})$ are reduced. These variances in turn depend on the $\lambda$ shrinkage coefficients. We plan to report some results on $\mathrm{Var}(\widetilde{\beta}_{r,n})$ elsewhere. Here we mention that $T_{r,n}$ is uncorrelated with $\widetilde{\beta}_{r,n-1}$, so that $\mathrm{Var}(n\widetilde{\beta}_{r,n}) = \mathrm{Var}((n-1)\widetilde{\beta}_{r,n-1}) + \mathrm{Var}(T_{r,n})$, and hence $\mathrm{Var}(\widetilde{\beta}_{r,n}) = (1/n^2)\sum_{i=1}^{n}\mathrm{Var}(T_{r,n})$. The covariance of $\widetilde{\beta}_{r,n}$ and $\widetilde{\beta}_{s,n}$ depends on quantities like $\mu_{rstu} = \int \psi_r(x)\psi_s(x)\psi_t(x)\psi_u(x)dx$ and this is why we assumed that $\int \psi_r(x)^4 dx < \infty$ in Section 3.

## 5 Example

Here we consider a black box function of the type used in machine learning. This one is a neural network. These and some other black box methods are described in recent books such as [18], [19], and [20].

Venables and Ripley [21, page 300] describe a problem of predicting the performance of a computer. The data originate in Ein-Dor and Feldmesser [22]. For 209 computers, the variables described in Table 1 were obtained. We measured speed by $\log_{10}(\mathsf{perf})$ and used all the other variables except name and estperf as predictors. The first three predictor variables were log transformed. Because $\mathsf{chmin} \le \mathsf{chmax}$, the latter was replaced by $\mathsf{chmax} - \mathsf{chmin}$. Then all 6 predictors were linearly transformed to $(Z_1, \ldots, Z_6) \in [0,1]^6$.

Following [21], we fit a neural network model of the form

$$b_o + \sum_{i=1}^{6} w_{i\to o} Z_i + \sum_{h=1}^{H} w_{h\to o}\, \ell\Big(b_h + \sum_{i=1}^{6} w_{i\to h} Z_i\Big), \tag{6}$$

where $\ell(z) = (1+\exp(-z))^{-1}$ is a sigmoidal function and the $b$ and $w$ factors are real valued bias and weight parameters of the neural network, respectively. The arrows subscripting $w$ are evocative of information flowing from 6 input nodes or units to $H$ hidden units and then to an output unit. There are also direct connections from input to output. The bias parameters $b$ are there to incorporate intercept terms through a constant predictor equal to 1.

Equation (6) describes a network with a single hidden layer of $H$ units. We used $H = 3$ hidden units. The 31 parameters given in Table 2 were obtained using the nnet function in S-Plus as described in [21].

A black box model such as Equation (6) can be hard to interpret. We would like to know something about the relative importance of the various inputs to the function it encodes, and also whether those inputs affect the response in a nearly additive way, or through some strong interactions. In other settings we might want to study the effects of the biases and weights on one or more predictions or on a loss function for the network, as a function on $[0,1]^{31}$.

We fit a quasi-regression and quasi-regression with shrinkage to the function in (6) over $[0,1]^6$. The basis functions we used were tensor products based on Legendre polynomials as described in Section 3. Of these we used

$$\mathsf{R} = \mathsf{R}_{8,3,4} = \{\psi_r \mid \|r\|_1 \le 8, \|r\|_0 \le 3, \|r\|_\infty \le 4\},$$

corresponding to $p = 1145$ basis functions.

For both approximations we used $500,000$ function evaluations. The shrinkage factors were $\gamma_{r,n} = \lambda_{r,n} = 0$ for $1 \le n \le 600$, and otherwise

$$\gamma_{r,n} = \lambda_{r,n} = \frac{\widetilde{\beta}_{r,n}^2}{\widetilde{\beta}_{r,n}^2 + \widehat{\mathrm{Var}}(\widetilde{\beta}_{r,n})}$$

with

$$\widehat{\mathrm{Var}}(\widetilde{\beta}_{r,n}) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \bigl(T_{r,i} - \widetilde{\beta}_{r,n}\bigr)^2.$$

A java implementation (version 1.3.1_01 on linux) of quasi-regression took 177 seconds on an 800 megahertz processor. Incorporating shrinkage coefficients increased the elapsed time by about 80% to 318 seconds.

| | |
|---|---|
| **name** | Manufacturer and model |
| **syct** | cycle time in nanoseconds |
| **mmin** | minimum main memory in kilobytes |
| **mmax** | maximum main memory in kilobytes |
| **cach** | cache size in kilobytes |
| **chmin** | minimum number of channels |
| **chmax** | maximum number of channels |
| **perf** | published performance relative to an IBM 370/158-3 |
| **estperf** | estimated performance (from [22]) |

Table 1
Variables given in the cpu data set.

| | $b$ | $h1$ | $h2$ | $h3$ | $i1$ | $i2$ | $i3$ | $i4$ | $i5$ | $i6$ |
|---|---|---|---|---|---|---|---|---|---|---|
| $o$ | 0.46 | −2.82 | 3.17 | 0.39 | −1.21 | 1.36 | 1.42 | −1.01 | −0.33 | 0.30 |

| | $b$ | $i1$ | $i2$ | $i3$ | $i4$ | $i5$ | $i6$ |
|---|---|---|---|---|---|---|---|
| $h1$ | −1.12 | 0.45 | 2.24 | 2.51 | −1.63 | −0.56 | 0.43 |
| $h2$ | −1.09 | 2.28 | −0.10 | 1.44 | 2.70 | 1.24 | 0.25 |
| $h3$ | 0.04 | −0.11 | 0.11 | 0.12 | −0.10 | −0.04 | 0.02 |

Table 2
The upper table shows the coefficients for the output unit. From left to right these are the bias, the coefficients on the hidden units, and the coefficients on the original inputs. The lower table shows the biases and weights from input nodes to hidden nodes.

The error in approximating $f$ was estimated by

$$\widehat{\mathrm{ISE}}(n_m) = \frac{1}{m}\sum_{i=n_{m-1}}^{n_m}[f(x_i) - \widetilde{f}_{i-1}(x_i)]^2$$

for $n_m = m(m+1)/2$, at values of $m$ with $100 \le n_m \le 500,000$. The error estimate for $n = n_m$ is approximately the average of the $\sqrt{2n}$ previous squared prediction errors. Making the error window grow with $n$ produces accuracy estimates that are smoother functions of $n$ than the ones in [1] which use a fixed window width.

We normalize $\mathrm{ISE}(n_m)$ by the sample estimate $s^2$ of the variance $\sigma^2 = \int(f(x)-\beta_0)^2dx$. As a reference, $\mathrm{ISE}/\sigma^2 = 1$ means that $\widetilde{f}$ is as good an approximation as the optimal constant $\beta_0$ while $\mathrm{ISE}/\sigma^2 = 0.01$ means that $\widetilde{f}$ explains 99% of the variance in $f$. Figure 1 shows $\widehat{\mathrm{ISE}}(n_m)/s^2$ versus $n$ on logarithmic axes for $100 \le n \le 500,000$. The upper line is for QR and the lower line is for QRS. The shrinkage, which was applied for $n \ge 600$ clearly is advantageous. At the end of the run, the QR approximation explains all but 6.57% of the variance in $f$ while QRS explains all but 1.17%.

For quasi-regression without shrinkage $\mathrm{IMSE} = a + b/n$ for a constant $b$, where the asymptote is $a = \sum_{r\notin\mathsf{R}}\beta_r^2$. We would expect a similar pattern for ISE. It appears from Figure 1 that the asymptotic regime has not been reached by $n = 500,000$. Instead, the error in quasi-regression is still decreasing at essentially the $1/n$ rate, suggesting that estimation errors in $\widetilde{\beta}_{r,n}$ for $r \in \mathsf{R}$ dominate ISE. Theory for quasi-regression with shrinkage is more complicated. That method also has a lower bound of $\mathrm{ISE}/\sigma^2 \ge \sum_{r\notin\mathsf{R}}\beta_r^2/\sigma^2$ and in Figure 1 the estimate of ISE appears at least initially to approach the bound at a faster rate.

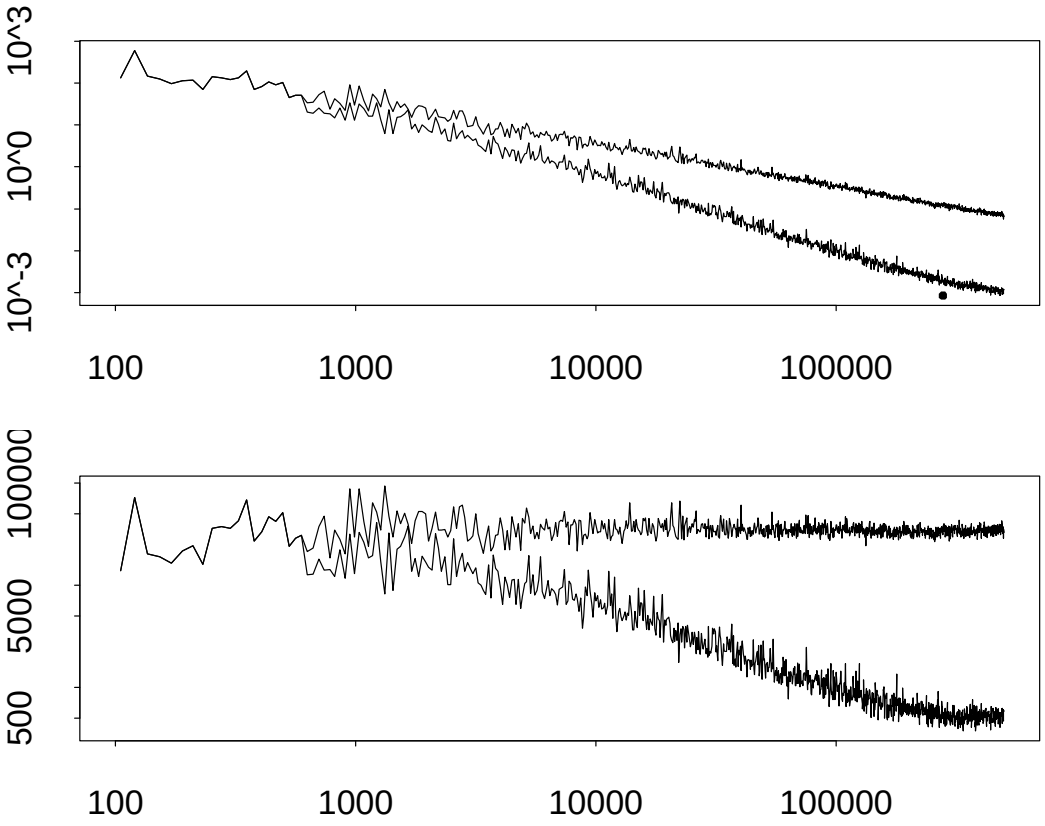


Fig. 1. The top plot shows the squared approximation error as a proportion of the variance of $f$. The upper line is for quasi-regression and the lower line is for quasi-regression with shrinkage (applied for $n \geq 600$). A reference point has been added near $500000 \times 177/318 \doteq 278302$ on horizontal axis. This is approximately the number of iterations that QRS could have done in the time taken by QR. The lower plot shows the squared approximation errors multiplied by $n$.

| syct | mmin | mmax | cach | chmin | chmax | total |
|---|---|---|---|---|---|---|
| 0.520 | 0.011 | 0.088 | 0.131 | 0.037 | 0.009 | 0.797 |

Table 3
Estimated proportion of variance of neural net model explained by variables individually. Approximately 79.7% of the variation in the function comes from an additive component. The individual effects above sum to 79.6 not 79.7, due to rounding.

Sums of squares of estimated coefficients can be formed and normalized by the sample variance. The additive proportion of the variance, corresponding to the sum of $\beta_r^2$ with $\|r\|_0 = 1$ is 0.797. Table 3 shows the estimated proportion of variance from each of the input variables. These estimates are based on quartic approximations because only $r$ with $\|r\|_\infty \leq 4$ were employed. In the presence of quintic and higher order components, the estimate presented here underestimates the importance of the additive approximation.

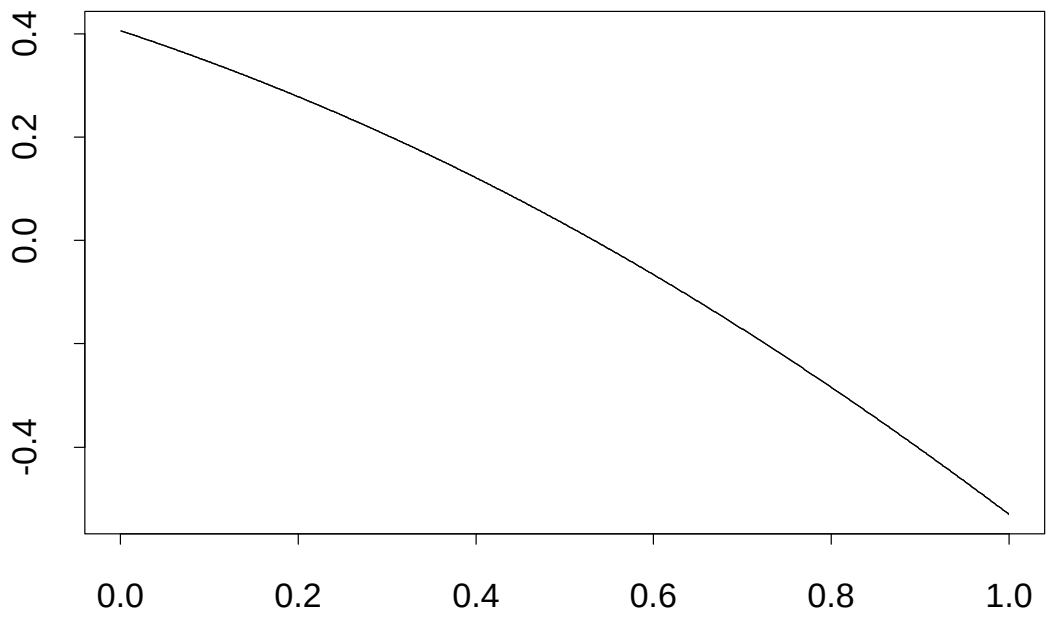


Fig. 2. Shown is an estimate of the main effect of syct for the neural network model of $\log_{10}(\mathsf{perf})$.

More than half of the variation in $f$ is explained by the effect of the first predictor variable syct acting alone. Figure 2 shows the estimate of the main effect of syct in the model. The curve plotted there is a quartic function taken from the QRS model. Linear and quadratic components dominate the curve.

The neural network model also has some non-negligible interactions. The interaction between syct and cach accounts for about 5.5% of the variation as does the interaction between mmax and cach. The interaction between mmin and mmax accounts for about 2.5%. The combined effect of all bivariate interactions is about 18.5% of the variance of $f$. Main effects and bivariate interactions together account for about 98.2% of the variance of $f$.

## 6 Discussion

Quasi-regression is a useful tool for approximating and interpreting black box functions. Shrinkage methods as described here can improve their accuracy. The $\gamma$-shrinkage terms become especially important for large $p$ because both IMSE and ISE contain sums of $p$ variance contributions.

Quasi-regression methods are well suited to functions from machine learning, especially those that are hard to interpret but very fast to evaluate. Some

caveats apply to the machine learning type of applications. The training data points are unlikely to have a sample distribution that resembles the uniform distribution on $(0,1)^d$. Then an interesting feature in $f$ captured by a corresponding feature in $\widetilde{f}$ might possibly exist over a region of $(0,1)^d$ where there were no data points. But when a variable or an interaction is virtually absent from $f$, it is hard for the data distribution to make that variable or interaction important.

A second caveat is that the variables that are causally important are not necessarily the ones that end up playing an important role in $f$ or $\widetilde{f}$. Even for linear models correlation is not causation, and similar issues are present in sophisticated black boxes.

Quasi-regression as presented here uses only ordinary Monte Carlo sampling. Monte Carlo sampling allows for simple error estimation without the need for choosing special samples sizes $n$ or sampling the function in blocks. That quasi-Monte Carlo sampling may offer an improvement in accuracy was seen in [3].

## References


[1] J. An, A. B. Owen, Quasi-regression, Journal of Complexity 17 (4) (2001) 588–607.

[2] A. B. Owen, Assessing linearity in high dimensions, Annals of Statistics 28 (1) (2000) 1–19.

[3] C. LeMieux, A. B. Owen, Quasi-regression and the relative importance of the ANOVA components of a function, in: K. T. Fang, F. J. Hickernell, H. Niederreiter (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2000, Springer-Verlag, Berlin, 2002, pp. 331–344.

[4] C. K. Chui, H. Diamond, A natural formulation of quasi-interpolation by multivariate splines, Proceedings of the American Mathematical Society 99 (4) (1987) 643–646.

[5] D. L. Donoho, I. M. Johnstone, Ideal spatial adaptation by wavelet shrinkage, Biometrika 81 (1994) 425–455.

[6] S. Efromovich, On orthogonal series estimators for random design nonparametric regression, in: Computing Science and Statistics. Proceedings of the 24rd Symposium on the Interface, 1992, pp. 375– 379.

[7] A. B. Owen, A central limit theorem for Latin hypercube sampling, Journal of the Royal Statistical Society, Series B 54 (1992) 541–551.

[8] J. Koehler, A. Owen, Computer experiments, in: S. Ghosh, C. Rao (Eds.), Handbook of Statistics, 13: Design and Analysis of Experiments, North-Holland, 1996, pp. 261–308.

[9] J. Sacks, W. J. Welch, T. J. Mitchell, H. P. Wynn, Design and analysis of computer experiments (c/r: P423-435), Statistical Science 4 (1989) 409–423.

[10] C. Currin, T. Mitchell, M. Morris, D. Ylvisaker, Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments, Journal of the American Statistical Association 86 (1991) 953–963.

[11] K. Ritter, Average case analysis of numerical problems, Ph.D. thesis, University of Erlangen (1995).

[12] A. Saltelli, K. Chan, E. M. Scott, Sensitivity Analysis, Wiley, Chichester, 2000.

[13] I. M. Sobol', Sensitivity estimates for non-linear mathematical models, Matematicheskoe Modelirovanie 2 (1990) 112–118, (In Russian).

[14] I. M. Sobol', Sensitivity estimates for nonlinear mathematical models, Mathematical Modeling and Computational Experiment 1 (1993) 407–414.

[15] G. E. B. Archer, A. Saltelli, I. M. Sobol', Sensitivity measures, ANOVA-like techniques and the use of bootstrap, Journal of Statistical Computing and Simulation 58 (1997) 99–120.

[16] I. M. Sobol', Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates, Mathematics and Computers in Simulation 55 (2001) 271–280.

[17] A. B. Owen, The dimension distribution and quadrature test functions, Tech. rep., Stanford University, Department of Statistics (2001).

[18] V. N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, Berlin, 1995.

[19] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification (2nd ed), Wiley, New York, 2000.

[20] T. J. Hastie, R. J. Tibshirani, J. H. Friedman, The Elements of Statistical Learning, Springer-Verlag, New York, 2001.

[21] W. Venables, B. Ripley, Modern Applied Statistics with S-Plus, 3rd Edition, Springer, New York, 1999.

[22] P. Ein-Dor, J. Feldmesser, Attributes of the performance of central processing units: a relative performance prediction model, Communications of the ACM 30 (1987) 308–317.