Quasi-regression for visualization and interpretation of black box functions

Tao Jiang
Department of Statistics
Stanford University
Stanford, CA 94305

Art B. Owen
Department of Statistics
Stanford University
Stanford, CA 94305

July 2002

Abstract

Many machine learning algorithms make use of black box functions. Given such a function $f(x)$ for d dimensional x, it can be difficult to tell which variables, if any, dominate f. In quasi-regression the function f is expanded in an orthogonal basis. A large number of the coefficients are estimated by Monte Carlo and then various parts of f are plotted. We illustrate quasi-regression on some neural network and support vector machine functions. Non-independence of the training data complicates interpretation, but the method does give a way of visualizing the contributions to f of different subsets of input variables.

1 Introduction

Let $f(x)$ be a function defined on the unit cube $(0,1)^d$. From evaluations of f alone we seek answers to certain questions about f, such as which of the d inputs to f are most important, and which of the interactions among these variables are most important. We also would like to construct plots showing the average effect on f of a small subset of the d inputs.

Our approach to these problem is to expand f in an orthonormal basis with an infinite number of coefficients. If we knew these coefficients we could use them, in particular certain sums of squared coefficients, to describe the extent to which f depends on various of its inputs, as well as the extent to which f has low or high order interactions or has a good low degree approximation. We use Monte Carlo methods to estimate these coefficients, and apply bias corrections to sample sums of squared coefficients.

Section 2 gives the notation for orthonormal basis expansions. Section 3 describes some interpretable subsets of coefficients, and introduces the ANOVA decomposition of $(0,1)^d$. Section 4 describes Monte Carlo methods of approximating these coefficients. Section 5 outlines how this method can be used on the sort of black box function arising in machine learning applications. By far the hardest issue to deal with is

that the predictor variables usually have nothing like an independent distribution in the training data. Section 6 presents an application to prediction functions from a support vector machine and from a neural network.

2 Orthonormal basis expansion

Let $\phi_0(x) \equiv 1$ for all real values $0 < x < 1$ and suppose that $\phi_0, \phi_1, \ldots$ is a complete orthonormal basis for $L^2(0,1)$. That is $\int_0^1 \phi_r(x)\phi_s(x)dx = 1_{r=s}$, and $f(x) = \sum_{r=0}^{\infty} \beta_r \phi_r(x)$ in mean square, where $\beta_r = \int_0^1 f(x)\phi_r(x)dx$. Familiar examples include orthogonal polynomials, Haar wavelets, and Fourier series.

We extend the analysis to $(0,1)^d$ by taking tensor products. For $r = (r_1, \ldots, r_d) \in \{0, 1, \ldots\}^d$ and $x = (x^{(1)}, \ldots, x^{(d)})$ define $\psi_r(x) = \prod_{j=1}^d \phi_{r_j}(x^{(j)})$. Then for $f \in L^2(0,1)^d$ we have

$$f(x) = \sum_r \beta_r \psi_r(x) \tag{1}$$

in mean square, where the coefficients are now

$$\beta_r = \int_{(0,1)^d} f(x)\psi_r(x)dx. \tag{2}$$

3 Interpretable sets of coefficients

The variance of $f(x)$ for $x \sim U(0,1)^d$ is $\sigma^2(f) = \sum_{r \neq 0} \beta_r^2$. One measure of the importance of a subset $\mathcal{R}$ of coefficients is $\sigma^2_{\mathcal{R}}(f) = \sum_{r \in \mathcal{R}} \beta_r^2$. A normalized version is $\sigma^2_{\mathcal{R}}/\sigma^2$.

The subsets of interest vary with applications. We can group together basis functions ψ_r involving a subset of the input variables. Or we can group together the ones that are of "low order" in r. There are several ways to describe low order of r, including

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

The first of these describes the number of inputs with a nontrivial role in ψ_r. The second reduces to the usual notion of degree for polynomials. When ϕ_{r_j} are orthogonal polynomials, then the closest approximation to f of degree k is $\sum_{\|r\|_1 \le k} \beta_r \psi_r(x)$.

When considering subsets of input variables, we have in mind an analysis of variance (ANOVA) decomposition of $(0,1)^d$ that generalizes the discrete ANOVA used in factorial experiments. The functional ANOVA was introduced by Hoeffding (1948) and has been studied by Efron and Stein (1981) and Sobol' (1969). Let $u \subset \{1, 2, \ldots, d\}$. Then we may write

$$f(x) = \sum_{u \subseteq \{1,2,\ldots,d\}} f_u(x) \tag{3}$$

where $f_u(x)$ only depends on those components of x in the subset u. The decomposition (3) is not unique. The usual choice has the line integral $\int_0^1 f_u(x)dx^{(j)}$ vanishing when $j \in u$, for any values of $x^{(k)}$ $k \neq j$.

If $u \neq v$ then $\int_{(0,1)^d} f_u(x)f_v(x)dx = 0$ and the variance of f can be written $\sigma^2 = \sum_{u\neq\emptyset} \sigma_u^2$ where σ_u^2 is the variance of $f_u(x)$ for $x \sim U(0,1)^d$. For the empty set, $f_\emptyset$ is everywhere equal to the integral of f, and $\sigma_\emptyset^2 = 0$.

The terms f_u where u has cardinality $|u| = 1$ are called main effects. Terms with $|u| > 1$ are called interactions. The ANOVA components can be written in terms of the orthogonal decomposition. For $u \neq \emptyset$, let $\mathcal{R}_u = \{r \mid r_j > 0 \text{ iff } j \in u\}$. Then

$$f_u(x) = \sum_{r\in\mathcal{R}_u} \beta_r\psi_r(x), \quad \text{and,} \quad \sigma_u^2 = \sum_{r\in\mathcal{R}_u} \beta_r^2.$$

The additive function closest to f in mean square is (Stein 1987)

$$f_{\text{add}}(x) = f_\emptyset + \sum_{j=1}^d f_{\{j\}}(x) = \sum_{\|r\|_0\le 1} \beta_r\psi_r(x).$$

The closest function to f having interactions of order at most k is

$$\sum_{\|r\|_0\le k} \beta_r\psi_r(x).$$

The importance of the interaction f_u may be measured by σ_u^2. The importance of the subset u of input variables can be described by either

$$\underline{\tau}_u^2 = \sum_{v\subseteq u} \sigma_v^2, \quad \text{or,} \quad \overline{\tau}_u^2 = \sum_{v\cap u\neq\emptyset} \sigma_v^2.$$

Normalized versions, $\underline{\tau}_u^2/\sigma^2$ and $\overline{\tau}_u^2/\sigma^2$, are the global sensitivity indices of Sobol' (1993). The former describes the variance of f attributable solely to the effects of variables $j \in u$ while the latter describes the variance attributable at least in part to variables $j \in u$. Clearly $\underline{\tau}_u^2 \le \overline{\tau}_u^2$. A large value of $\underline{\tau}_u^2$ indicates a subset of variables that acting together can strongly affect f. A small value of $\overline{\tau}_u^2$ indicates a subset of variables that have little effect on f, even in combination with other variables.

4 Monte Carlo coefficient estimation

The coefficient β_r is equal to the expected value of $f(x)\psi_r(x)$ when $x \sim U(0,1)^d$. An and Owen (2001) present Monte Carlo sampling to estimate β_r. Let $x_i \sim U(0,1)^d$ for $i = 1,\dots,n$, and define

$$\widetilde{\beta}_{r,n} = \frac{1}{n}\sum_{i=1}^n f(x_i)\psi_r(x_i), \quad \text{and,} \tag{4}$$

$$S_{r,n} = \frac{1}{n-1}\sum_{i=1}^n \Big(f(x_i)\psi_r(x_i) - \widetilde{\beta}_{r,n}\Big)^2. \tag{5}$$

Both $\widetilde{\beta}_r^{(n)}$ and $S_{r,n}$ can be computed in one pass over the Monte Carlo sample by numerically stable updating formulas given by Chan, Golub, and LeVeque (1983). When it is unambiguous, the dependence on n is suppressed from the notation. The expected value of S_r/n is $\mathrm{Var}(\widetilde{\beta}_r)$.

Quasi-regression estimators are not least squares estimates. Least squares estimation of p coefficients takes $O(np^2)$ time where quasi-regression requires only $O(np)$. The savings is possible because the functions $\psi_r(x)$ are by construction orthogonal with respect to $x \sim U(0,1)^d$, even though they are not orthogonal with respect to the sample distribution of $x_1, \ldots, x_n$. For fixed p and $n \to \infty$, Owen (2000) shows that least squares is ordinarily more accurate than quasi-regression. But quasi-regression allows much larger p to be used for a given computational budget.

In practice we estimate only finitely many coefficients. We truncate the infinite set of indices to a finite set $\mathcal{U} = \{r \mid \|r\|_0 \le B_0, \|r\|_1 \le B_1, \|r\|_\infty \le B_\infty\}$ for problem dependent values B_0, B_1, and B_∞. For estimates $\widetilde{\beta}_{r,n}$ of p coefficients β_r, $r \in \mathcal{U}$, the quasi-regression approximation to f is $\widetilde{f}_n(x) = \sum_{r\in\mathcal{U}} \widetilde{\beta}_{r,n}\psi_r(x)$.

The accuracy of $\widetilde{f}_n$ can be assessed by averaging $(f(x_i) - \widetilde{f}_n(x_i))^2$ over a number of values $i > n$. This provides a form of cross-validated error because $\widetilde{f}_n$ depends only on $x_1, \ldots, x_n$ and is independent of x_{n+1}. We have found it convenient to update a running average of the most recent values of $(f(x_n) - \widetilde{f}_{n-1}(x_n))^2$. The number of values we use is approximately $\sqrt{2n}$. This mean squared error combines truncation errors due to using a finite set $\mathcal{U}$ and estimation errors $(\widetilde{\beta}_{r,n} - \beta_r)^2$ for $r \in \mathcal{U}$.

For a large set $\mathcal{R}$ of coefficients, an unbiased estimate of $\sum_{r\in\mathcal{R}} \beta_r^2$ is $\sum_{r\in\mathcal{R}} \widetilde{\beta}_r^2 - S_r/n$. The bias correction can be very important when the cardinality of $\mathcal{R}$ is large.

Jiang and Owen (2001) introduce shrinkage methods to reduce the variance of quasi-regression estimates, taking

$$\widetilde{f}_n(x) = \widetilde{f}_{n,\gamma}(x) = \sum_{r\in\mathcal{U}} \gamma_{r,n}\widetilde{\beta}_{r,n}\psi_r(x),$$

where $\gamma_{r,n} \in [0,1]$ is a shrinkage parameter that may depend on $x_1, \ldots, x_n$. Shrinkage is also employed in constructing the coefficient estimates, taking

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

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 $n \ge 0$, only finitely many of the $\lambda_{s,n}$ are nonzero, and $\lambda_{s,0} = \widetilde{\beta}_{s,0} = 0$.

Updatable unbiased variance estimates are available for use with shrinkage as are cross-validated error estimates. The cost remains $O(np)$. For details, see Jiang and Owen (2001).

5 Application to machine learning

Suppose that training data of the form (x_i, y_i) are used to construct a rule $f(x)$ on which to base prediction of y from x. (The x_i here are not meant to be the same values

as those used in quasi-regression.) In regression problems where $y \in \mathbb{R}$ the function f may be a direct estimate of y. When y takes only two values then $f(x)$ may be compared to a threshold to determine which value is the prediction. In such cases f may have an interpretation related to a conditional probability of y given x, or to a margin.

Many of the most effective methods for machine learning are built up as combinations of simple functions. Examples include radial basis functions, feedforward neural networks, support vector machines (Vapnik 1995) and combinations of trees, as in Freund and Schapire (1996), Friedman, Hastie, and Tibshirani (2000), and Breiman (2001). Though each individual part of the function is simple looking, the function as a whole is complex.

We would like to know which components of x are most important to f. There are at least three different notions of importance. The first is causality. The component $x^{(j)}$ is causally important if changing it in the real world will cause a change in y. Causality cannot ordinarily be inferred from observational data, due to the possibility of unknown variables affecting both $x^{(j)}$ and y. (An exception can be made for some designed experiments employing randomization.) We do not attempt to infer which variables are causal.

A second notion of variable importance is that a variable $x^{(j)}$ is important to the extent that out-of-sample predictions of y are better for rules trained with $x^{(j)}$ than for rules trained on data with $x^{(j)}$ absent. Predictive variable importance of this sort is most directly addressed by training the machine learning rule multiple times employing different subsets of the input variables, and comparing accuracy on a test set. See, for example, John, Kohavi, and Pfleger (1994).

A third notion of variable importance is that taking the function $f(x)$ at face value, the variable $x^{(j)}$ plays a large role in f. In this setting we regard the function f as the object of study and look at which of the input variables are important in determining it.

To fix ideas, consider a linear predictor $\alpha_0 + \sum_{j=1}^{d} \alpha_j x^{(j)}$. We cannot ordinarily tell from the α_j whether $x^{(j)}$ is causal, and because of possibility of collinearity, the α_j do not even tell us which of the $x^{(j)}$ are necessary for a good prediction. But in a linear model, a large value $|\alpha_j|$ is an indication of the importance of $x^{(j)}$ to the function f, assuming that the $x^{(j)}$ have been rescaled to be comparable by some measure of spread. It is also immediate in a linear model that the variables act singly without any interaction. Our goal is to develop comparable face value interpretations for black box functions.

Our approach is similar to that of Roosen (1995) who sampled on a randomized orthogonal array design (Owen 1992) and averaged function values to estimate ANOVA components. We find that quasi-regression estimates are much smoother than averages over orthogonal arrays. Researchers in global sensitivity analysis (Saltelli, Chan, and Scott 2000) also study black box functions at face value, regarding the inputs as independent.

A serious issue in face value interpretation for the machine learning context is that the components $x^{(j)}$ in the training data are typically not close to independent. We may identify important structure in f, only to find that it takes place in a region of $(0,1)^d$ containing none of the training sample points. As a simple example, suppose that $d = 2$

and that all of the points $(x_i^{(1)}, x_i^{(2)})$ for $i = 1, \ldots, n$ are in the triangle below the line $x^{(1)} + x^{(2)} = 1$. The quasi-regression approach considers points $x \in (0,1)^2$ above the line as well as below the line where the data are.

A similar issue arises in the noising up method of Breiman (2001). There, the importance of $x^{(j)}$ is assessed by how badly performance degrades when the values $x_i^{(j)}$ for different cases are randomized. After noising-up, the pairs $(x_i^{(1)}, x_i^{(2)})$ will include some points above the line. Similarly, the partial dependence plots of Friedman (2001) present averages of f over predictor combinations combining sample values of one subset of predictors $x^{(j)}$ with all values between 0 and 1 for the complementary subset. In this simple example partial dependence plots will also include in the average some part of the region above the line.

Face value interpretations of variable importance can still be used to garner insight into f. Ideally a black box function should extrapolate conservatively into any regions where there are no training data. Contours of the quasi-regression approximation can be plotted along with the training data to check this.

6 Example

For our example, we consider a widely studied data set from the UC Irvine repository (Blake and Merz 1998). The response variable is a determination of whether a given woman is diabetic. There are 7 predictors, including medical measurements and personal history. All of the women are Pima Indians. We used the version of this data set found in Ripley (1996). There are 200 complete cases for training and 332 for a test set. The number of pregnancies was replaced by $\log(1 + \text{number of pregnancies})$. Then it and the other predictors were scaled linearly to the interval $[0, 1]$.

We considered support vector machines and feedforward neural networks for making the predictions. The neural network was run using an Splus version described in Venables and Ripley (1999). All squashing functions were logistic, $(1 + e^{-z})^{-1}$. The probability of being diabetic was modelled as a linear logistic regression in a combination of the original variables and two hidden units which were themselves constructed by logistic squashing of linear functions of the inputs. We took for f the output of the neural network just before the final squashing into $(0, 1)$. The comparable logistic regression is then simply a linear function f. For the support vector machine we took f to be the distance from the separating hyperplane.

The neural network was trained twice, yielding quite different functions, with similar performance. We also used support vector machines with Gaussian kernels. Table 1 shows the results of quasi-regression approximations to these black boxes. The quasi-regression approximations were built using shrinkage and with n = 500,000 evaluations of the black box functions. Coefficients in $\mathcal{U} = \{r \mid \|r\|_0 \le 4, \|r\|_1 \le 8, \|r\|_\infty \le 4\}$ were used. There are $p = 4215$ of these coefficients. The basis functions used are tensor products of orthogonal polynomials over the unit interval.

The two neural networks have the same parametric form. As is well known, network training can lead to different models, even for the same architecture on the same data. In this case the first fit almost ignores the number of pregnancies. The second neural network has the number of pregnancies explaining 6.9% of the variation in f all

	Logistic	NNet1	NNet2	SVM(G)
Training Error	0.230	0.200	0.170	0.200
Test Error	0.244	0.214	0.232	0.205
Number of Pregnancies	0.009	0.002	0.069	0.136
Plasma Glucose	0.436	0.294	0.238	0.259
Diastolic B.P.	0.002	0.000	0.030	0.000
Skin fold thickness	0.002	0.051	0.006	0.017
Body mass index	0.134	0.010	0.057	0.105
Diabetes Pedigree Fn	0.320	0.429	0.343	0.257
Age	0.098	0.060	0.036	0.096
Total additive	1.000	0.846	0.779	0.869
Total two-factor	0.000	0.085	0.125	0.118
Total three-factor	0.000	0.050	0.067	0.012
Total four-factor	0.000	0.012	0.018	0.000
QREG $1-R^2$		6.6e$-$3	1.0e$-$2	4.1e$-$4

Table 1: The top rows of numbers show training and test error rates for logistic regression, two neural network fits, and support vector machines using a Gaussian kernel. The next seven rows show the proportion of the variation in each model attributable to the predictor variables taken one at a time. The following four rows show the total contributions of main effects and interactions involving two, three, and four of the variables. The proportions are unbiased estimates based on a Monte Carlo sample, divided by the sample variance and then rounded to 3 places. The functions on which they are based fits the data with a cross-validated $1-R^2$ given in the bottom row.

by itself. This difference is larger than the quasi-regression error. The quasi-regression model fit the original black boxes to within 0.66% and 1%, based on a cross-validated squared error.

The support vector machine function is 86.9% additive and these main effects together with two factor interactions explain about 97.7% of the variation. The two largest main effects, those for plasma glucose and the diabetes pedigree function, were of nearly equal size. Both had positive slopes. The main effect for plasma glucose was virtually linear while there was some slight negative curvature in the effect for the pedigree function. The largest interaction is that between $\log(1+\text{Number of pregnancies})$ and the diabetes pedigree function, and it accounts for 2.7% of the SVM variation. Figure 1 shows the interaction and compares it to the same interaction for neural network 1 in which it explained only 5×10^{-5} of the variation.

It is clear that one prominent corner in the SVM interaction is responding to a single point with zero pregnancies and the highest observed pedigree function. There is also some nonlinear structure in the corner where both of these variables are high, despite the absense of any data points there. In this instance it appears to be a small amount of nonlinearity.

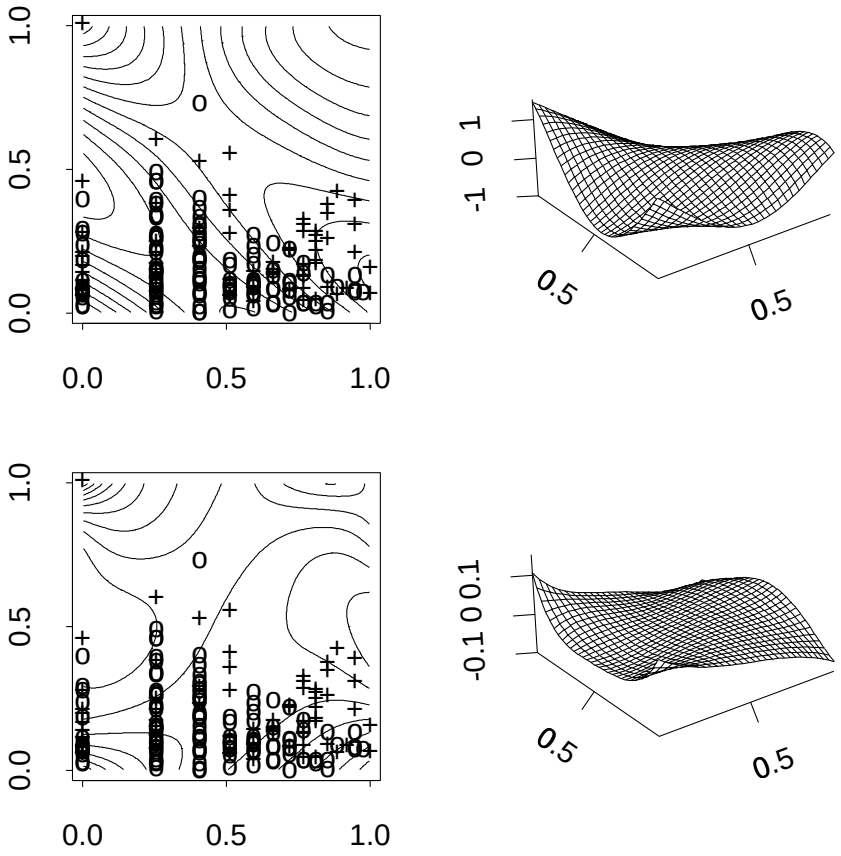

Figure 1: The top two plots show the interaction between $\log(1 +$ Number of pregnancies$)$ and the diabetes pedigree function in the support vector machine prediction of diabetes. The plot on the left is a contour plot, with the training data superimposed. Pluses are plotted for diabetics, and open circles for non-diabetics. The corresponding perspective plot appears on the right. The bottom two plots show the same interaction for the neural network.

Acknowledgments

We thank Ji Zhu for making available his Splus functions for training support vector machines. This work was supported by the U.S. National Science Foundation under contract DMS-0072445.

References

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

Blake, C. and C. Merz (1998). UCI repository of machine learning databases.

Breiman, L. (2001). Random forests. Technical report, University of California, Berkeley, Department of Statistics.

Chan, T. F., G. H. Golub, and R. J. LeVeque (1983). Algorithms for computing the sample variance: Analysis and recommendations. *The American Statistician 37*, 242–247.

Efron, B. and C. Stein (1981). The jackknife estimate of variance. *Annals of Statistics 9*, 586–596.

Freund, Y. and R. Schapire (1996). Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, San Francisco, pp. 148–156. Mrogan Kauffman.

Friedman, J. (2001). Greedy function approximation: a gradient boosting machine. Technical report, Stanford University, Statistics Department.

Friedman, J., T. Hastie, and R. Tibshirani (2000). Additive logistic regression: a statistical view of boosting. *The Annals of Statistics 38*(2), 337–374.

Hoeffding, W. (1948). A class of statistics with asymptotically normal distribution. *Annals of Mathematical Statistics 19*, 293–325.

Jiang, T. and A. B. Owen (2001). Quasi-regression with shrinkage. Technical report, Stanford University, Statistics Department.

John, G. H., R. Kohavi, and K. Pfleger (1994). Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the Eleventh International Conference*, San Francisco, pp. 121–129. Morgan Kaufmann.

Owen, A. B. (1992). Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica 2*, 439–452.

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

Ripley, B. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press.

Roosen, C. B. (1995). *Visualization and exploration of high-dimensional functions using the functional ANOVA decomposition*. Ph. D. thesis, Stanford University, Department of Statistics.

Saltelli, A., K. Chan, and E. M. Scott (2000). *Sensitivity Analysis*. Chichester: Wiley.

Sobol', I. M. (1969). *Multidimensional Quadrature Formulas and Haar Functions*. Moscow: Nauka. (In Russian).

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

Stein, M. (1987). Large sample properties of simulations using Latin hypercube sampling. *Technometrics 29*(2), 143–51.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Berlin: Springer-Verlag.

Venables, W. and B. Ripley (1999). *Modern Applied Statistics with S-Plus, 3rd Edition*. New York: Springer.