Statistics and Computing **14**: 333–336, 2004

Polynomial approximations of multivariate smooth functions from quasi-random data

SYLVAIN MAIRE

ISITV, Université de Toulon et du Var, Avenue G, Pompidou BP 56 - 83262, La Valette du Var, Cedex, France
maire@univ-tln.fr

Received April 2003 and accepted May 2004

We improve a Monte Carlo algorithm which computes accurate approximations of smooth functions on multidimensional Tchebychef polynomials by using quasi-random sequences. We first show that the convergence of the previous algorithm is twice faster using these sequences. Then, we slightly modify this algorithm to make it work from a single set of random or quasi-random points. This especially leads to a Quasi-Monte Carlo method with an increased rate of convergence for numerical integration.

Keywords: iterative Monte Carlo algorithm, quasi-random sequences, polynomial approximations, numerical integration

1. Introduction

Monte Carlo methods are not usually very accurate unless one can find efficient variance reduction schemes (Kalos and Whitlock 1986, Lapeyre, Pardoux and Sentis 1998) to build methods with higher rates of convergence (Atanassov and Dimov 1999, Lepage 1978). Using control variates, we have developed in previous works (Maire 2003a, b) an iterative Monte Carlo algorithm to compute mean-square approximations of a function f on orthonormal bases e_k on the hypercube $D = [0, 1]^Q$. We write

$$f(x) = \sum_{k=1}^{p} a_k e_k(x) + r(x)$$

and we give initial Monte Carlo approximations of $a_k = \langle f, e_k \rangle$ and f by

$$a_k^{(1)} = \frac{1}{N} \sum_{i=1}^{N} f(X_i) e_k(X_i), \quad f^{(1)}(x) = \sum_{k=1}^{p} a_k^{(1)} e_k(x)$$

using N uniform sample values X_i. We compute then a correction b_k at step $M-1$ using N sample values Y_i independent from all the previous ones by

$$b_k^{(M-1)} = \frac{1}{N} \sum_{i=1}^{N} \big(f(Y_i) - f^{(M-1)}(Y_i)\big) e_k(Y_i)$$

to achieve new approximations at step M by

$$a_k^{(M)} = a_k^{(M-1)} + b_k^{(M-1)}, \quad f^{(M)}(x) = \sum_{k=1}^{p} a_k^{(M)} e_k(x).$$

The approximation of a regular function $f \in C^L([-1, 1]^Q)$ on orthogonal polynomial bases can be written

$$f(x) = \sum_{m \in \mathbb{N}^Q} a_m e_m(x) = \sum_{m \in \mathbb{N}^Q} a_m P_{m_1}(x_1) P_{m_2}(x_2) \dots P_{m_Q}(x_Q),$$

where $P_{m_j}(x_j)$ is the orthogonal polynomial of degree m_j with respect to x_j. The coefficients a_m of this approximation on either Legendre or Tchebychef polynomials (Bernardi and Maday 1992) verify

$$|a_m| \le \frac{C_1}{(\widehat{m_1}\widehat{m_2} \dots \widehat{m_Q})^L}$$

where the positive constant C_1 depends only on f and where $\hat{m} = \max(1, m)$. We only have to keep the coefficients a_m for which $\widehat{m_1}\widehat{m_2} \dots \widehat{m_Q}$ is small. Hence we define similarly as for Korobov spaces (Krommer and Ueberhuber 1998)

$$W_{Q,d} = \{m \in \mathbb{N}^Q / (\widehat{m_1} \dots \widehat{m_Q}) \le d\}, \quad Card(W_{Q,d}) = L_{Q,d}$$

and we write f as

$$f(t) = \sum_{m \in W_{Q,d}} a_m e_m(t) + r(t).$$

0960-3174

The following theorem gives the performances of the algorithm when the approximation of the function f is given by

$$f^{(M)}(t) = \sum_{m \in W_{Q,d}} a_m^{(M)} e_m(t).$$

Theorem 1.1. *Assuming the previous hypotheses and if*

(i)

$$\frac{L_{Q,d}}{N} < 1,$$

(ii)

$$\tau = \sup\left(\sup_k \int_D \left(1 - e_k^2(x)\right)^2 dx, \sup_{j,k,j \neq k} \int_D e_k^2(x) e_j^2(x)\, dx \right) \leq \frac{N}{4},$$

we have

$$E\left(a_m^{(M)}\right) = a_m,$$

$$Var\left(a_m^{(M)}\right) \leq 2\left(\mu_1 \frac{K(L_{Q,d})^{M-1}}{N^M} C(L_{Q,d})^M + \mu(L_{Q,d}) \frac{1}{d^{2L-1-\varepsilon}} \right)$$

and the mean-square approximation of f by $f^{(M)}$

$$E\left(\int_D \left(f(x) - f^{(M)}(x)\right)^2 dx \right) \leq 2L_{Q,d}\left(Var\left(a_m^{(M)}\right)\right) + \frac{\mu_2}{d^{2L-1-\varepsilon}}.$$

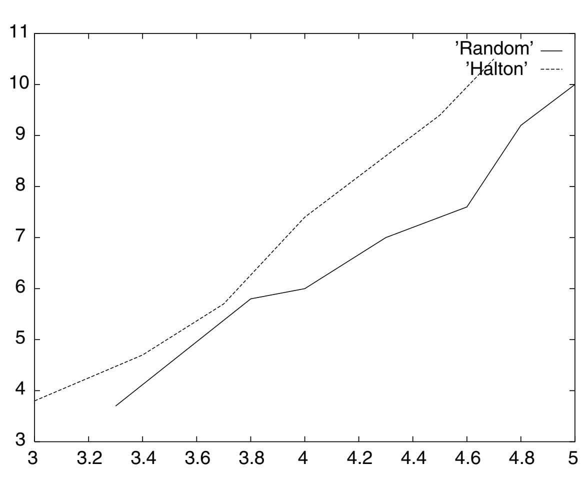
The proof relies on an accurate study of the statistical properties of the $a_m^{(M)}$ and on the peculiar decay of the a_m. Looking at $Var(a_m^{(M)})$, we can see that the first term goes to zero geometrically as the number of steps increases if $\frac{K(L_{Q,d})C(L_{Q,d})}{N} < 1$. We achieve an accuracy as $O(\frac{1}{d^{L-0.5-\varepsilon}})$ in the approximation of a_m by $a_m^{(M)}$. We have proved in Maire (2003a) that the Tchebychef bases mimimizes the constants $C(L_{Q,d})$ and $K(L_{Q,d})$. We slightly modify the original algorithm in computing the approximations of the coefficients $\langle f, \tilde{T}_n \rangle$ on the normalized Tchebychef polynomials $\tilde{T}_n(x)$ by the Monte Carlo approximation of $E(\pi f(V)\tilde{T}_n(V))$ where the density of V is $\frac{1}{\pi\sqrt{1-v^2}} 1_{[-1,1]}(v)$. The numerical accuracy in terms of numerical integration was comparable to a previous method with an optimal rate of convergence developed in Atanassov and Dimov (1999). We will also use the Tchebychef polynomials in the following.

2. Acceleration using quasi-random sequences

The term $\frac{K(L_{Q,d})C(L_{Q,d})}{N}$ which determins the geometrical decay of the variance of the $a_m^{(M)}$ until convergence comes more or less from the computation of the corrections $b_k^{(M-1)}$ using a Monte Carlo approximation of the integrals $\int_D (f(x) - f^{(M)}(x))e_k(x)\, dx$. In order to speed up the convergence of the algorithm, we intend now to replace the random drawings by low-discrepancy sequences (Krommer and Ueberhuber 1998, Niederreiter 1992). These sequences have a better distribution than usual number generators in the unit cube. The discrepancy measures the deviation of a given sequence of N points to the uniform distribution. Low-discrepancy sequences have approximatively the same number of points in each subdomain of same volume. The most commonly used low-discrepancy sequences are Halton (Krommer and Ueberhuber 1998, Niederreiter 1992), Sobol (1967) and (t, N) sequences introduced by Niederreiter (1987, 1992). The main property of these quasi-random sequences is that they can achieve a rate of convergence for numerical integration of $\frac{(\log(N))^{(Q-1)}}{N}$ instead of $\frac{1}{\sqrt{N}}$ for crude Monte Carlo (Krommer and Ueberhuber 1998, Niederreiter 1987). This means that we can expect that the term $\frac{K(L_{Q,d})C(L_{Q,d})}{N}$ will be replaced by $\frac{(\log(N))^{2(Q-1)}K(L_{Q,d})C(L_{Q,d})}{N^2}$ in Theorem (1.1). The algorithm is especially useful for moderate dimensions say less than 10. So, if we compare this two terms the number of steps until convergence should be divided by around two. This assesment will be confirmed in the numerical experiments if we replace the random drawings by Halton sequences. A good way to check the accuracy of all the coefficients $a_m^{(M)}$ is to compute an approximate value $\tilde{I}(f)$ of $I(f) = \int_D f(x)\, dx$ by

$$\tilde{I}(f) = \sum_{k \in W_{Q,d}} a_k^{(M)} \int_D e_k(x)\, dx.$$

We plot $-\log_{10} \frac{|I - \tilde{I}|}{|I|}$ as a function of $\log_{10}(N_t)$ where N_t is a number of sample values used in the approximation algorithm for $f(x, y, z) = \exp(\frac{x+y+z}{2})$ using both random numbers and Halton sequences



An approximate value of the order of the method is given by the slope of the curve. We obtain respectively around 1.7 for random sequences and 2.1 for Halton sequences. The difference is roughly log(2) which corresponds to a number of steps divided by around two. We have done other numerical experiments up to a dimension equal to 6. We still notice that we need twice less points to achieve the desired accuracy if we use the quasi-random version of the algorithm. We can guess that the use of quasi-random sequences would not be so favourable in higher dimensions.

3. Approximation from a single set of points

To try to make an even better use of the data, the next step is to use the same ideas but from a single low- discrepancy sequence X_i of size N. This corresponds for instance to a physical situation when one wants to take the maximum of information from measurements. We now compute the corrections on the coefficients at each step of the algorithm using N_1 random variables Y_i which are taken uniformly with replacement from the X_i. This can be seen as random rational weights quadrature formulas involving the $f(X_i)$. We have to choose the value of N_1 with respect to N to make the algorithm as efficient as possible. We are in the following situation. Defining N i.i.d random variables W_i such that $E(W_i) = 0$ and $E(W_i^2) = \sigma^2$, we have to compute the variance of $\bar{Z} = \frac{1}{N_1}\sum_{i=1}^{N_1} Z_i$ where the Z_i are taken uniformly with replacement from the W_i. We assume for the sake of simplicity that W_i takes p discrete values. We have $E(\bar{Z}) = 0$ and

$$E(\bar{Z}^2) = \frac{1}{N_1^2}\sum_{i=1}^{N_1}\sum_{j=1}^{N_1} E(Z_i Z_j) = \frac{E\big(Z_1^2\big)}{N_1} + \frac{N_1^2 - N_1}{N_1^2} E(Z_1 Z_2).$$

Furthermore

$$E(Z_1 Z_2) = \sum_{x=1}^{p}\sum_{y=1}^{p} xy P(Z_1 = x, Z_2 = y)$$

and

$$P(Z_1 = x, Z_2 = y) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} P(W_i = x, W_j = y).$$

As

$$\sum_{i=1}^{N}\sum_{j=1}^{N} P(W_i = x, W_j = y)$$

$$= \sum_{i,j,j\neq i} P(W_i = x, W_j = y) + \sum_{i=1}^{N} P(W_i = x, W_i = y)$$

and because $E(W_i) = 0$, we have

$$E(Z_1 Z_2) = \frac{1}{N}\sum_{x=1}^{p}\sum_{y=1}^{p} xy P(W_1 = x, W_1 = y)$$

$$= \frac{1}{N}\sum_{x=1}^{p} x^2 P(W_1 = x) = \frac{E\big(W_i^2\big)}{N}$$

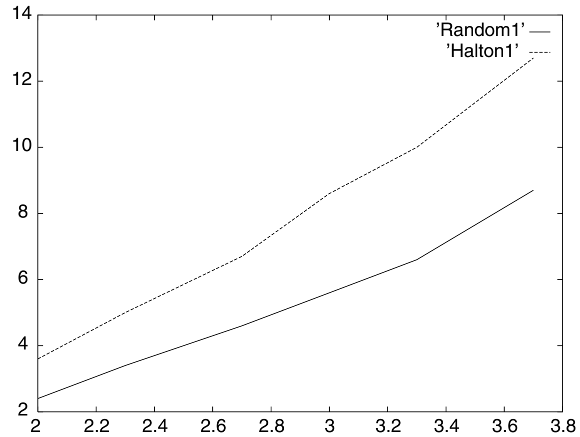
and at last

$$E(\bar{Z}^2) = \frac{E\big(W_1^2\big)}{N_1} + \frac{N_1^2 - N_1}{N_1^2}\frac{E\big(W_1^2\big)}{N} = \frac{N + N_1 - 1}{N N_1}\sigma^2.$$

This computation shows that the variance on the corrections depends on $\frac{N+N_1-1}{NN_1}$. Hence, we take from now on $N_1 = 3N$ to make the algorithm more stable and not too expansive. We have done numerical experiments using once again random sequences and Halton sequences to compute approximations of a function f as

$$f(t) \simeq \sum_{m\in W_{Q,d}} a_m e_m(t)$$

using this new version of the algorithm. The algorithm always converges if we take $N \geq 5L_{Q,d}$ using Halton sequences. We compare the accuracy we can reach from random data and quasi-random data of size N on the previous example.



Approximate values of the order of the method are respectively 3 for Halton sequences and 2 for random sequences which should be compared to respectively 1 and 0.5. This new version of the algorithm is significantly more accurate than the previous one. Its convergence is however really slow and quite erratic. No theorical proof of convergence is for the moment available. We give a last numerical example taken from Atanassov and Dimov (1999) of the linear transformation of the function

$$f_1(x, y, z, t) = \exp(x)\cos(y)\sin(z)\ln(1 + t)$$

from $[0, 1]^4$ into $[-1, 1]^4$ using the quasi-random version of the new algorithm. We compare the exact value of the integral

I of the linear transformation of f_1 over $[-1, 1]^4$ and its approximation $\tilde{I}$ in the following table.

d	$L_{4,d}$	$5L_{4,d}$	$\frac{\|I-\tilde{I}\|}{\|I\|}$
5	168	940	1.2×10^{-5}
10	504	2520	4.2×10^{-8}
15	880	4400	5.5×10^{-9}
30	2453	12265	1.7×10^{-10}
40	3650	18250	1.3×10^{-11}

For a given value of N, we of course achieve a better accuracy using this method than with our previous algorithm but also than with the method developed in Atanassov and Dimov (1999). All these numerical results show that we have made a better choice for the integration points and a better use of these points than with our initial Monte Carlo algorithm. Nevertheless, the new version of the algorithm may be of less practical interest because of its very slow speed of convergence which implies large CPU times.

The complexity of the algorithms studied here depends of their numbers of steps until convergence and of the cost of a step. At each step one has to compute $L_{Q,d}$ coefficients using $O(L_{Q,d})$ points which means that $O(L_{Q,d}^2)$ operations are required. As all the algorithms have a geometric convergence here, we can consider that the complexity of these algorithms depends mainly on the complexity of one step. It is not really easy to describe exactly how $L_{Q,d}$ depends on Q and d. We have nevertheless proved in Maire (2003b) that $\forall \varepsilon > 0$, there exists a constant $\theta(\varepsilon)$ such that $L_{Q,d} \leq \theta(\varepsilon) 2^Q d^{\varepsilon}$, which shows that $L_{Q,d}$ does not grows to quickly with Q and d.

4. Conclusion and future work

We have shown in this work that the use of quasi-random sequences has improved our previous algorithm Monte Carlo algorithm which computes approximations on reduced size polynomial bases. Moreover, these approximations can be achieved from the most accurate version of our algorithm using only $5L_{Q,d}$ points. For low dimensions, say less than 5, product rules are usually considered as a reasonable choice for the numerical integration of smooth functions. They require $(d + 1)^Q$ points to be built. As $(d + 1)^Q$ grows a lot quicker than $5L_{Q,d}$ when d increases, we can expect to achieve a better accuracy with our algorithm then with product rules even for small values of Q. This is especially true for functions which are not too polynomial like. We can also add that fitting directly the random or quasi-random values to the approximation model seems to be even more promising. It could especially allow us to make an exhaustive use of the data and to reduce drastically the CPU times. An other possible improvement is to try to make an even better choice of the points which represents the density defined as a product of Tchebychef weights by using other quasi-random sequences or by building these points from a quantization problem (Pages 1997).

References

Atanassov E.I. and Dimov I.T. 1999. A new optimal Monte Carlo method for calculating integral of smooth functions. Monte Carlo Methods and Appl. 5(2): 149–167.

Bernardi C. and Maday Y. 1992. Approximations spectrales de problèmes aux limites elliptiques. Springer-Verlag.

Kalos M.H. and Whitlock P.A. 1986. Monte Carlo Methods. John Wiley & Sons.

Krommer A.R. and Ueberhuber A.R. 1998. Computational integration. SIAM.

Lapeyre B., Pardoux B., and Sentis B. 1998. Méthodes de Monte-Carlo pour les équations de transport et de diffusion. Springer-Verlag.

Lepage G.P. 1978. A new algorithm for adaptative multidimensional integration. Journal of Computational Physics 27: 192–203.

Maire S. 2003a. Reducing variance using iterated control variates. The Journal of Statistical Computation and Simulation 73(1): 1–29.

Maire S. 2003b. An iterative computation of approximations on Korobov-like spaces. Journal of Computational and Applied Mathematics 157: 261–281.

Niederreiter H. 1987. Point sets and sequences with small discrepancy. Monatsh. Math. 104: 273–337.

Niederreiter H. 1992. Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia.

Pages G. 1997. A space vector quantization for numerical integration. Journal of Computational and Applied Mathematics 89: 1–38.

Sobol I.M. 1967. The distibutions of points in a cube and the approxiamate evaluation of integrals. Zh. Vychisl. Mat. I. Mat. Fiz 784–802.