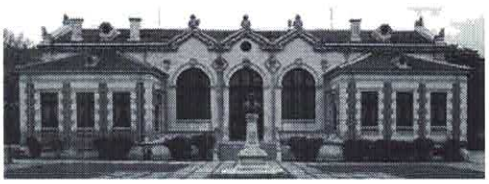


Journal scientific and applied research, vol. 10, 2016
Association Scientific and Applied Research
International Journal

Original Contribution ISSN 1314-6289

LATIN HYPERCUBE AND IMPORTANCE SAMPLING ALGORITHMS FOR MULTIDIMENSIONAL INTEGRALS

Venelin Todorov[1], Valerij Dzurov[2],Petar Stojanov[2] ,Angel Angelov[2]

[1] IICT, BULGARIAN ACADEMY OF SCIENCES
[2] ROUSSE UNIVERSITY "ANGEL KANCHEV"
E-MAIL: venelinltodorov@gmail.bg, vdzhurov@yahoo.com

ABSTRACT: *Monte Carlo method is the only viable method for high-dimensional problems since its convergence is independent of the dimension. In this paper we implement and analyze the computational complexity of the Latin hypercube sampling algorithm. We compare the results with Importance sampling algorithm which is the most widely used variance reduction Monte Carlo method. We show that the Latin hypercube sampling has some advantageous over the importance sampling technique.*

KEYWORDS: *Monte Carlo algorithms, multidimensional integrals, Latin hypercube sampling, Importance sampling.*

1 Introduction

Monte Carlo methods always produce an approximation to the solution of the problem or to some functional of the solution, but one can control the accuracy in terms of the probability error [4]. An important advantage of the Monte Carlo methods is that they are suitable for solving multi-dimensional problems [3], since the computational complexity increases polynomially, but not exponentially with the dimensionality. Another advantage of the method is that it allows to compute directly functionals of the solution with the same complexity as to determine the solution. In such a way this class of methods can be considered as a good candidate for treating innovative problems related to modern areas in quantum physics and finance. The Monte Carlo methods for multidimensional integrals can be used in problems where the data is taken in randomized way such that locating targets or searching for radio signals. Multidimensional numerical quadratures are of great importance in these areas.

The crude Monte Carlo method has rate of convergence $O(N^{-1/2})$, which is independent of the dimension of the integral, and that is why Monte Carlo integration is the only practical method for many high-dimensional problems. Much of the efforts to improve Monte Carlo are in construction of variance reduction methods which speed up the computation. Importance sampling (IS) is probably the most widely used Monte Carlo variance reduction method, [1]. One use of importance sampling is to emphasize rare but important events, i.e., small regions of space in which the integrand is large [5]. One of the difficulties in this method is that sampling from the importance density is required, but this can be performed using acceptance-rejection. It is also known that importance sampling can greatly increase the variance in some cases, [10]. In Hesterberg (1995, [6]) a method of defensive important sampling is presented; when combined with suitable control variates, defensive importance sampling produces a variance that is never worse than the crude Monte Carlo variance, providing some insurance against the worst effects of importance sampling. Defensive importance sampling can however be much worse than the original importance sampling. Owen and Zhow (1999) recommend an importance sampling from a mixture of m sampling densities with m control variates, one for each mixture component. In [10] it is shown that this method is never much worse than pure importance sampling from any single component of the mixture.

The layout of the rest of this paper is as follows. The next Section 2 discusses the problem that we are considered in this paper. Section 3 describes the Latin hypercube sampling algorithm (LHS). The numerical experiments and results are given in Section 4. Finally, some conclusions have been made concerning the advantages and disadvantages of the algorithms studied.

2 Description of the problem

Consider the problem of approximate integration of the multiple integral:

$$\int_{[0,1)^d} f(\mathrm{x})d\mathrm{x} = \int_0^1 d\mathrm{x}^{(1)} \int_0^1 d\mathrm{x}^{(2)} \dots \int_0^1 d\mathrm{x}^{(\mathrm{d})} f(x^{(1)}, x^{(2)}, \dots, x^{(d)}),$$

where $x = (x^{(1)}, \dots, x^{(d)}) \in [0,1)^d$.

For small values of d, numerical integration methods such as Simpson's rule or the trapezoidal rule (see Davis and Rabinowitz [2]) can be used to approximate the integral (1). These methods, however, suffer from the so-called curse of dimensionality and be-come impractical as s increases beyond 3 or 4. The Crude Monte Carlo method has rate of convergence $O(N^{-1/2})$, where N is the number of samples, which is independent of the dimension of the integral, and that is why Monte Carlo integration is the only practical method for many high-dimensional problems.

3 Latin Hypercube Sampling

The main problem of some of the widely used Monte Carlo methods such as Importance sampling is that a sample that is very close to another does not provide much new information about the function being integrated. One powerful variance-reduction technique that addresses this problem is called stratified sampling. Stratified sampling works by splitting up the original integral into a sum of integrals over sub-domains. In its simplest form, stratified sampling divides the domain G_d into N sub-domains (or stratas) and places a random sample within each of these intervals. A quantity of interest is the variance of the obtained approximation, considered as a random variable. It can be shown that stratified sampling can never result in higher variance than pure random sampling [11]. If N = 1, we have random sampling over the entire sample space (see [12]).

The Latin hypercube sampling (LHS) was described by McKay in 1979 [8]. If we wish to en-sure that each of the input variables x_i has all portions of its distribution represented by input values, we can divide the range of each x_i, in our case the interval [0,1], into M strata of equal marginal probability 1/M, and sample once from each stratum. In the case of integral approximation we must simply divide the interval [0,1] into M disjoint intervals, each of length 1/M and to sample one point from each of them. Let this sample be X_{kj}, for dimensions k = 1,..,d, j = 1,..., M. Those of them having first index k (k = 1,..., d) are the different components for the k-th dimension of the random points that are used for the integral's approximation. These components are matched at random. Thus the maximum number of combinations for a Latin Hypercube of M divisions and s variables (i.e., dimensions) can be computed by the formula $(M!)^{d-1}$. In the context of statistical sampling, a square grid containing sample positions is a Latin square if (and only if) there is only one sample in each row and each column. Thus following the described algorithm, we obtain a set of points with positions forming a Latin square. Note that this sampling scheme does not require more samples for more dimensions (variables); this independence is one of the main advantages of LHS scheme. In LHS one must first decide how many sample points to use and for each sample point remember in which row and column the sample point was taken.

Note that such configuration is similar to having N rooks on a chess board without threatening each other.

There are examples of random, stratified and Latin hypercube samplings with 16 points in [7]. A Latin hypercube is the generalisation of this concept to an arbitrary number of dimensions, whereby each sample is the only one in each axis-aligned hyperplane containing it. To prove that the variance of the LHS is smaller than the variance of IS we use a theorem proved in [8].

Generally, the time complexity of the algorithm depends on the integrand. However, it follows easily that the computational complexity of the Latin

hypercube sampling is linear, we will have only a constant number of additional operations compared to the regular Crude MC method and it is easy to show that the computational complexity of the crude Monte Carlo is linear.

4 Numerical example

We want to compute the following 7 dimensional integral:

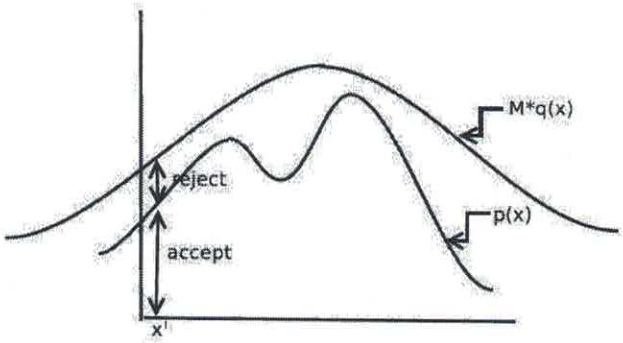
$$\int_{[0,1]^7} e^{1-\sum_{i=1}^{3} sin(\frac{\pi}{2}.x_i)}.arcsin\left(sin(1)+\frac{\sum_{j=1}^{7} x_j}{200}\right) \approx 0.7515.$$

We will use the IS algorithm with probability density function

$$f(\overline{x}) = \frac{1}{\eta} e^{1-\sum_{i=1}^{3} sin(\frac{\pi}{2}.x_i)}.$$

The value of η must be found separately. It is equal to the value of the integral $\int_{[0,1]^3} e^{1-\sum_{i=1}^{3} sin(\frac{\pi}{2}.x_i)}$.

We evaluate the last integral with Crude Monte Carlo method for a number of samples N = 10^5. After that we use the method of selection (the acceptance-rejection method). The idea of the method is given by the following graphics:



We choose density q(x), and constant M, such that f(x) ≤ M.q(x), then we generate a random variable with density q(x) and its value x_0 is accepted with probability density function p(x_0). In our realization we use the probability density function q(x) = 1 and constant M=e/η. We are also using two quasi-Monte Carlo methods just for a quick comparison. The detailed comparison between Sobol algorithm and Lattice rule based on Fibonacci numbers will be an object of a future study. We obtain the following graphics given below for the relative error and the total amount of time for performing each of the algorithms- see Figure 1 and Figure 2.

Now we make a comparison between the Latin hypercube sampling (LHS) and the importance sampling (IS) and crude Monte Carlo which is the goal of this paper. As we expected LHS outperforms IS by far. The results are given in the tables below. Each table contains information about the MC approach which is applied, the obtained relative error, the needed CPU time and the number of points.

It can be seen than the CPU time for the Latin hypercube sampling and crude MC method is equal, while importance sampling method needs much higher time. It can be seen that for large number of samples the LHS outperforms the importance sampling. Therefore the Latin hypercube sampling is definitely better from the Importance sampling method for computing multidimensional integrals.

Table 1. Relative error for 7 dimensional integral for given number of samples

N	Crude	time	LHS	time	IS	time
1000	7.28e-2	0.15	2.46e-3	0.13	6.61e-3	3.1
5000	2.71e-2	0.75	9.03e-4	0.71	3.35e-3	15.8
10000	5.92e-3	1.52	4.43e-4	1.50	1.09e-3	30.2
25000	3.73e-3	3.51	1.10e-4	3.27	7.51e-4	74.5
100000	1.62e-3	13.87	7.26e-5	14.02	5.41e-4	315
1000000	8.02e-4	104	6.26e-6	110	2.53e-4	3056

Table 2. Relative error for 7 dimensional integral and equal execution times

time,sec.	Crude	LHS	IS
0.1	2.38e-2	2.37e-3	5.51e-2
1	8.87e-3	3.37e-4	2.31e-2
5	5.16e-3	1.38e-4	8.05e-3
10	1.28e-3	8.78e-5	4.91e-3
20	2.03e-3	6.87e-5	2.58e-3
100	8.61e-4	7.01e-6	7.18e-4

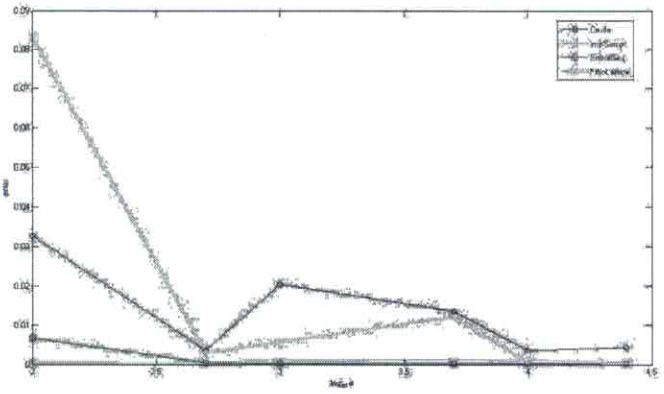


Figure 1. Relative error for 7 dimensional integral with importance sampling and crude method.

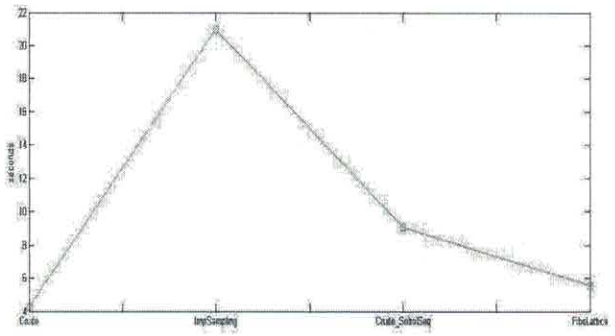


Figure 2. The total amount of time for the importance sampling and crude method for the 7 dimensional integral.

5. Conclusion

In this paper we present and study the Importance sampling and Latin hypercube sampling algorithms. We compare the results with crude Monte Carlo method. The developed Latin hypercube sampling algorithm shows the best results and can be used in various problems in science where multidimensional integrals are needed. The Importance sampling algorithm requires higher computational time than the crude Monte Carlo method, but we show that for a fixed computational time it gives closer accuracy to the crude Monte Carlo method. However, Latin hypercube sampling is far better than the Importance sampling in terms of lower relative error and shorter computational time. Therefore one have to use the Latin hypercube sampling algorithm instead of Importance sampling algorithm in evaluating multidimensional integrals arising in different areas of science.

Acknowledgement

This work was supported by the Program for career development of young scientists, BAS, Grant No. DFNP-91/04.05.2016, as well as by the Bulgarian National Science Fund under grant DFNI I02-20/2014.

References

[1]. Caflisch, R. E.: Monte Carlo and quasi-Monte Carlo methods. Acta Numerica, 7 (1998) 149.

[2]. Davis P. and Rabinowitz. P.: Methods of Numerical Integration. Academic Press, London, 2nd edition, (1984).

[3]. Dimov I., Atanassov E.: What Monte Carlo models can do and cannot do efficiently?, Applied Mathematical Modelling 32 (2007) 1477–1500.

[4]. Dimov I.: Monte Carlo Methods for Applied Scientists, New Jersey, London, Singapore, World Scientific, (2008), 291p.

[5]. Dimov I., Karaivanova A., Georgieva R., Ivanovska S.: Parallel Importance Separation and Adaptive Monte Carlo Algorithms for Multiple Integrals, 5th Int. conf. on NMA, August, 2002, Borovets, Bulgaria, Springer Lecture Notes in Computer Science, 2542, (2003), Springer-Verlag, Berlin, Heidelberg, New York, pp. 99- 107.

[6]. Hesterberg, T.: Weighted average importance sampling and defensive mixture distributions. Technometrics, 37(2) (1995) 185194.

[7]. Jarosz, W.: Efficient Monte Carlo Methods for Light Transport in Scattering Media, PhD dissertation, UCSD, (2008)

[8]. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 21(2), 23945 (1979)

[9]. Michael I. Jordan, Stat260: Bayesian Modeling and Inference- Monte Carlo sampling, (2010).

[10]. Owen, A. and Zhou, Y.: Safe and effective importance sampling. Technical report, Stanford University, Statistics Department (1999)

[11]. Vose, D.: The pros and cons of Latin Hypercube sampling, (2014)

[12]. http://www.columbia.edu/ ~mh2078/MCS04/MCS_var_red2.pdf