Proceedings of the 16th Conference on Computer Science and Intelligence Systems pp. 243–246
DOI: 10.15439/2021F53
ISSN 2300-5963 ACSIS, Vol. 25

Optimized Method based on Lattice Sequences for Multidimensional Integrals in Neural Networks

Venelin Todorov *†, Ivan Dimov†, Stefka Fidanova †
*Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
8 Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
†Institute of Information and Communication Technologies
Bulgarian Academy of Sciences
25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
Email: vtodorov@math.bas.bg, venelin@parallel.bas.bg, ivdimov@bas.bg, stefka@parallel.bas.bg

***Abstract*—In this work we investigate advanced stochastic methods for solving a specific multidimensional problem related to neural networks. Monte Carlo and quasi-Monte Carlo techniques have been developed over many years in a range of different fields, but have only recently been applied to the problems in neural networks. As well as providing a consistent framework for statistical pattern recognition, the stochastic approach offers a number of practical advantages including a solution to the problem for higher dimensions. For the first time multidimensional integrals up to 100 dimensions related to this area will be discussed in our numerical study.**

I. Introduction

IN 2011 Shaowei Lin in his works [5],[6] consider the problem of evaluating multidimensional integrals in Bayesian statistics which are used in neural networks. The first has the form

$$\int_{\Omega} p_1^{u_1}(x)\dots p_s^{u_s}(x)dx, \tag{1}$$

where $\Omega \in \mathcal{R}^s$, $x = (x_1, \dots, x_s)$, $p_i(x)$ are polynomials and u_i are integers. The second kind of integrals has the form

$$\int_{\Omega} e^{-Nf(x)}\phi(x)dx, \tag{2}$$

where $f(x)$ and $\phi(x)$ are multidimensional polynomials and N is an integer number. These integrals are evaluated unsatisfactory with deterministic [11] and algebraic methods [9] up to now, and it is known that Monte Carlo methods [3] outperform these methods especially for high dimensions [12].

We will now give a brief explanation which demonstrates the strength of the MC and QMC approach [3]. According to [3] we will choose 100 nodes on the each of the coordinate axes in the s-dimensional cube $G = E^s$ and we have to evaluate about 10^{100} values of the function $f(x)$. Assume a time of $10^{-7}s$ is necessary for calculating one value of the function [3]. So, a time of order 10^{93}s will be necessary for computation of the integral, and 1 year has 31536×10^3s.

Now MC approach consists of generating N pseudo random values (points) (PRV) in G; in evaluating the values of $f(x)$ at these points; and averaging the computed values of the function. For each uniformly distributed random (UDR) point in G we have to generate 100 UDR numbers in $[0, 1]$. Assume that the expression in front of h^{-6} is of order 1 [3]. Here $h = 0.1$, and we have $N \approx 10^6$; so, it will be necessary to generate $100 \times 10^6 = 10 \times 10^7$ PRV. Usually, 2 operations are sufficient to generate a single PRV. According to [3] the time required to generate one PRV is the same as that for computation the value of $f(x)$. So, in order to solve the task with the same accuracy, a time of

$$10 \times 10^7 \times 2 \times 10^{-7} \approx 20s$$

will be necessary. We summarize that in the case of 100-dimensional integral it is 5×10^{91} times faster than the deterministic one. That motivates our study on the new highly efficient stochastic approaches for the problem under consideration.

II. The new stochastic approach

We will use this rank-1 lattice sequence [10]:

$$\mathbf{x}_k = \left\{ \frac{k}{N}\mathbf{z} \right\}, \ k = 1, \dots, N, \tag{3}$$

where N is an integer, $N \geq 2$, $\mathbf{z} = (z_1, z_2, \dots z_s)$ is the generating vector and $\{z\}$ denotes the fractional part of z. For the definition of the $E_s^{\alpha}(c)$ and $P_{\alpha}(z, N)$ see [10] and for more details, see also [1].

Definition 1: Consider the point set $X = \{x_i \mid i = 1, 2, \dots N\}$ in $[0, 1)^s$ and $N > 1$. Denote by $x_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(s)})$ and $J(v) = [0, v_1) \times [0, v_2) \times \dots \times [0, v_s)$. Then the discrepancy of the set is defined as

$$D_N^* := \sup_{0 \leq v_j \leq 1} \left| \frac{\#\{x_i \in J(v)\}}{N} - \prod_{j=1}^{s} v_j \right|. \tag{4}$$

Venelin Todorov is supported by the Bulgarian National Science Fund under Project KP-06-M32/2 - 17.12.2019 "Advanced Stochastic and Deterministic Approaches for Large-Scale Problems of Computational Mathematics" and by the National Scientific Program "Information and Communication Technologies for a Single Digital Market in Science, Education and Security (ICT in SES)", contract No DO1-205/23.11.2018, financed by the Ministry of Education and Science in Bulgaria. The work is also supported by Bulgarian National Science Fund under Project DN 12/5-2017 "Efficient Stochastic Methods and Algorithms for Large-Scale Problems" and by the Project KP-06-Russia/17 "New Highly Efficient Stochastic Simulation Methods and Applications" funded by the National Science Fund - Bulgaria.

In 1959 Bahvalov proved that [1] there exists an optimal choice of the generating vector $\mathbf{z}$:

$$\left| \frac{1}{N} \sum_{k=1}^{N} f\left(\left\{ \frac{k}{N} \mathbf{z} \right\} \right) - \int\limits_{[0,1)^s} f(u)du \right| \leq cd(s,\alpha) \frac{(\log N)^{\beta(s,\alpha)}}{N^{\alpha}}, \tag{5}$$

for the function $f \in E_s^{\alpha}(c)$, $\alpha > 1$ and $d(s,\alpha), \beta(s,\alpha)$ does not depend on N.

The generating vector $\mathbf{z}$ which satisfies (5), is an optimal generating vector [10] and while the existence of optimal generating vectors is proved by the theoretical result, the main bottleneck lies in the construction of the optimal vectors, especially for very high dimensions [3].

The first generating vector in our study is the generalized Fibonacci numbers of the corresponding dimension:

$$\mathbf{z} = (1, F_n^{(s)}(2), \ldots, F_n^{(s)}(s)), \tag{6}$$

where we use that $F_n^{(s)}(j) := F_{n+j-1}^{(s)} - \sum_{i=0}^{j-2} F_{n+i}^{(s)}$ and $F_{n+l}^{(s)}$ ($l = 0, \ldots, j-1, j$ is an integer, $2 \leq j \leq s$) is the term of the s-dimensional Fibonacci sequence [10].

If we change the generating vector to be optimal in the way described in [4] we have improved the lattice sequence. We will now give the description of the steps of our algorithms. At the beginning of the algorithm the input is the number of dimensionality s and the number of samples N. At the first step of the algorithm s dimensional optimal generating vector

$$\mathbf{z} = (z_1, z_2, \ldots z_s) \tag{7}$$

is generated by the fast construction method described by Dirk Nuyens [4]. The second step of the algorithm includes generating the points of lattice rule by formula

$$\mathbf{x}_k = \left\{ \frac{k}{N} \mathbf{z} \right\}, \ k = 1, \ldots, N. \tag{8}$$

And at the third and last step of the algorithm an approximate value I_N of the multidimensional integral is evaluated by the formula:

$$I_N = \frac{1}{N} \sum_{k=1}^{N} f\left(\left\{ \frac{k}{N} \mathbf{z} \right\} \right). \tag{9}$$

The special choice of this optimal generating vector is definitely more efficient than the Fibonacci generating vector, which is only optimal for the two dimensional case [10]. For our improved lattice rule is satisfied [4]:

$$D_N^* = \mathcal{O}\left(\frac{log^s N}{N} \right). \tag{10}$$

The steps of working of the algorithm are given on the flowchart on Fig. 1.

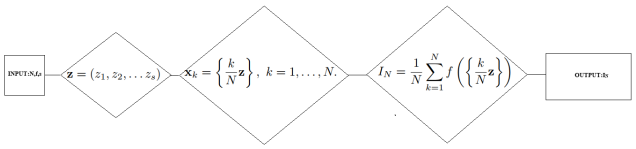

Figure 1. The flowchart of the optimized lattice algorithms

III. Numerical results

We considered different examples of 4,7,10,30 and 100 dimensional integrals, respectively, for which we have computed their referent values.

Example 1. s = 4.

$$\int\limits_{[0,1]^4} x_1 x_2^2 e^{x_1 x_2} \sin(x_3) \cos(x_4) \approx 0.108975. \tag{11}$$

Example 2. s = 7.

$$\int\limits_{[0,1]^7} e^{1 - \sum\limits_{i=1}^{3} sin(\frac{\pi}{2} \cdot x_i)} .arcsin\left(sin(1) + \frac{\sum\limits_{j=1}^{7} x_j}{200} \right) \approx 0.7515. \tag{12}$$

Example 3. s = 10.

$$\int\limits_{[0,1]^{10}} \frac{4x_1 x_3^2 e^{2x_1 x_3}}{(1 + x_2 + x_4)^2} e^{x_5 + \cdots + x_{10}} \approx 14.808435. \tag{13}$$

Example 4. s= 30.

$$\int\limits_{[0,1]^{30}} \frac{4x_1 x_3^2 e^{2x_1 x_3}}{(1 + x_2 + x_4)^2} e^{x_5 + \cdots + x_{20}} x_{21} \ldots x_{30} \approx 3.244. \tag{14}$$

We also consider the 100-dimensional multidimensional integral defined by the following way:

Example 5. s= 100.

$$I_{100} = \int\limits_{[0,1]^{100}} \exp\left(\prod_{i=1}^{100} x_i \right), \tag{15}$$

whose reference value is calculated by expanding the exponential function in Taylor series and integrating the terms $(x_1 \cdots x_{100})^n$ namely

$$\int_{[0,1]^{100}} \exp\left(\prod_{i=1}^{100} x_i \right) =$$

$$= \sum_{n=0}^{\infty} \frac{1}{(n+1)^{100} n!} =_{100} F_{100}(1, \cdots, 1; 2, \cdots, 2; 1),$$

where ${}_pF_q(a_1, \cdots, a_p; b_1, \cdots, b_q; x)$ is the generalized hypergeometric function

$${}_pF_q(a_1, \cdots, a_p; b_1, \cdots, b_q; x) = \sum_{n=0}^{\infty} \frac{(a_1)_n \cdots (a_p)_n}{(b_1)_n \cdots (b_q)_n} \frac{x^n}{n!},$$

and $(c)_n = c(c+1) \cdots (c+n-1)$ is the Pochhammer symbol.

We make a comparison between the optimized lattice sequence with an optimal generating vector (OPT), Fibonacci lattice sets (FIBO), Latin hypercube sampling (LHS) [7] and the scrambled Sobol sequence (SOBOLS) [8]. Each Table below contains information about the stochastic approach which is applied, the obtained relative errors (REs), the needed CPU-time in seconds and the number of points. Note that when the FIBO method is tested, the number of sampled points are always generalized Fibonacci numbers of the corresponding dimensionality. The computer working architecture is Core i7-4710MQ at 2.50GHz and 8GB of RAM. We performs 10 algorithmic runs using MATLAB on CPU Core i7-4710MQ for the algorithms to validate our assumptions of experimentation.

Table I
ALGORITHM COMPARISON OF THE REs FOR THE 4-DIMENSIONAL INTEGRAL FOR DIFFERENT NUMBER OF POINTS.

# of points	OPT	t,s	FIBO	t,s	LHS	t,s	SOBOLS	t,s
1490	6.11e-4	0.002	1.01e-3	0.004	8.16e-4	0.005	3.78e-3	0.47
10671	2.13e-5	0.01	8.59e-5	0.02	6.11e-4	0.01	6.10e-4	1.59
20569	6.56e-6	0.02	3.89e-5	0.03	5.01e-5	0.02	1.97e-5	4.54
39648	9.14e-7	0.06	3.01e-5	0.07	4.18e-5	7.09	9.67e-6	8.26
147312	4.78e-7	0.15	3.71e-6	0.24	2.19e-5	0.28	1.40e-6	27.91

Table II
ALGORITHM COMPARISON OF THE REs FOR THE 4-DIMENSIONAL INTEGRAL FOR A PRELIMINARY GIVEN TIME.

t, s	OPT	FIBO	LHS	SOBOLS
1	5.66e-7	5.62e-6	1.54e-5	6.32e-4
5	3.12e-7	5.38e-7	9.18e-6	1.23e-5
10	5.14e-8	3.77e-7	6.51e-6	8.48e-6
20	3.18e-8	2.67e-8	2.31e-6	1.16e-6

Table III
ALGORITHM COMPARISON OF THE REs FOR THE 7-DIMENSIONAL INTEGRAL FOR DIFFERENT NUMBER OF POINTS.

# of points	OPT	t,s	FIBO	t,s	LHS	t,s	SOBOLS	t,s
2000	6.39e-4	0.14	2.81e-3	0.23	5.45e-3	0.25	2.51e-3	1.42
7936	3.23e-4	0.64	1.38e-3	0.87	2.11e-3	0.91	1.16e-3	3.08
15808	1.23e-5	0.95	9.19e-4	1.73	8.31e-4	1.81	7.58e-4	5.89
62725	3.15e-6	2.54	2.78e-5	3.41	6.22e-4	3.5	3.11e-4	15.64
124946	1.12e-6	6.48	6.87e-5	6.90	4.34e-4	7.1	8.22e-5	31.41

Numerical results show significant advantage for the optimized lattice sets algorithm based on an optimal generating vector in comparison with FIBO, LHS and SOBOLS scramble sequence (1-2 orders). For the 4-th dimensional integral the best approach is produced by the optimized method OPT - a relative error $4.78e-7$ for $N = 147312$ - see Table I and for 20s the best approach is FIBO - $2.67e-8$ in Table II with two orders better results than both SOBOLS and LHS. For the 7-th dimensional integral the best approach is produced by the optimized method OPT - a relative error $1.12e-6$ for $N = 124946$ - see Table III and for 20s the best approach is OPT - $7.43e-7$ in Table IV with one order better REs than FIBO and two order better REs than both SOBOLS and LHS. For the 10-th dimensional integral the best approach is produced by the optimized method OPT for $N = 3524578$ the

Table IV
ALGORITHM COMPARISON OF THE REs FOR THE 7-DIMENSIONAL INTEGRAL FOR A PRELIMINARY GIVEN TIME.

t, s	OPT	FIBO	LHS	SOBOLS
0.1	7.38e-4	2.38e-3	6.65e-3	8.37e-3
1	1.17e-5	6.19e-4	3.05e-3	1.37e-3
5	2.32e-6	8.81e-5	4.89e-4	8.38e-4
10	9.11e-7	1.88e-5	2.16e-4	4.78e-4
20	7.43e-7	3.87e-6	8.56e-5	9.87e-5

Table V
ALGORITHM COMPARISON OF THE REs FOR THE 10-DIMENSIONAL INTEGRAL FOR DIFFERENT NUMBER OF POINTS.

# of points	OPT	t,s	FIBO	t,s	LHS	t,s	SOBOLS	t,s
1597	3.14e-4	0.002	4.39e-3	0.003	7.31e-3	0.01	1.46e-3	0.05
17711	6.21e-5	0.02	1.81e-3	0.04	4.45e-3	0.07	1.83e-4	0.21
121393	4.34e-6	0.15	1.20e-3	0.16	7.23e-4	0.21	3.12e-5	1.47
832040	4.11e-7	0.75	1.19e-5	0.70	3.11e-4	0.83	8.25e-6	14.41
3524578	5.32e-8	6.35	2.63e-6	6.45	8.57e-5	6.7	7.71e-7	139.1

Table VI
ALGORITHM COMPARISON OF THE REs FOR THE 10-DIMENSIONAL INTEGRAL FOR A PRELIMINARY GIVEN TIME.

t, s	OPT	FIBO	LHS	SOBOLS
0.1	4.95e-6	9.19e-6	4.13e-3	4.19e-4
1	8.10e-7	5.63e-6	2.55e-4	1.21e-4
5	3.56e-8	2.15e-6	1.23e-4	7.21e-5
10	4.31e-8	1.79e-6	7.17e-5	3.51e-5
20	9.13e-9	8.61e-7	3.42e-5	7.09e-6

Table VII
ALGORITHM COMPARISON OF THE REs FOR THE 30-DIMENSIONAL INTEGRAL FOR DIFFERENT NUMBER OF POINTS.

# of points	OPT	t,s	SOBOLS	t,s	LHS	t,s	FIBO	t,s
1024	1.21e-2	0.02	5.78e-2	0.53	5.68e-2	0.03	8.81e-1	0.02
16384	4.11e-3	0.16	1.53e-2	5.69	8.60e-3	0.18	6.19e-1	0.14
131072	5.24e-4	1.34	1.35e-3	42.1	5.38e-3	1.2	2.78e-1	1.16
1048576	8.81e-5	9.02	6.78e-4	243.9	9.31e-4	8.9	9.86e-2	8.61

Table VIII
ALGORITHM COMPARISON OF THE REs FOR THE 30-DIMENSIONAL INTEGRAL FOR A PRELIMINARY GIVEN TIME.

t, s	OPT	SOBOLS	LHS	FIBO
1	3.48e-3	2.38e-2	7.21e-3	2.38e-1
5	4.23e-4	5.46e-3	5.16e-3	1.81e-1
10	8.91e-5	1.25e-3	8.21e-4	9.48e-2
20	2.33e-5	6.11e-4	4.35e-4	7.87e-2

Table IX
ALGORITHM COMPARISON OF THE REs FOR THE 100-DIMENSIONAL INTEGRAL FOR DIFFERENT NUMBER OF POINTS.

# of points	OPT	t,s	FIBO	t,s	LHS	t,s	SOBOLS	t,s
2^{10}	5.18e-3	0.05	4.13e-1	0.06	5.18e-2	0.08	6.31e-2	18
2^{12}	3.18e-3	0.17	1.15e-1	0.18	3.22e-2	0.2	1.23e-2	34
2^{16}	1.44e-4	9.1	6.12e-2	9.2	8.32e-3	9.7	2.31e-3	170
2^{20}	6.38e-5	57.6	3.18e-2	58.7	4.51e-3	60	2.34e-4	861

Table X
ALGORITHM COMPARISON OF THE REs FOR THE 100-DIMENSIONAL INTEGRAL FOR A PRELIMINARY GIVEN TIME.

t, s	OPT	FIBO	LHS	SOBOLS
1	2.14e-3	7.18e-2	2.83e-2	9.31e-2
2	1.56e-3	6.02e-2	1.17e-2	8.66e-2
10	2.58e-4	4.12e-2	8.34e-3	6.94e-2
100	8.86e-6	1.13e-2	1.18e-3	3.88e-3

RE is $5.32e-8$ - see Table V and for 20s the best approach is again OPT - $9.13e-9$ in Table VI with two order better REs than FIBO and 3-4 order better REs than both SOBOLS and LHS. For the 30-th dimensional integral the best approach is produced by the optimized method OPT for $N = 1048576$ $8.81e-5$ - see Table VII and for 20s the best approach is again OPT - $2.33e-5$ in Table VIII with one order better REs than both SOBOLS and LHS and 3 order better REs than both FIBO, which shows that FIBO becomes inefficient for high dimensions. Finally, for the 100-th dimensional integral the best approach is produced by the optimized method OPT for $N = 2^{20}$ the RE is $6.38e-5$ - see Table IX and for 20s the best approach is OPT - $8.86e-6$ in Table X with 4 order better REs than FIBO and 3 orders better REs than SOBOLS and LHS. From all Tables we can conclude that the optimized lattice sequence OPT, used for the first time for the evaluation of this type of multidimensional integrals up to 100 dimensions, gives the best results compared to the other stochastic approaches with increasing the dimensionality of the multidimensional integral.

IV. CONCLUSION

In this paper an optimized lattice rule has been tested on multidimensional integrals reflated to neural networks up to 100 dimensions. A comprehensive experimental study of optimized lattice rule, Fibonacci lattice sets, Sobol scrambled sequence and Latin hypercube sampling has been done on some case test functions. Our approach is one of the best available algorithms for high dimensional integrals and the only possible methods, because the deterministic algorithms need an huge amount of time for the evaluation of the multidimensional integral, as it was discussed in this paper. At the same time the new method is suitable to deal with 100-dimensional problems for less than a minute on a laptop. It is an important element since this may be crucial in order to achieve a more reliable interpretation of the results in Bayesian statistics which is foundational in neural networks, artificial intelligence and machine learning.

REFERENCES

[1] N. Bahvalov (1959) On the Approximate Computation of Multiple Integrals, *Vestnik Moscow State University* **4**, 3–18.

[2] Centeno, V., Georgiev, I. R., Mihova, V., Pavlov, V. (2019, October). Price forecasting and risk portfolio optimization. In AIP Conference Proceedings (Vol. 2164, No. 1, p. 060006). AIP Publishing LLC.

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

[4] F.Y. Kuo and D. Nuyens (2016) Application of quasi-Monte Carlo methods to elliptic PDEs with random diffusion coefficients - a survey of analysis and implementation, *Foundations of Computational Mathematics* **16**(6), 1631–1696.

[5] Lin S., "Algebraic Methods for Evaluating Integrals in Bayesian Statistics," Ph.D. dissertation, UC Berkeley, May 2011.

[6] Lin, S., Sturmfels B., Xu Z.: Marginal Likelihood Integrals for Mixtures of Independence Models, Journal of Machine Learning Research, Vol. 10, pp. 1611-1631, 2009.

[7] Minasny B., McBratney B.: A conditioned Latin hypercube method for sampling in the presence of ancillary information Journal Computers and Geosciences archive, Volume 32 Issue 9, November, 2006, Pages 1378-1388.

[8] S.H. Paskov, *Computing high dimensional integrals with applications to finance*, Technical report CUCS-023-94, Columbia University (1994).

[9] Song, J., Zhao, S., Ermon, S., A-nice-mc: Adversarial training for mcmc. In Advances in Neural Information Processing Systems, pp. 5140-5150, 2017.

[10] Wang Y., Hickernell F., *An historical overview of lattice point sets*, 2002.

[11] Watanabe S., Algebraic analysis for nonidentifiable learning machines. NeuralComput.(13), pp. 899—933, April 2001.

[12] S. L. Zaharieva, I. Radoslavov Georgiev, A. N. Borodzhieva and V. Angelov Mutkov, "Classical Approach For Forecasting Temperature In Residential Premises Part 1," 2021 20th International Symposium Infoteh-Jahorina (infoteh), 2021, pp. 1-6.