Randomized Algorithms to Approximate Discrepancies and Applications

Michael Gnewuch[1]

School of Mathematics and Statistics
University of New South Wales, Sydney

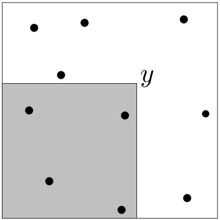
June 28, 2012

Joint Work with
Magnus Wahlström & Carola Winzen
Max Planck Institute of Computer Science, Saarbrücken

[1] Supported by the German Science Foundation (DFG) under grants GN-91/3 and GN-91/4

Star Discrepancy: Definition

$X \subset [0,1]^d$ n-point set, $[0,y) := [0,y_1) \times \cdots \times [0,y_d)$ "test box".



Local discrepancy: $\delta(y) = \delta(y,X) = \mathrm{vol}([0,y)) - \frac{1}{n}|X \cap [0,y)|$

Star discrepancy: $\mathrm{disc}^*(X) = \sup_{y \in [0,1]^d} |\delta(y,X)|$

Known Methods for Calculation

Elementary Method for Exact Calculation

For $X = \{x^1, \ldots, x^n\} \subset [0,1]^d$ put

$$\Gamma_j(X) := \{x^i_j \,|\, i = 1, \ldots, n\} \cup \{1\}\,, \;\; j = 1, \ldots, d\,,$$
$$\Gamma(X) := \Gamma_1(X) \times \cdots \times \Gamma_d(X)$$

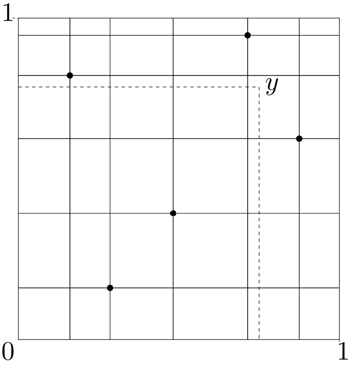
Then $\mathrm{disc}^*(X) =$

$$\max_{y \in \Gamma(X)} \max \left\{ \mathrm{vol}([0,y)) - \frac{1}{n} \big|[0,y) \cap X\big|\,,\; \frac{1}{n} \big|[0,y] \cap X\big| - \mathrm{vol}([0,y)) \right\}$$

Thus $\mathrm{disc}^*(X)$ can be calculated by considering at most $2(n+1)^d$ test boxes.

Known Methods for Calculation

Elementary Method for Exact Calculation



Simple Observation: It suffices to consider $2(n+1)^d$ test boxes to calculate the discrepancy.

Further Methods for Calculation

Improvements of the Elementary Method:

Exact formula for the star discrepancy in dimension $d = 1$ by Niederreiter '72 ($d = 1$).

Faster methods than the elementary one by De Clerck '86 ($d = 2$) and Bundschuh and Zhu '93 ($d \geq 3$). Time to calculate the star discrepancy still $O(n^d)$.

Fastest algorithm to calculate the star discrepancy needs time $O(n^{1+d/2})$ [Dobkin, Eppstein, Mitchell '96].

Known Methods for Calculation

Observation: Exact calculation of star discrepancy is discrete optimization problem.

Bad news from Discrete Complexity Theory:

Theorem [G., Srivastav, Winzen '09].
The calculation of the star discrepancy is NP-hard.

Theorem [Giannopoulos, Knauer, Wahlström, Werner '11].
Calculation of star discrepancy is $W[1]$-hard with respect to parameter d.

Methods for Approximation

Algorithms with error guarantee

- Algorithms from [Thiémard '00, '01a] to approximate the star discrepancy up to some user-specified error δ.

  Cost of the algorithms is $\geq \Omega(d\delta^{-d})$ [G.'08].

Algorithms based on optimization heuristics

- Algorithm from [Thiémard'01b] formulates problem as integer linear program (ILP) and relies on cutting plane and branch-and-bound techniques.
- Algorithm of Winker & Fang '97 is a local search algorithm relying on the meta heuristic "Threshold Accepting".
- Genetic algorithm of Shah '10.

Randomized Approach

Algorithm of Winker & Fang ("Threshold Accepting")

Idea: Calculate lower bound for star discrepancy by choosing test boxes randomly within a (refined) local search algorithm

Local neighborhood structure for $x^* \in \Gamma(X)$

$N_k(x^*) \simeq$ subgrid of $\Gamma(X)$ of cardinality $(2k+1)^d$ with center x^*,

with uniform distribution.

Algorithm of Winker & Fang

For $y \in [0,1]^d$ put

$$\delta(y) := \mathrm{vol}([0,y)) - \frac{1}{n}\big|[0,y) \cap X\big|, \qquad \overline{\delta}(y) := \frac{1}{n}\big|[0,y] \cap X\big| - \mathrm{vol}([0,y)),$$

and $\delta^*(y) := \max\{\delta(y), \overline{\delta}(y)\}$.

Choose threshold values $T_1 > T_2 > \cdots > T_I \geq 0$

Concrete Algorithm

Choose x randomly from $\Gamma(X)$ and put $x^* := x$.

For $i = 1$ to I
  For $j = 1$ to J
    Choose $x \in N_k(x^*)$ randomly
    If $\delta^*(x^*) - \delta^*(x) \leq T_i$ then $x^* := x$
Return $\delta^*(x^*)$

Improving on the W&F-Algorithm

Main Idea: Modification of the Neigborhoods

Endow neighborhoods $N_k(x)$ with new probability measure putting more weight on

- larger coordinates
- large gaps in grid $\Gamma(X)$

Concrete Approach:

$C_k(x) := \mathrm{conv}(N_k(x))$, endowed with probability measure

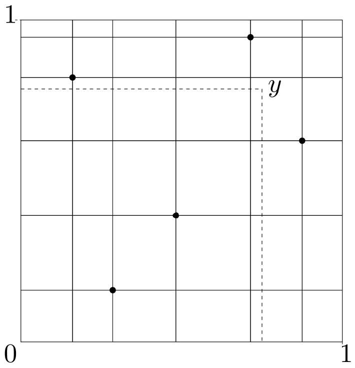
$$\mu_d := \bigotimes_{j=1}^{d} dy_j^{d-1}\, \lambda(\mathrm{d}y_j)\,,$$

λ the Lebesgue measure on $\mathbb{R}$. Choose $y \in C_k(x)$ randomly.

Further Improvement on the W&F-Algorithm

Procedures "snapping up" and "snapping down":

Rounding y up and down to critical test boxes $y^{+,\mathrm{sn}}$ and $y^{-,\mathrm{sn}}$.



Some Numerical Results

			$\text{disc}^*(\cdot)$	TA improved		Winker & Fang	
Name	d	n	found	Hits	Best-of-10	Hits	Best-of-10
Faure	7	343	0.1298	100	0.1298	0	0.1143
Faure	8	121	0.1702	100	0.1702	0	0.1573
Faure	9	121	0.2121	100	0.2121	0	0.1959
Faure	10	121	0.2574	100	0.2574	0	0.2356
Faure	11	121	0.3010	100	0.3010	0	0.2632
Faure	12	169	0.2718	100	0.2718	0	0.1708
Sobol'	50	2000	0.1030*	0	0.1024	0	0.0005
Sobol'	50	4000	0.0677*	0	0.0665	0	0.00025
Faure	50	2000	0.3112*	100	0.3112	0	0.0123
Faure	50	4000	0.1979*	0	0.1978	0	0.0059
GLP	50	2000	0.1465*	0	0.1450	0	0.0005
GLP	50	4000	0.1205*	0	0.1201	0	0.0003

Table: New instance comparisons. Discrepancy values marked with a star are lower bounds only (i.e., largest discrepancy found over all executions of algorithm variants). All data is computed using 100 trials of $100,000$ iterations; reported is the average value of best-of-10 calls, and number of times (out of 100) that the optimum (or a value matching the largest known value) was found.

Further Numerical Results

				TA improved		Shah	
Class	d	n	$\mathrm{disc}^*(\cdot)$	Hits	Best-of-10	Hits	Best Found
Halton	5	50	0.1886	100	0.1886	81	0.1886
Halton	7	50	0.2678	100	0.2678	22	0.2678
Halton	7	100	0.1714	100	0.1714	13	0.1714
Halton	7	1000	0.0430	81	0.0430	$8^{(1)}$	$0.0430^{(1)}$
Faure	10	50	0.4680	100	0.4680	97	0.4680
Faure	10	100	0.2483	100	0.2483	28	0.2483
Faure	10	500	0.0717*	100	0.0717	$0^{(1)}$	$0.0689^{(1)}$

Table: Comparison against point sets used by Shah. Reporting average value of best-of-10 calls, and number of times (out of 100) that the optimum was found; for Shah, reporting highest value found, and number of times (out of 100) this value was produced. The discrepancy value marked with a star is lower bound only (i.e., largest value found by any algorithm). Values marked (1) are recomputed using the same settings as in [Sha'10].

Applications

Quality Testing of Point Sets

[Doerr, G., Wahlström 2010]: Basic version of randomized TA algorithm used to compare quality of small samples generated by derandomized algorithm with classical low-discrepancy point sets and Monte Carlo points in dimension $d \leq 21$.

Generating Low-Discrepancy Point Sets via Optimization Approach

[De Rainville, Gagné, Teytaud, Laurendeau 2012]: Genetic algorithm used to optimize the choice of permutations for generalized Halton sequences with respect to Hickernell's modified L_2-discrepancy.

Work in progress: With our randomized TA algorithm the optimization should now be done with respect to star discrepancy.

Applications

Scenario Reduction in Stochastic Programming

Goal of Stochastic Programming: Solving optimization problems with random input data obeying law $\mathbb{P}$.

Common Approach: Approximate distribution $\mathbb{P}$ by discrete distribution P supported on N atoms ($=$ scenarios).

Practical Problem: Good approximation of $\mathbb{P}$ leads to large N, which is usually too large for efficient computation.

Scenario Reduction: Try to reduce the scenarios from N to $n << N$ by replacing P by new measure Q.

Quality of scenario reduction is measured in terms of discrepancy with respect to axis-parallel rectangles:

$$\mathrm{disc}_{\mathcal{R}}(P, Q) = \sup_{R \in \mathcal{R}} |P(R) - Q(R)|.$$

Started work with W. Römisch to design efficient algorithm for scenario reduction that uses modified TA randomized algorithm.

Some References

Algorithms for Discrepancy Approximation

1. M. Gnewuch, C. Winzen, M. Wahlström. *A new randomized algorithm to approximate the star discrepancy based on threshold accepting*, SIAM J. Numer. Anal. 50 (2012), 781–807.
2. M. Shah. *A genetic algorithm approach to estimate lower bounds of the star discrepancy*, Monte Carlo Methods Appl. 16 (2010), 379–398.
3. E. Thiémard. An algorithm to compute bounds for the star discrepancy, J. Complexity 17 (2001), 850–880.
4. E. Thiémard. Optimal volume subintervals with k points and star discrepancy via integer programming, Math. Methods Oper. Res. 54 (2001), 21–45.
5. P. Winker, K.-T. Fang. *Application of threshold accepting to the evaluation of the discrepancy of a set of points*, SIAM J. Numer. Anal. 34 (1997), 2028–2042.

Applications

1. F.-M. De Rainville, C. Gagné, O. Teytaud, D. Laurendeau. *Evolutionary optimization of low-discrepancy sequences*, ACM Trans. Model. Comp. Simul. 22 (2012), article 9.
2. B. Doerr, M. Gnewuch, M. Wahlström. *Algorithmic construction of low-discrepancy point sets via dependent randomized rounding*, J. Complexity 26 (2010), 490–507.

Modified Measure is Superior in High Dimension

GLP-sets: $0 < h_1 < h_2 < \cdots < h_d < n, \quad \exists j : \gcd(h_j, n) = 1$

$$T := \{t^1, \ldots, t^n\}\,,\ t^i_j := \left\{\frac{2ih_j - 1}{2n}\right\}\,,\ i = 1, \ldots, n\,,\ j = 1, \ldots, d$$

Mean values of coordinates of optimal test boxes for randomly chosen GLP-sets:

$$d = 4: \quad 0.799743$$
$$d = 5: \quad 0.840825$$
$$d = 6: \quad 0.873523$$

Expectation of coordinates of randomly chosen y with respect to μ_d is $d/(d+1)$:

$$d = 4: \quad 0.8$$
$$d = 5: \quad 0.8\overline{3}$$
$$d = 6: \quad 0.857143$$