Geometrical Intuition and Failure of Monte-Carlo in Multidimensional Optimization

Boris Polyak

with Pavel Scherbakov, Elena Gryazina

Institute for Control Science, Moscow

Seminar MIPT

March, 2017

Content

Blend of 3 topics

- Geometry of n dimensions
- Monte-Carlo method
- Optimization: MC and other grids

$\mathbb{R}^n$ contradicts our intuition

- Volume of n dimensional unit ball tends to zero

  $V_n(B_r) = v_n r^n, r$ radius of the ball $B_r = \{x : ||x|| \le r\}$,
  $v_n = \frac{2\pi}{n} v_{n-2}, v_1 = 2, v_2 = \pi, \max_n v_n = v_5 = 5.26$

- Unit cube $C = \{x : |x_i| \le 0.5\}$ is mostly outside unit ball: $\max_{x\in C} ||x|| = \sqrt{n}/2$

- The ball inscribed in the cube is a small part of its volume (ratio is $v_n 2^{-n}$).

- Volume of a ball is concentrated near the surface:

$$V_n(B_{1-1/n})/V_n(B_1) = (1 - \frac{1}{n})^n \simeq \frac{1}{e}$$

- Center of gravity x^* of a cone K lies near the bottom:

$$K = \{x : 1 \ge x_1 \ge \alpha\sqrt{x_2^2 + ... + x_n^2}\}, x_1^* = 1 - \frac{1}{n+1}$$

- Volume of spherical cap is small: $H = \{x : x_1 \ge h < 1, ||x|| \le 1\}$. $V_3(H)$ was calculated by Archimedes, for large $n, h = \frac{\varepsilon}{\sqrt{n-1}}, \frac{V_n(H)}{V_n(B_1)} \le f(\varepsilon)$.

The $\mathbb{R}^n$ geometry

John Hopcroft and Ravindran Kannan,

"Foundations of Data Science, 2014," Chapter 2, High-Dimensional Space

Monte Carlo: Beginning

J. von Neumann, E. Teller, S. Ulam and N. Metropolis (1949).

First applications

- Simulation (physical processes of particle motion and binary collisions).
- Multidimensional integration.
- Computational mathematics.

Recent applications

- Convex and global optimization
- Robustness analysis in control and optimization.
- Sampling in complex domains.
- Data mining, Learning, Signal processing, Image processing

Goal: Sampling in complicated domains

Possible approaches:

- Special regions (box, simplex, l_p-ball, positive definite matrices cone)
- Rejection
- Markov chain Monte Carlo (MCMC): random walk in the domain

1D Sampling

Long ago: Tables of random numbers, physical devices (analogs of roulette or coin).

Present: there exist highly effective pseudo-random generators, fast and with good statistical properties.

Example: routine `rand` in Matlab generates points uniformly distributed in $[0, 1]$.

How to generate random variables with a given cdf $F(x)$?

Solution: $x = \texttt{rand}, y = F^{-1}(x)$.

Gaussian $\mathcal{N}(0, 1)$: use routine `randn`; alternatively

$$x_1 = \texttt{rand}, x_2 = \texttt{rand},$$

$$y_1 = \sqrt{-2 \log x_1} \cos(2\pi x_2),$$

$$y_2 = \sqrt{-2 \log x_1} \sin(2\pi x_2),$$

Then $y \sim \mathcal{N}(0, I_2)$.

nD Sampling

1. $\texttt{rand}(n, 1)$ generates the uniform distribution over the box $B = [0, 1]^n \in \mathbb{R}^n$.
2. $\texttt{randn}(n, 1) \sim N(0, I_n)$.

Exercises: How to generate samples uniformly distributed in:

1) sphere; 2) ball; 3) simplex; 4) l_p-ball; 5) ellipsoid; 6) matrix balls

General problem: given a bounded set $Q \in \mathbb{R}^n$, how to generate points uniformly distributed in Q? Typical example: Q is a polytope.

Rejection: Take simple $G \supseteq Q$ (e.g. a box), generate points uniformly in G, reject those which are not in Q.

Bad: $Q = B_1$, $G = [-1, 1]^n$. Then $V_n(G)/V_n(Q) = v_n 2^{-n} = 4 \cdot 10^7$ for $n = 20$, that is we should generate 10^8 points to get 1–2 points in Q. For polytopes this ratio can be much larger.

Markov Chain Monte Carlo (MCMC)

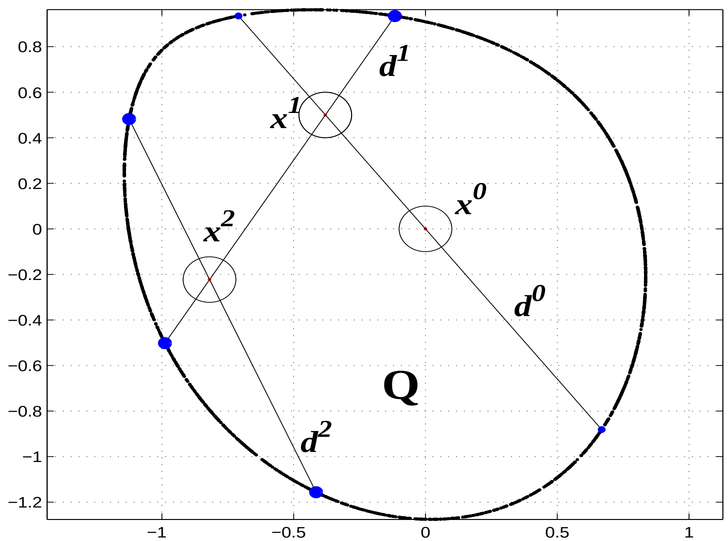
Idea: Apply random walks (vs independent generation of points as in MC).

P.Diaconis, The Markov Chain Monte Carlo Revolution, Bull. AMS, Vol. 46, No. 9, pp. 179–205 (2009).

A lot of applications in applied math.

Hit-and-Run algorithm (sampling in $Q \subset \mathbb{R}^n$)

This is random walk in Q. Turchin (1971), Smith (1984)



1. Initial point $x^0 \in Q$.
2. $d = s/||s||, s = randn(n, 1)$ — random direction uniform on the unit sphere
3. Boundary oracle: $L = \{t \in \mathbb{R} : x^0 + td \in Q\}$
4. Next point $x^1 = x^0 + t_1 d$, t_1 is uniform random in L.
5. x^0 is replaced by x^1, go to Step 2.

Hit-and-Run: properties

Theorem: The points x^k are asymptotically uniform in Q, i.e., the probability to reach a subset $A \subset Q$ can be estimated as: $|P_k(A) - P(A)| \leq cq^k$,

where $P(A) = Vol(A)/Vol(Q)$, $P_k(A) =$ (number of times x visits $xA)/k$

where $q < 1$ does not depend on the initial point x^0 (but it depends on the dimension and the geometry of the set).

Idea: for any sets $A \subset Q$, $B \subset Q$ with nonzero volume the probabilities of transition are equal and positive.

Advantages of HR

- Very simple.
- Q should not be neither convex nor connected (boundary oracle may consist of several intervals), the only assumption is that Q is a closure of an open set.
- Points on the boundary are also generated.
- Boundary oracle is available for numerous sets.

Extensions and different sampling algorithms (Shake-and-Bake, Billiard walk).

Polyak B., Gryazina E., "Random sampling: Billiard Walk algorithm", European Journal of Operational Research, 2014.

Optimization

Two different setups:

- Convex optimization
- Global optimization

Convex optimization

Advanced theory, efficient methods + validation, available software for various dimensions. No need for Monte-Carlo.

$\min f(x), x \in Q \subset \mathbb{R}^n, f, Q$ – convex.

"Best method": centers of gravity method.

Estimate $\min_{0\leq i\leq k}(f(x_k) - f^*) \leq cq^k, q \simeq (1 - \frac{1}{ne})$.

Example (K is the cone), the estimate is tight:

$$\min x_1, x \in K$$

Global optimization

Numerous heuristic methods (simulated annealing, tabu search, ant colony etc.) Randomization is highly helpful. Negative theoretical results: "Global optimization is unsolvable for $n \geq 15$ " Nemirovski, Yudin 1976; Nesterov 2009. Constrained minimization of Lipschitz functions on a cube in R^n, a function equals 0 everywhere beyond a cell of a grid; its minimum equals $-\varepsilon$. We need $(\frac{L}{2\varepsilon})^n$ calculations of the function to get into the cell.

Another example:

$$f(x) = \min\Big\{99 - c^\top x,\ \big(c^\top x - 99\big)/398\Big\}$$

to be minimized over the ball $B_{100} \subset \mathbb{R}^n$. It has one local minimum $x_1 = -100c$, $f_1 = -0.5$, and one global minimum $x^* = 100c$, $f^* = -1$. Any standard method (say, multi-start) misses global optimum with probability larger than $1 - 10^{-15}$ for $n = 15$.

MC for a ball

Conclusion: for hard problems Monte-Carlo-like methods have no competitors.

Question: how they behave for "good" problems?

Example: $\min f(x), x \in B, f(x) = (c, x), ||c|| = 1, f^* = -1$

Algorithm: $x^i, i = 1, ...k$ are sample of uniformly distributed points in the unit ball B, $f_k = \min_{1 \le i \le k} f(x^i)$.

Estimate How close is f_k to f^*?

Rigorous result

Theorem Given $p \in]0,\, 1[$ and $\delta \in]0,\, 1[$, the minimal sample size $N_{\min}$ that guarantees, with probability at least p, for the empirical maximum of f_N to be at least a δ-accurate estimate of f^*, is given by

$$N_{\min} = \frac{\ln(1-p)}{\ln\left[\frac{1}{2} + \frac{1}{2} I\left((1-\delta)^2; \frac{1}{2}, \frac{n+1}{2}\right)\right]},$$

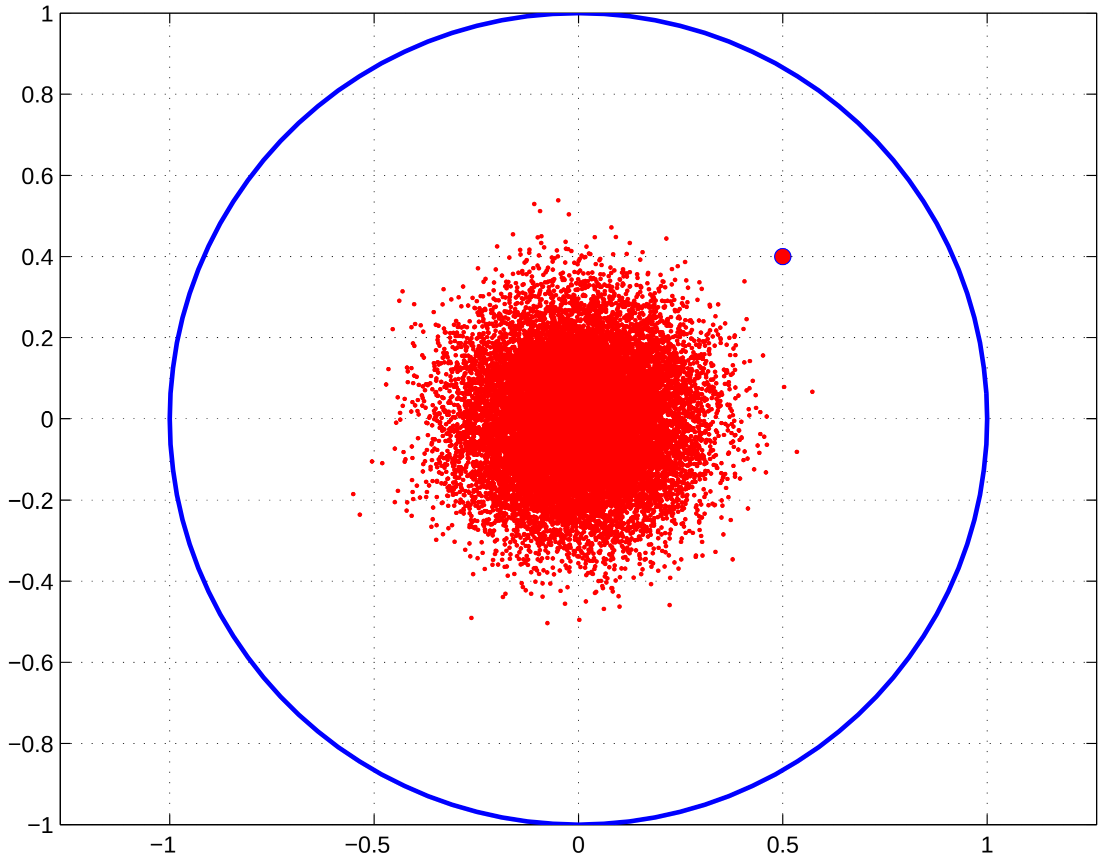
where $I(x; a, b)$ is the regularized incomplete beta function with parameters a and b.

For example for $n = 10$, $\delta = 0.05$, and $p = 0.95$, it gives $N_{\min} \approx 8.9 \cdot 10^6$. There are numerous approximate formulae and other estimates.

MC for multiobjective optimization

Objective function is $f : \mathbb{R}^n \to \mathbb{R}^m$, construct Pareto front.

Example: $m = 2$, $f(x)$ is linear, B is the unit ball, $n = 50, N = 100000$



MC for a box

$\min(c, x), x \in Q = [-1, 1]^n, c = (1, ..., 1)$

$$N_{\min} = \frac{\ln(1-p)}{\ln\left(1 - \frac{n^n \delta^n}{2^n n!}\right)} \,.$$

$n = 10$, $\delta = 0.1$ and $p = 0.95$, provide a huge $N_{\min} \approx 1.12 \cdot 10^{10}$.

Deterministic grids

Example $\max(c, x), x \in Q = [-1, 1]^n, c = (1, ..., 1)$

Uniform grid Uniform grid on Q, mesh points do not cover the boundary.

Sobol sequences (LP_τ sequences)

Results for $N = 10^6$

n; true max	2	3	4	5	10	15	20
Uniform grid	1.9960	2.9406	3.7576	4.4118	6.0000	7.5000	6.6667
LP_τ	1.9999	2.9792	3.8373	4.6844	7.9330	10.2542	10.9470
Monte Carlo	1.9974	2.9676	3.8731	4.6981	7.8473	10.0796	11.8560

Literature

B.Polyak, P.Shcherbakov "Why does Monte Carlo Fail to Work Properly in High-Dimensional Optimization Problems?" JOTA, 2017 (available on-line)