New random sampling: billiard walks

B. Polyak, E.Gryazina
Institute for Control Sciences, Moscow

January 8, 2013

Workshop "Optimization and Statistical Learning",
Les Houches

Outline

- Random sampling
- Hit-and-Run
- Billiard walks
- Examples
- Applications

Random sampling

Goal: generate points, uniformly distributed in a set $Q \subset R^n$.
Applications: integration over Q, volume and center of gravity
calculation, optimization (convex and nonconvex), control
(e.g. generate stabilizing controllers), robustness (generate
uncertainties) and many others.

Approaches: explicit algorithms for simple sets (boxes, balls,
simplices etc.); rejection method (take simple $S \supset Q$, generate
points x^i in S and reject x^i which are not in Q). However
these methods do not work for most Q. General technique —
random walks (Markov Chain Monte Carlo). Among them
Hit-and-Run techniques is the most popular.

Hit-and-Run

Random walk in Q. [Turchin(1971), Smith(1984)]
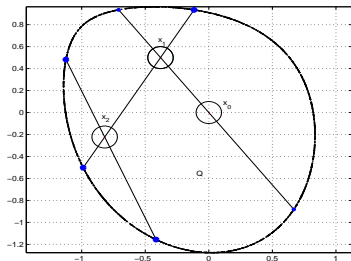
1. Choose initial point $x^0 \in Q$.

2. $d = s/||s||, s = randn(n, 1)$ — random direction 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 with x^1, go to Step 2.

Advantages

1. Distribution of x^i tends to uniform on Q.
2. Method is simple and works for nonconvex and nonconnected sets.
3. Boundary oracle is available for many descriptions of sets (linear inequalities, LMIs).

Drawbacks

- HR jams in corners.
- HR jams for narrow bodies.
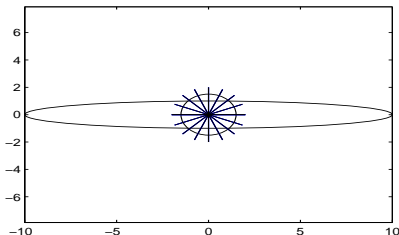
(Lovasz, Vempala. Hit-and-Run from a corner, 2007): Estimate of number of iterations to achieve uniformity with precision ε

$$N > 10^{10} \frac{n^2 R^2}{r^2} \ln \frac{M}{\varepsilon}$$

How to improve convergence?

Simple tools: transformations of the set, other distributions of directions d. However this medicine is not universal. We try to exploit another idea to improve HR.

Physical motivation

Our algorithm is motivated by physical phenomena of a gas diffusing in a vessel. A particle of gas moves with constant speed until it meets a boundary of the vessel, then it reflects (the angle of incidence equals the angle of reflection) and so on. When the particle hits another one, its direction and speed changes. In our simplified model we assume that direction changes randomly while speed remains the same. Thus our model combines ideas of Hit-and-Run technique with the use of billiard trajectories.

Billiards

There exist vast literature on mathematical theory of billiards:

- S. Tabachnikov, Geometry and Billiards, RI: Amer. Math. Soc., 1995.
- G. Galperin, A. Zemlyakov, Mathematical Billiards, Nauka, Moscow (in Russian), 1990.
- Y. G. Sinai, Dynamical systems with elastic reflections, Russian Mathematical Surveys 25 (2) (1970) 137–189.
- Y. G. Sinai, Billiard trajectories in a polyhedral angle, Russian Mathematical Surveys 33 (1) (1978) 219–220.

However billiard trajectories are deterministic. We introduce randomness in them.

New method - Billiard Walk

1. Choose starting point $x^0 \in \text{Int } Q$; $i = 0$, $x = x^0$.

2. Generate the length of the trajectory $\ell = -\tau \log \xi$, ξ is uniform random in $[0, 1]$, τ is a specified parameter.

3. Choose random direction $d \in \mathbb{R}^n$ uniform on the unit sphere.

4. Construct billiard trajectory of length ℓ with initial direction d. If it reaches a nonsmooth boundary point or the number of reflections is greater than $10n$, go to Step 3.

5. $i = i + 1$, the end point of the trajectory take as x^i and go to Step 2.

New method - Billiard Walk
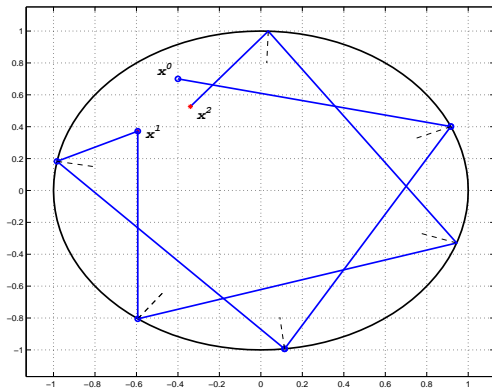
Asymptotical uniformity

Theorem

Suppose Q is connected, bounded and open (or a closure of such set) set, the boundary of Q is piecewise smooth. Then the distribution of points x^i sampled by BW algorithm tends to uniform on Q.

Hint of the proof.

- The algorithm is well defined.
- $p(y|x) > 0$ for all $x, y \in \text{Int} Q$.
- $p(y|x) = p(x|y)$

□

Implementation issues

① Choice of τ. There is trade-off between τ small and large. $\tau \approx \operatorname{diam} Q$

② Preliminary transformation. If Q has a barrier function $F(x)$ with $x^* \approx \operatorname{argmin} F$, then take

$$\left(\nabla^2 F(x^*)\right)^{-1/2} d.$$

③ Boundary oracle and normals. In most cases they are easily available, see examples below.

Boundary oracle and normals

For Q convex, boundary oracle is the segment $[-\underline{t}, \hat{t}]$,

$$\underline{t} = \max\{t : x^k - td \in Q\}, \hat{t} = \max\{t : x^k + td \in Q\}$$

If Q is a polytope

$$Q = \{x \in R^n : (a_i, x) \leq b_i, i = 1, \ldots, m\}$$

then

$$\underline{t} = \max_{i:(a_i,d)<0} \frac{(a_i, x^k) - b_i}{(a_i, d)}$$

$$\hat{t} = \max_{i:(a_i,d)>0} \frac{-(a_i, x^k) + b_i}{(a_i, d)}$$

while the normal to the boundary at the point $x^k + \hat{t}d$ equals a_i, where i is the index, for which maximum in above formulas is achieved.

Calculations for Q given by LMIs are also simple.

Comparison with HR

Each iteration of HR is less expensive than BW. However
number of iterations for BW is significantly smaller, as
demonstrated in examples below.

Behavior in the corner

- **Angle α at a plane.**
  Billiard trajectory: quits with probability 1 after no more
  than $N^* = \lceil \pi/\alpha \rceil$ reflections.
  HR: quits with probability $1 - (1 - \alpha/\pi)^k$ after k
  iterations
  (for large N^* HR quits with probability $1 - 1/e = 0.63$
  after N^* iterations).

- **Multidimentional case - polyhedral Q.**
  [Sinai (1967)] there exists M independent on initial point
  such that billiard trajectory quits Q after no more than M
  reflections.

- **Orthant $Q = \{x \in R^n : x \geq 0\}$.**
  Billiard trajectory: quits with probability 1 after no more
  than n reflections.
  HR: quits with probability $1 - (1 - 2^{-n})^n$ after
  approximately 2^{n-1} iterations (for large n with probability
  $1 - 1/e = 0.63$).

Concave corner $Q = \{x \in R^2 : ||x||_\infty \leq 1, ||x - a_i|| \geq 1\}$

Concave corners can be attractions for billiard trajectories.

a_i are vertices of $\{||x||_\infty \leq 1\}$. $N = 200$
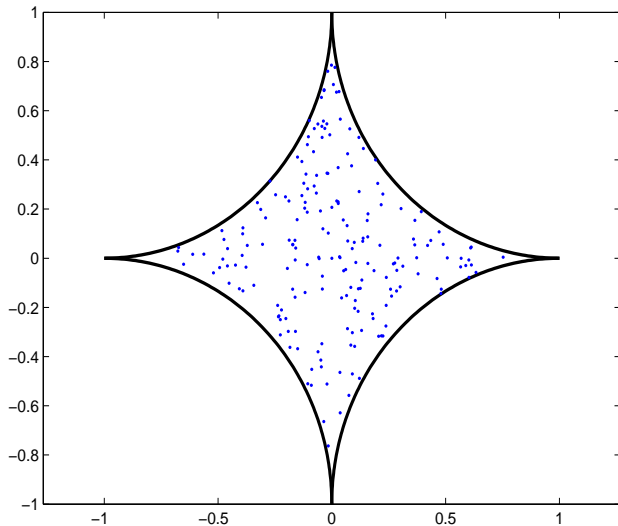
Curvilinear triangle

Curvature tends to 0 — more dangerous case.

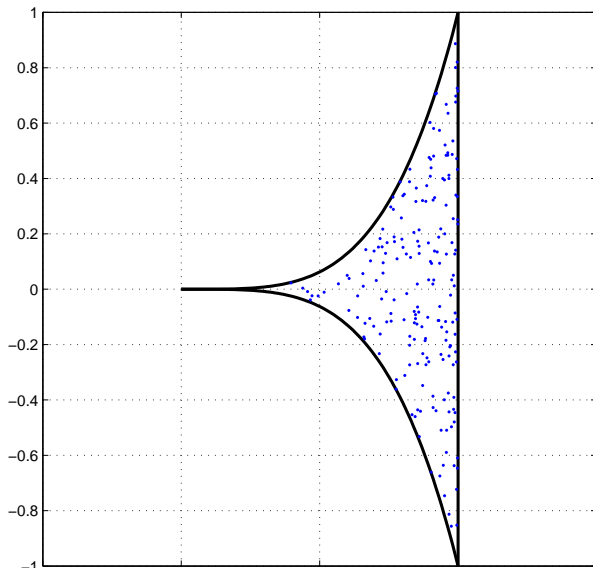$$Q = \{x \in \mathbb{R}^2 : x_1 < 1, -x_1^4 < x_2 < x_1^4\}$$

The number of reflections increases dramatically as the first coordinate of x^0 tends to zero and even for $x_1^0 = 10^{-4}$ the trajectory becomes unreliazable.

$$x_2^0 = 0.9, \quad \ell = 1 \quad d = [-1; 0]$$

x_1^0	Number of reflections
1e-3	746
5e-4	1851
4e-4	2480
3e-4	3617
2e-4	6158
1.1e-4	13496
1.01e-4	>5e+6

Curvilinear triangle

$$Q = \{x \in \mathbb{R}^2 : x_1 < 1, -x_1^4 < x_2 < x_1^4\}, \ N = 500$$

Cube $Q = \{x \in \mathbb{R}^n : 0 \leq x_i \leq 1\}$
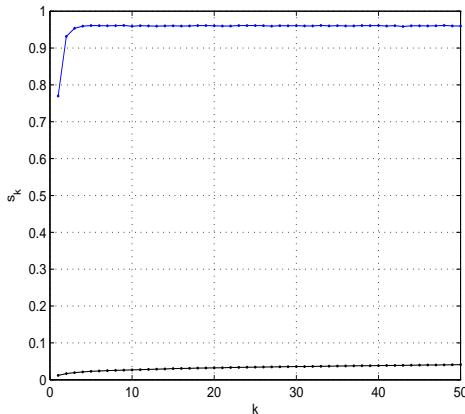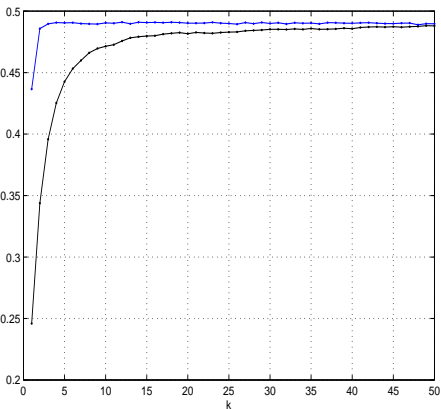
The next point of the BW algorithm is derived explicitly!

Current point x, direction d, length of the trajectory ℓ.
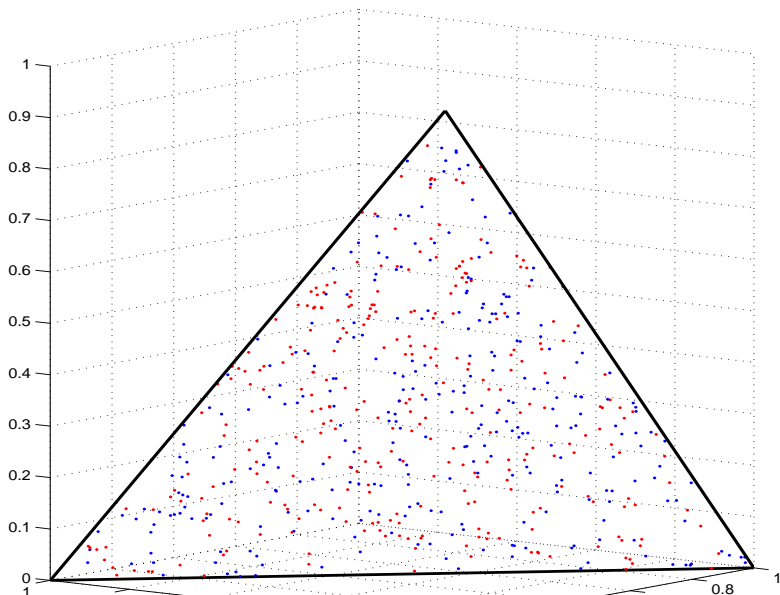Calculate $k_i = \lfloor x_i + \ell d_i \rfloor$ and go to y:

$$y_i = \begin{cases} x_i + \ell d_i - k_i, & k_i \text{ is even} \\ 1 - (x_i + \ell d_i - k_i), & k_i \text{ is odd} \end{cases}, \quad i = 1, \ldots, n.$$

Serial correlation: BW vs HR

we compare $\mathbb{E}||x^k - x^0||_\infty$ for $n = 50$ averaged over 500 runs for two initial points $x^0 = [1/2, \ldots, 1/2]^T$ (left) and $x^0 = [1/n, \ldots, 1/n]^T$ (right). Implementing BW (blue) we take $\tau = \sqrt{n}$, HR (black).

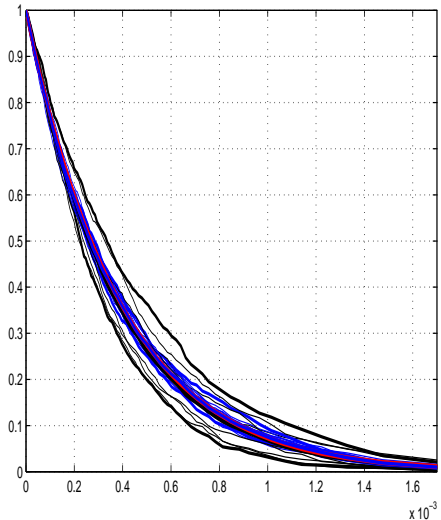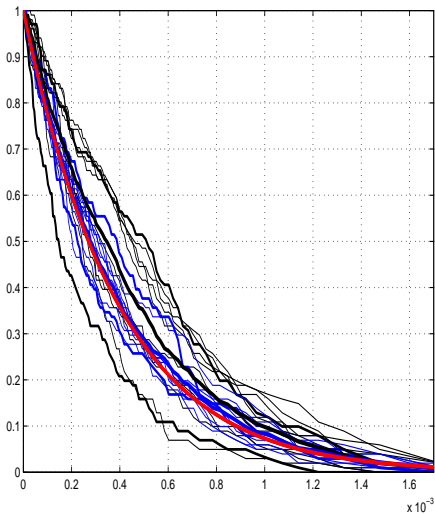# Simplex $S_n = \{x_i \geq 0, \sum x_i = 1, i = 0, 1, \ldots, n\}$

Simplex: uniformity estimation

$$S_\alpha = \{x \in \mathbb{R}^{n+1} : x_i \geq \alpha, \sum x_i = 1\}, \quad 0 \leq \alpha \leq \frac{1}{n+1}$$
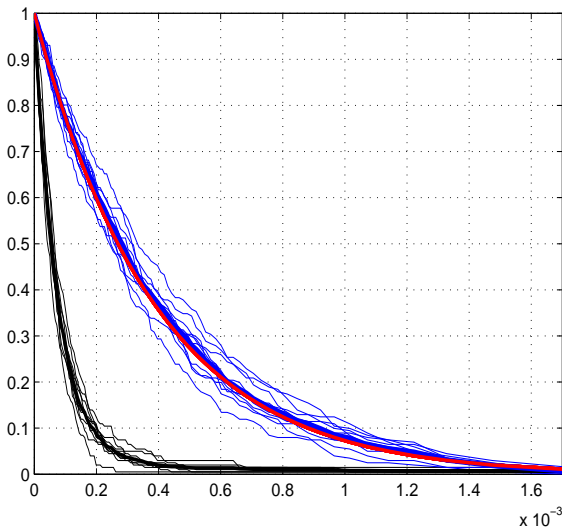
$$f(\alpha) = \frac{\text{vol}S_\alpha}{\text{vol}S_0} = (1 - (n+1)\alpha)^n.$$
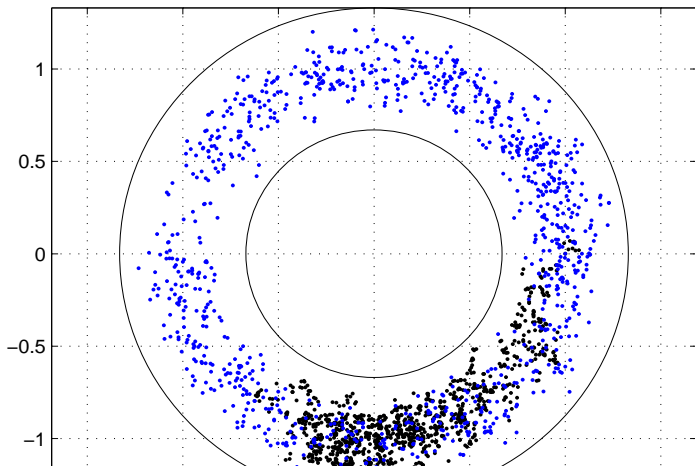
$n = 50$, 100 and 1000 samples

Red - uniform random, black - HR, blue - BW.

Simplex: $x^0 = \left[0.9, \frac{0.1}{n}, \ldots, \frac{0.1}{n}\right]^T$
$n = 50, \ N = 200$

Toroid $Q = \{x \in \mathbb{R}^n : ||x - c_x|| \leq \frac{1}{3}\}$

$c_{xi} = \frac{x_i}{\sqrt{x_1^2 + x_2^2}}$, $i = 1, 2$, $c_{xi} = 0$, $i > 2$ is a projection to the circumference $x_1^2 + x_2^2 = 1$, $x_3 = \cdots = x_n = 0$. $n = 10$, $N = 10^3$.

Applications — optimization

1. Convex optimization.
2. Concave optimization.
3. Global optimization

Convex optimization

Approximation of center of gravity method
Cutting plane methods for SDP
However it is hard to compete with modern deterministic
methods for convex optimization.

Concave optimization

$$\min f(x), x \in Q$$

$f(x)$ concave, Q is a polytope. Random $x^0 \in Q$ and conditional gradient method starting at x^0. That is we solve several LP at each iteration. Branch and Bound ideas can be exploited.

Global optimization

Multi-start methods with random initial points. For unconstrained minimization we can generate random points in level sets. First simulations look promising.

Conclusions

Billiard walk algorithm seems to be more effective if compared
with Hit-and-Run. However future work is needed.