March 9, 2018

Abstract

We would like to find point sets with low discrepancy, and also of known discrepancy. There are some algorithms to compute the discrepancy of point sets and a number of candidate point sets work with. For the purposes of applications in low dimensions, we can compare sets of several thousand points using the most naive algorithms in a reasonable amount of time.

We will describe some algorithms for explicitly computing discrepancy, analyze some problems in optimizing the quality point sets, and look at some experimental data related to conjectured “good” point sets.

Based on various work with Peter Grabner, Johann Brauchart, Alden Walker

- A simple form of *discrepancy* is a function which measures the irregularity of the distribution of a truncated sequence $\{x_i\}_{i=1}^{\infty}$

$$D_N(\{x_i\}_{i=1}^{\infty}) := \sup_{0 \le a < b \le 1} \left| \frac{\#(\{x_i\}_{i \le N} \cap [a,b])}{N} - (b-a) \right| .$$

- This notion generalizes in many ways to other spaces. Identify the endpoints of the interval and consider the supremum over all connected subsets gives a function of configurations in $\mathbb{S}^1$, the *spherical cap discrepancy.*

Discrepancy

We would like to have low discrepancy configurations to replace uniformly distributed random points.

For reasonable $f : \mathbb{I} \to \mathbb{R}$, the error in the approximate integral

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i) + err = \int_0^1 f(x)dx$$

is bounded by the discrepancy up to a factor of the total variation of f, a version of the Koksma-Hlawka inequality.

On a unit sphere $\mathbb{S}^d$ in $\mathbb{R}^{d+1}$ with uniform measure σ and given a spherical cap C in $\mathbb{S}^d$, the *local spherical cap discrepancy* of a set X_N of N distinct points is given by

$$D_C[X_n] := \left| \frac{1}{n} \#(X_n \cap C) - \sigma(C) \right|,$$

This is the normalized difference between the expected and the actual number of points found in cap C

Remark

By integrating over caps of a fixed radius $\arccos(t)$*, we can define*

$$\mathrm{Var}_{C(t)}[X_n] := \int_{\mathbb{S}^d} D^2_{C(t)}[X_n] \, d\sigma.$$

There are several natural measures of the quality of X_n including:

- The classical **spherical cap discrepancy** given as

$$D[X_n] := \sup_{C \subset \mathbb{S}^d} D_C[X_n].$$

- The L^2**-discrepancy**, given as

$$D_{L^2}[X_n] := \sqrt{\int_{-1}^{1} \mathrm{Var}_{C(t)}[X_n] \, \mathrm{d}t}.$$

Spherical Cap Discrepancy

- It is know there exist good point sets X_N on the sphere satisfying $D(X_N)$ bounded by $O(N^{-3/4}\sqrt{\log N})$ and a lower bound for all points sets of $\Omega(N^{-3/4})$. Constructions of reasonable point sets are probabilistic; we can check samples.

Spherical Cap Discrepancy

det5000.pdf

5000 determinantal points

Spherical Cap Discrepancy

detgraph.png

- It is know there exist good point sets X_N on the sphere satisfying $D(X_N)$ bounded by $O(N^{-3/4}\sqrt{\log N})$ and a lower bound for all points sets of $\Omega(N^{-3/4})$. Constructions are probabilistic; we can check samples.
- Uniform random points are bounded by $O(N^{-1/2})$ (*a.s.*).

Spherical Cap Discrepancy

uniform5000.pdf

5000 psudo-random points

Spherical Cap Discrepancy

uniformgraph.png

$\log(N)$-$\log(D)$ plot for uniform random points

Spherical Cap Discrepancy

- It is know there exist good point sets X_N on the sphere satisfying $D[X_N]$ bounded by $O(N^{-3/4}\sqrt{\log N})$ and a lower bound for all points sets of $\Omega(N^{-3/4})$. Constructions are probabilistic; we can check samples.
- Uniform random points are bounded by $O(N^{-1/2})$ (*a.s.*).
- It is conjectured that the Fibonacci points on the sphere have low discrepancy, perhaps optimal, but they are only known to be as good as random points.

Spherical Cap Discrepancy

fib5000.pdf

5000 Fibonacci points

fibgraph.png

$\log(N)$ $\log(D)$ plot for Fibonacci points

Some other "good" point sets.

- Lubotzky-Phillips-Sarnak points generated by irrational rotations.
- Spiral points other than the Fibonacci points.

Spherical Cap Discrepancy

lps5000.pdf

5000 Lubotzky-Phillips-Sarnak points

Spherical Cap Discrepancy

lpsgraph.png

Spherical Cap Discrepancy

e5000.pdf

5000 spiral points

egraph.png

det5000.pdf

uniform5000.pdf

fib5000.pdf

lps5000.pdf

e5000.pdf

Spherical Cap Discrepancy

pointplot.png

Computation

- Computing the L^2-discrepancy of a set of points can be done via **Stolarsky's Invariance Principle**, which states

$$\frac{1}{n^2}\sum_{i\neq j}\|x_i - x_j\| + K_d\, D_{L^2}^2[X_n] = \iint \|x-y\|\,\mathrm{d}\sigma(x)\mathrm{d}\sigma(y),$$

  where K_d is a known constant depending only on the dimension d.
- Based on results for discrepancy in boxes, it is likely that computing the spherical cap discrepancy is NP-hard.

Remark

There is still a nice algorithm for computing the discrepancy.

Computation

In the case of the star discrepancy, an algorithm was described by Niederreiter that exactly computes the star discrepancy.

- Note that the discrepancy function achieves local extrema when the measuring sets are "captured."
- This is a finite set, so enumerate and take the maximum.

A similar approach works for spherical caps. It is a brute force approach that is exponential in dimension, but it is polynomial in the number of points. To "capture" a cap on $\mathbb{S}^d$ requires $d+1$ points or fewer, and $\binom{N}{d+1}$ is $O(N^{d+1})$.

Comparing points gives an extra factor of N, so the runtime is $O(N^{d+2})$. There are possible improvements with clever sorting, at some cost in memory.

Computation

twopoints.pdf

three.pdf

Computation

- This algorithm illustrates some of the issues with large constants in polynomial time algorithms, and with the implementation. Just optimizing the code for the brute force algorithm can the time for several thousand points from several hundred years to a few hours.
- Ways to increase speed: Sort with respect to a known good set using the triangle inequality. Use a sweep pivoting on d points. These both cost memory.
- It is also a massively parallel problem. In principle, we can compute discrepancy for millions of points relatively quickly.

Optimization

This algorithm hints at some optimization methods, but it really seems to depends on the labeled partitioning of N. For a simplex, it works out nicely.

Four points on $\mathbb{S}^2$ can almost be optimized by hand. For a lower bound, consider that discrepancy is a max-type function and symmetric across a cap C, which allows one to pass supporting points across the boundary of a cap to find stationary points.

This gives an easy system to solve, where the important part is:

$$|\mathrm{Vol}(C) - 1/4| = |1 - \mathrm{Vol}(C)| \implies \mathrm{Vol}(C) = 5/8,\ D[X_N] > 3/8.$$

Remark

This does not hold for the regular simplex.

fourpoint2.pdf

A very simple family of four points.

We can symbolically solve to give 3-point discrepancies of

$$\frac{512 + 19\sqrt{466 - 38\sqrt{105}} + \sqrt{210\left(233 - 19\sqrt{105}\right)}}{2048}$$

$$\frac{1024 - 19\sqrt{466 - 38\sqrt{105}} - \sqrt{210\left(233 - 19\sqrt{105}\right)}}{2048}$$

(which also happen to be 3/8!)

Optimization

pointplot.pdf

3-point and 2-point discrepancies for the tetrahedral family

Optimization

fourpoint.pdf

A minimal discrepancy set