A WORK-IN-PROGRESS CHAT

Wöden Kusner[1]
Technische Universität Graz

unilogo2.png

January 2015

Abstract

Right now, one of my projects is focused on exploring the quality of measure on spheres. We'll look at the implementation of an algorithm for computing discrepancy, some of the analysis of a problem related to the quality of a distribution of points in the compact setting, and some of the open problems that go with them.

Discrepancy

You are probably familiar with discrepancy as a function $D : \mathbb{N} \times \mathbb{I}^{\mathbb{N}} \to \mathbb{I}$ measures the irregularity of the distribution of a sequence $\{x_i\}$ when truncated at a natural number

$$D_N(\{x_i\}) = \sup_{0 \leq a < b \leq 1} |\frac{\#(\{x_i\}_{i \leq N)} \cap [a,b]}{N} - (b-a)|.$$

This notion can be generalized in many ways to other spaces, for example, by identifying the end points of the interval and including all connected subsets containing the point $\{0 = 1\}$ gives a discrepancy for a sequence of points on $\mathbb{S}^1$, the spherical cap discrepancy.

Discrepancy

In general, given an geometric sphere $\mathbb{S}^d$ with radius 1 and normalized uniform measure σ and an (open) spherical cap C embedded in $\mathbb{R}^{d+1}$, the *local spherical cap discrepancy* of a set X_N of N distinct points in the d-sphere is given by

$$\mathrm{D}_C[X_N] := |[\mathrm{Vol}(C) - \frac{1}{N}\#|X_N \cap C|]|$$

which may be viewed as a normalized difference between the expected number of points in a cap of Vol(C) and the number of points found in cap C

$$\mathrm{D}_C[X_N] = \frac{1}{N}(\mathbb{E}[\#|X_N \cap C|] - \#|X_N \cap C|).$$

Integrating over the space of caps of fixed size (equivalently, integrated over the sphere), we define a variance

$$\mathrm{Var}_C[X_N] = \int_{\mathbb{S}^d} \mathrm{D}_C^2 \, \mathrm{d}\sigma.$$

Discrepancy

There are several measurements of the quality of the point set based on this method. One is the *spherical cap discrepancy*, given as

$$\mathrm{D}(X_N) := \sup_{C \in \mathbb{S}^d \times [-1,1]} \mathrm{D}_C[X_N].$$

Another is the $\mathbb{L}^2$-*discrepancy*, given by

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

where $C(t)$ is a cap defined by it's center z and all points with inner product greater than t.

Remark

the local discrepancy is symmetric through the boundary of the cap, so these integrals double count in some sense.

Discrepancy

fiblonglat.jpg

Discrepancy

- Computing the $\mathbb{L}^2$-discrepancy of a set of points is fairly easy, it follows from the *Stolarsky Invariance Principle*, which roughly states

$$\frac{1}{N^2}\sum_{i\neq j}|x_i - x_j| + \mathrm{D}_{\mathbb{L}^2}[X_N] = C_d.$$

- Based on results for star-discrepancy, it is likely that computing the spherical cap discrepancy is NP-hard. However, there is still a nice algorithm for computing the cap discrepancy.

Discrepancy

stollarsky sketch intersection measures

Discrepancy

stollarsky sketch of reproducing kernels

Computation

star discrepancy for points in a box.

Computation

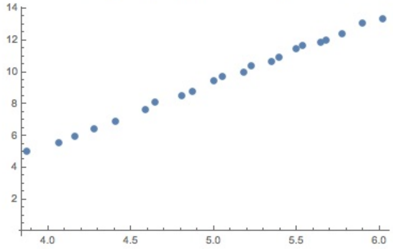
In the case of the star discrepancy an algorithm was described by Niederreiter that exactly computes the star discrepancy.

- 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 but it is polynomial in the number of points (so is the original algorithm for star discrepancy)! Why? Because to "capture" a cap on $\mathbb{S}^d$ requires $d+1$ points or fewer, and $\binom{N}{d+1} = O(N^{d+1})$. Then comparing points gives an extra factor of N, so the runtime should be $O(N^{d+2})$, with possible improvements from sorting, etc.

Computation

```
ListPlot[Drop[Log[{Map[PottsTest3, Take[files, 32]], PrePots[[All, 3]]}] // Transpose, 10]]
```



```
Fit[Drop[Log[{Map[PottsTest3, Take[files, 32]], PrePots[[All, 3]]}] // Transpose, 10], {1, x}, x]
-10.5834 + 3.99418 x
```

Computation

This algorithm illustrates the issue of large constants in polynomial time algorithms, as using the loops, this would probably take several hundred years for several thousand points. HOWEVER, it is a massively parallel problem. Using matrix multiplication reduces the runtime by at least an order of 100 (strictly from parallelization, since these are generally not sparse matrices and I don’t think there is any good fast matrix multiplication implemented in my code.) Memory then becomes a problem.

Remark

Implement and optimize this code in as a open parallel program, with good numerical error tracking. Build a database of discrepancies.

Computation

This algorithm also allows optimization to find minimal discrepancy point sets, but it is horrible since it depends on the partitions of N. So for a simplex, it is nice. Four points on $\mathbb{S}^2$ can be optimized by hand (almost). Under various assumptions, the minimum configuration and associated cap C allows one to pass support points across the boundary by symmetry. Then at the minimum discrepancy configuration

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

This is attained at configurations that do not correspond to the regular simplex. Then the 2-point discrepancy comes into play.

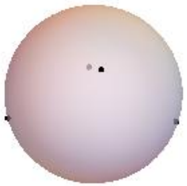
Computation

Numerical minimization hinted at $.375 = 3/8$ and can be symbolically solved to give discrepancies of

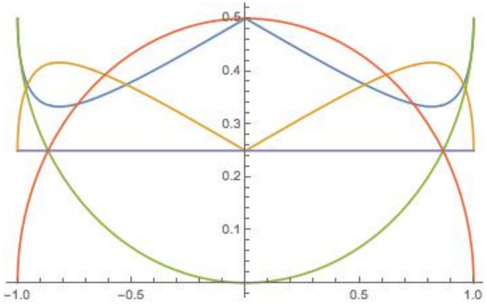
$$\frac{512 + 19\sqrt{466 - 38\sqrt{105}} + \sqrt{210\left(233 - 19\sqrt{105}\right)}}{2048} = \frac{3}{8}$$

$$\frac{1024 - 19\sqrt{466 - 38\sqrt{105}} - \sqrt{210\left(233 - 19\sqrt{105}\right)}}{2048} = \frac{3}{8}$$

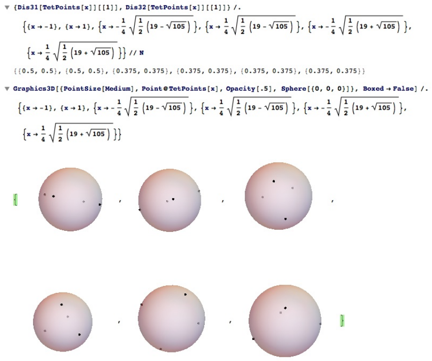
Computation



Computation



Discrepancy



```
{Dis31[TetPoints[x]][[1]], Dis32[TetPoints[x]][[1]]} /.
  {{x → -1}, {x → 1}, {x → -1/4 √(1/2 (19 - √105))}, {x → 1/4 √(1/2 (19 - √105))}, {x → -1/4 √(1/2 (19 + √105))},
   {x → 1/4 √(1/2 (19 + √105))}} // N
{{0.5, 0.5}, {0.5, 0.5}, {0.375, 0.375}, {0.375, 0.375}, {0.375, 0.375}, {0.375, 0.375}}

Graphics3D[{PointSize[Medium], Point@TetPoints[x], Opacity[.5], Sphere[{0, 0, 0}]}, Boxed → False] /.
  {{x → -1}, {x → 1}, {x → -1/4 √(1/2 (19 - √105))}, {x → 1/4 √(1/2 (19 - √105))}, {x → -1/4 √(1/2 (19 + √105))},
   {x → 1/4 √(1/2 (19 + √105))}}
```

Variance

One last remark is that the variance

$$\mathrm{Var}_C[X_N] = \int_{\mathbb{S}^d} \mathsf{D}_C^2 \, \mathrm{d}\sigma.$$

is a quantity that we are actively investigating but is perhaps currently both fuzzy and technical. There are some nice ways to look at it as a quality of measures parametrized by the radius of the cap size and the function spaces. It also seems to fit well with a statistical mechanical property of point processes (the hyperuniformity) and their behavior in the thermodynamic limit.

Remark

Understand these quantities from multiple perspectives (integral geometry, measure theory, optimal transport, statistical mechanics, discrete/combinatorial geometry).