Practical volume estimation by a new annealing schedule for cooling convex bodies

Vissarion Fisikopoulos

Joint work with A.Chalkis, I.Z.Emiris

Dept. of Informatics & Telecommunications, University of Athens

ULB, 2019

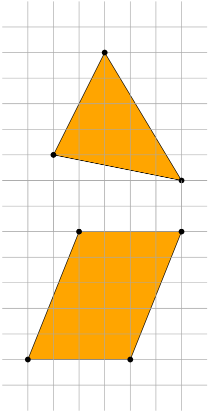
Our problem

Given P a convex polytope in $\mathbb{R}^d$ compute the volume of P.

Our problem

Given P a convex polytope in $\mathbb{R}^d$ compute the volume of P.

Some elementary polytopes (simplex, cube) have simple determinantal formulas.



$$\begin{vmatrix} 1 & 2 & 1 \\ 3 & 6 & 1 \\ 6 & 1 & 1 \end{vmatrix} / 2! = 11$$

$$\begin{vmatrix} 2 & 5 \\ 4 & 0 \end{vmatrix} = 20$$

Birkhoff polytopes

- Given the complete bipartite graph $K_{n,n} = (V, E)$ a perfect matching is $M \subseteq E$ s.t. every vertex meets exactly one member of M

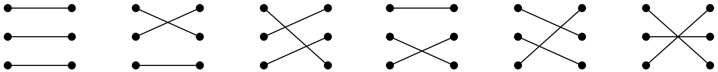
Birkhoff polytopes

- Given the complete bipartite graph $K_{n,n} = (V, E)$ a perfect matching is $M \subseteq E$ s.t. every vertex meets exactly one member of M
- $S \subseteq E,\ \chi_e^S = \{1 \text{ if } e \in S, 0 \text{ otherwise}\}$

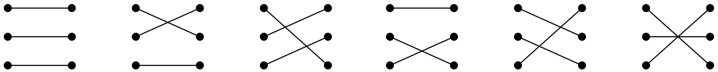
Birkhoff polytopes

- Given the complete bipartite graph $K_{n,n} = (V, E)$ a perfect matching is $M \subseteq E$ s.t. every vertex meets exactly one member of M
- $S \subseteq E,\ \chi_e^S = \{1 \text{ if } e \in S, 0 \text{ otherwise}\}$
- $B_n = conv\{\chi^M \mid M \text{ is a perfect matching of } K_{n,n}\}$

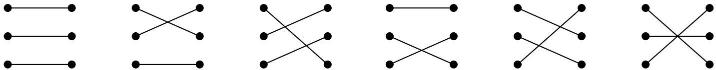
Birkhoff polytopes

- Given the complete bipartite graph $K_{n,n} = (V, E)$ a perfect matching is $M \subseteq E$ s.t. every vertex meets exactly one member of M
- $S \subseteq E$, $\chi_e^S = \{1 \text{ if } e \in S, 0 \text{ otherwise}\}$
- $B_n = conv\{\chi^M \mid M \text{ is a perfect matching of } K_{n,n}\}$
- 

Birkhoff polytopes

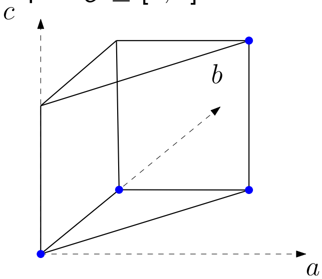
- Given the complete bipartite graph $K_{n,n} = (V, E)$ a perfect matching is $M \subseteq E$ s.t. every vertex meets exactly one member of M
- $S \subseteq E,\ \chi_e^S = \{1 \text{ if } e \in S, 0 \text{ otherwise}\}$
- $B_n = conv\{\chi^M \mid M \text{ is a perfect matching of } K_{n,n}\}$
- 
- # faces of B_3: 6, 15, 18, 9; $\mathrm{vol}(B_3) = 9/8$

Birkhoff polytopes

- Given the complete bipartite graph $K_{n,n} = (V, E)$ a perfect matching is $M \subseteq E$ s.t. every vertex meets exactly one member of M
- $S \subseteq E$, $\chi_e^S = \{1 \text{ if } e \in S, 0 \text{ otherwise}\}$
- $B_n = conv\{\chi^M \mid M \text{ is a perfect matching of } K_{n,n}\}$
- 
- $\#$ faces of B_3: 6, 15, 18, 9; $\mathrm{vol}(B_3) = 9/8$
- there exist formulas for the volume [deLoera et al '07] but values only known for $n \leq 10$ after 1yr of parallel computing [Beck et al '03]

Volumes and counting

- Given n elements & partial order; order polytope $P_O \subseteq [0,1]^n$ coordinates of points satisfies the partial order



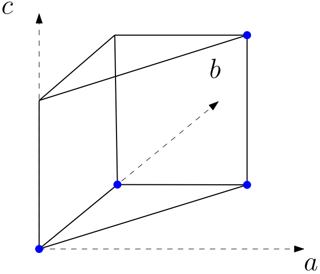
a, b, c

partial order: $a < b$

3 linear extensions: abc, acb, cab

Volumes and counting

- Given n elements & partial order; order polytope $P_O \subseteq [0,1]^n$ coordinates of points satisfies the partial order



a, b, c

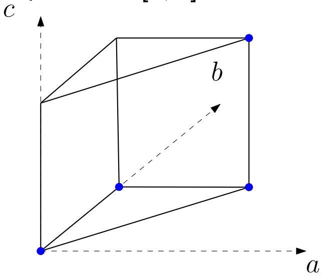
partial order: $a < b$

3 linear extensions: abc, acb, cab

- # linear extensions = volume of order polytope $\cdot\ n!$ [Stanley'86]

Volumes and counting

- Given n elements & partial order; order polytope $P_O \subseteq [0,1]^n$ coordinates of points satisfies the partial order



a, b, c

partial order: $a < b$

3 linear extensions: abc, acb, cab

- # linear extensions $=$ volume of order polytope $\cdot$ n! [Stanley'86]

- Counting linear extensions is #P-hard [Brightwell'91]

Mixed volume

Let $P_1, P_2, \ldots, P_d$ be polytopes in $\mathbb{R}^d$ then the mixed volume is

$$M(P_1, \ldots, P_d) = \sum_{I \subseteq \{1,2,\ldots,d\}} (-1)^{(d-|I|)} \cdot \mathrm{Vol}(\sum_{i \in I} P_i)$$

where the sum is the Minkowski sum.

Mixed volume

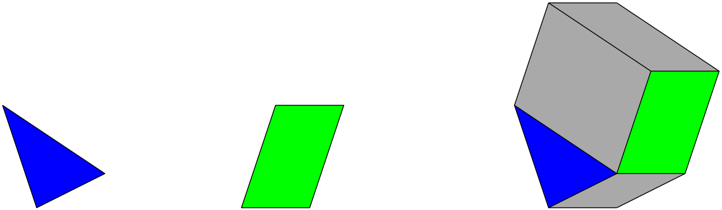
Let $P_1, P_2, \ldots, P_d$ be polytopes in $\mathbb{R}^d$ then the mixed volume is

$$M(P_1, \ldots, P_d) = \sum_{I \subseteq \{1,2,\ldots,d\}} (-1)^{(d-|I|)} \cdot \mathrm{Vol}(\sum_{i \in I} P_i)$$

where the sum is the Minkowski sum.

Example

For $d = 2$: $M(P_1, P_2) = \mathrm{Vol}(P_1 + P_2) - \mathrm{Vol}(P_1) - \mathrm{Vol}(P_2)$



P_1 $\quad$ P_2 $\quad$ $P_1 + P_2$

Volumes and algebraic geometry

Toric geometry

If P is an integral d-dimensional polytope, then $d!$ times the volume of P is the degree of the toric variety associated to P.

BKK bound (Bernshtein, Khovanskii, Kushnirenko)

Let $f_1, \ldots, f_n$ be polynomials in $\mathbb{C}[x_1, \ldots, x_n]$. Let $N(f_j)$ denote the Newton polytope of f_j, i.e. the convex hull of it's exponent vectors. If $f_1, \ldots, f_n$ are generic, then the number of solutions of the polynomial system of equations

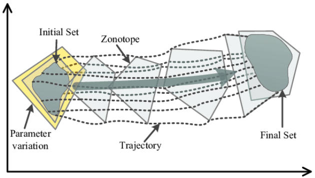
$$f_1 = 0, \ldots, f_n = 0$$

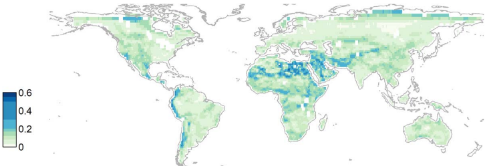
with no $x_i = 0$ is equal to the normalized mixed volume

$$n! M(N(f_1), \ldots, N(f_n))$$

Applications

- Volume of zonotopes is used to test methods for order reduction which is important in several areas: autonomous driving, human-robot collaboration and smart grids [Althoff et al.]
- Volumes of intersections of polytopes are used in bio-geography to compute biodiversity and related measures e.g. [Barnagaud, Kissling, Tsirogiannis, Fisikopoulos, Villeger, Sekercioglu'17]



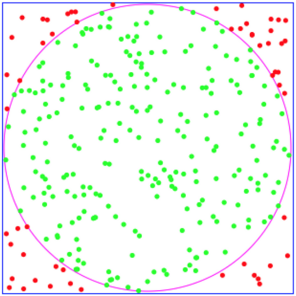


First thoughts for volume computation

- Triangulation (or sign decomposition) methods – exponential size in d

First thoughts for volume computation

- Triangulation (or sign decomposition) methods – exponential size in d
- Sampling/rejections techniques (sample from bounding box) fail in high dimensions



volume(unit cube) $= 1$
volume(unit ball) $\sim (c/d)^{d/2}$
–*drops exponentially with d*

Our setting

Given P a convex polytope in $\mathbb{R}^d$ compute the volume of P.

H-polytope : $P = \{x \mid Ax \leq b,\ A \in \mathbb{R}^{q \times d},\ b \in \mathbb{R}^q\}$

V-polytope : P is the convex hull of a set of points in $\mathbb{R}^d$

zonotope : Minkowski sum of segments (eq. linear projections of d-cubes)

Our setting

Given P a convex polytope in $\mathbb{R}^d$ compute the volume of P.

H-polytope : $P = \{x \mid Ax \leq b,\ A \in \mathbb{R}^{q \times d},\ b \in \mathbb{R}^q\}$

V-polytope : P is the convex hull of a set of points in $\mathbb{R}^d$

zonotope : Minkowski sum of segments (eq. linear projections of d-cubes)

Complexity

- #P-hard for both representations [DyerFrieze'88]

Our setting

Given P a convex polytope in $\mathbb{R}^d$ compute the volume of P.

- H-polytope : $P = \{x \mid Ax \leq b,\ A \in \mathbb{R}^{q \times d},\ b \in \mathbb{R}^q\}$
- V-polytope : P is the convex hull of a set of points in $\mathbb{R}^d$
- zonotope : Minkowski sum of segments (eq. linear projections of d-cubes)

Complexity

- #P-hard for both representations [DyerFrieze'88]
- open if both representations available

Our setting

Given P a convex polytope in $\mathbb{R}^d$ compute the volume of P.

H-polytope : $P = \{x \mid Ax \leq b,\ A \in \mathbb{R}^{q \times d},\ b \in \mathbb{R}^q\}$

V-polytope : P is the convex hull of a set of points in $\mathbb{R}^d$

zonotope : Minkowski sum of segments (eq. linear projections of d-cubes)

Complexity

- #P-hard for both representations [DyerFrieze'88]
- open if both representations available
- no deterministic poly-time algorithm can compute the volume with less than exponential relative error (oracle model) [Elekes'86]

State-of-the-art

Authors-Year	Complexity (oracle steps)	Algorithm
[Dyer, Frieze, Kannan'91]	$O^*(d^{23})$	Sequence of balls + grid walk
[Kannan, Lovasz, Simonovits'97]	$O^*(d^5)$	Sequence of balls + ball walk + isotropy
[Lovasz, Vempala'03]	$O^*(d^4)$	Annealing + hit-and-run
[Cousins, Vempala'15]	$O^*(d^3)$	Gaussian cooling (* well-rounded)
[Lee, Vempala'18]	$O^*(Fd^{\frac{2}{3}})$	Hamiltonian walk (** H-polytopes)

State-of-the-art

Authors-Year	Complexity (oracle steps)	Algorithm
[Dyer, Frieze, Kannan'91]	$O^*(d^{23})$	Sequence of balls + grid walk
[Kannan, Lovasz, Simonovits'97]	$O^*(d^5)$	Sequence of balls + ball walk + isotropy
[Lovasz, Vempala'03]	$O^*(d^4)$	Annealing + hit-and-run
[Cousins, Vempala'15]	$O^*(d^3)$	Gaussian cooling (* well-rounded)
[Lee, Vempala'18]	$O^*(Fd^{\frac{2}{3}})$	Hamiltonian walk (** H-polytopes)

Software:

- [Emiris, F'14] Sequence of balls + coordinate hit-and-run
- [Cousins, Vempala'16] Gaussian cooling + hit-and-run

State-of-the-art

Authors-Year	Complexity (oracle steps)	Algorithm
[Dyer, Frieze, Kannan'91]	$O^*(d^{23})$	Sequence of balls + grid walk
[Kannan, Lovasz, Simonovits'97]	$O^*(d^5)$	Sequence of balls + ball walk + isotropy
[Lovasz, Vempala'03]	$O^*(d^4)$	Annealing + hit-and-run
[Cousins, Vempala'15]	$O^*(d^3)$	Gaussian cooling (* well-rounded)
[Lee, Vempala'18]	$O^*(Fd^{\frac{2}{3}})$	Hamiltonian walk (** H-polytopes)

Software:

- [Emiris, F'14] Sequence of balls + coordinate hit-and-run
- [Cousins, Vempala'16] Gaussian cooling + hit-and-run

Limitation: Efficient only for H-polytopes (scale up-to few hundred dims in hrs), for V-polytopes or zonotopes:

- oracles are expensive (solution of an LP)
- [Cousins, Vempala'16] needs a bound on the number of facets

State-of-the-art

Authors-Year	Complexity (oracle steps)	Algorithm
[Dyer, Frieze, Kannan'91]	$O^*(d^{23})$	Sequence of balls + grid walk
[Kannan, Lovasz, Simonovits'97]	$O^*(d^5)$	Sequence of balls + ball walk + isotropy
[Lovasz, Vempala'03]	$O^*(d^4)$	Annealing + hit-and-run
[Cousins, Vempala'15]	$O^*(d^3)$	Gaussian cooling (* well-rounded)
[Lee, Vempala'18]	$O^*(Fd^{\frac{2}{3}})$	Hamiltonian walk (** H-polytopes)

Software:

- [Emiris, F'14] Sequence of balls + coordinate hit-and-run
- [Cousins, Vempala'16] Gaussian cooling + hit-and-run

Limitation: Efficient only for H-polytopes (scale up-to few hundred dims in hrs), for V-polytopes or zonotopes:

- oracles are expensive (solution of an LP)
- [Cousins, Vempala'16] needs a bound on the number of facets

Goal: Efficient algorithm for V-polytopes and zonotopes

Methodology

Volume algorithms parts

1. **M**ultiphase **M**onte **C**arlo: e.g. Sequence of balls, Annealing of functions
2. Sampling: grid-walk, ball-walk, hit-and-run, Hamiltonian walk

- Typically MMC (1) at each phase solves a sampling problem (2)

Methodology

Volume algorithms parts

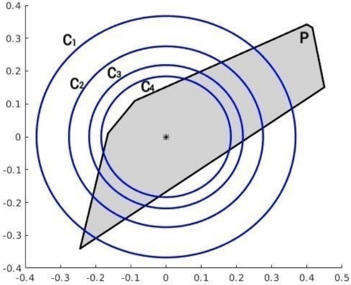
1. **M**ultiphase **M**onte **C**arlo: e.g. Sequence of balls, Annealing of functions
2. Sampling: grid-walk, ball-walk, hit-and-run, Hamiltonian walk

- Typically MMC (1) at each phase solves a sampling problem (2)

Our approach:

- MMC using convex bodies (balls or H-polytopes that "fit well" and are "easy" to sample from)
- A new annealing method to construct the sequence of bodies
- A new convergence criterion

Multiphase Monte Carlo



Construct a sequence of convex bodies $C_1 \supseteq \cdots \supseteq C_m$ intersecting the given polytope P, then:

$$\mathrm{vol}(P) = \frac{\dfrac{\mathrm{vol}(P_m)}{\mathrm{vol}(C_m)}}{\dfrac{\mathrm{vol}(P_i)}{\mathrm{vol}(P_0)}\dfrac{\mathrm{vol}(P_{i+1})}{\mathrm{vol}(P_i)}\cdots\dfrac{\mathrm{vol}(P_m)}{\mathrm{vol}(P_{m-1})}}\mathrm{vol}(C_m)$$

where $P_0 = P$ and $P_i = C_i \cap P$ for $i = 1, \ldots, m$.

Annealing Schedule

How we compute the sequence of bodies

Inputs: Polytope P, error ϵ, cooling parameters $r, \delta > 0$ and a significance level (s.l.) $\alpha > 0$ s.t. $0 < r + \delta < 1$.

Output: A sequence of convex bodies $C_1 \supseteq \cdots \supseteq C_m$ s.t.

$\text{vol}(P_{i+1})/\text{vol}(P_i) \in [r, r+\delta]$ with high probability

where $P_i = C_i \cap P,\ i = 1, \ldots, m$ and $P_0 = P$.

t-test

one tailed

Let ν observations from a r.v. $X \sim \mathcal{N}(\mu, \sigma^2)$ with unknown variance σ^2.

- t-test checks the null hypothesis that the population mean exceeds a specified value μ_0 using the statistic $t = \frac{\bar{x}-\mu_0}{s/\sqrt{\nu}} \sim t_{\nu-1}$, *where $\bar{x}$: sample's mean, s: sample's st.d., $t_{\nu-1}$: t-student distribution with $\nu - 1$ degrees of freedom*
- Given a s.l. $\alpha > 0$ we test the null hypothesis for the mean value of the population, $H_0 : \mu \leq \mu_0$ against $H_1 : \mu > \mu_0$
- We say that we reject H_0 if,

$$t \geq t_{\nu-1,\alpha} \Rightarrow \bar{x} \geq \mu_0 + t_{\nu-1,\alpha} s/\sqrt{\nu}$$

which implies $\Pr(H_0 \text{ true} \mid \text{reject } H_0) = \alpha$ ($t_{\nu-1,\alpha}$ is the critical value). Otherwise we fail to reject H_0

t-test

for the ratio of volumes of convex bodies

Given convex bodies $P_i \supseteq P_{i+1}$, s.l. α, and cooling parameters $r, \delta > 0$, $0 < r + \delta < 1$, we define two t-tests:

testL$(P_i, P_{i+1}, r, \delta, \alpha, \nu, N)$:	**testR**$(P_i, P_{i+1}, r, \alpha, \nu, N)$:
H_0 : $\text{vol}(P_{i+1})/\text{vol}(P_i) \leq r + \delta$	H_0 : $\text{vol}(P_{i+1})/\text{vol}(P_i) \leq r$
H_1 : $\text{vol}(P_{i+1})/\text{vol}(P_i) \geq r + \delta$	H_1 : $\text{vol}(P_{i+1})/\text{vol}(P_i) \geq r$
Successful if we **fail to reject** H_0	**Successful** if we **reject** H_0

- **testL** and **testR** are used by annealing schedule to restrict each ratio $r_i = \text{vol}(P_{i+1})/\text{vol}(P_i)$ in the interval $[r, r + \delta]$.

t-test

Compute **testL**, **testR** for $P_i \supseteq P_{i+1}$

1. sample νN points from P_i and split the sample into ν sublists of length N.
2. The ν mean values are experimental values that follow $\mathcal{N}(r_i, (r_i(1-r_i))/N)$.
3. Use those values to perform two t-tests with null hypotheses:

$$\textbf{testL}\ H_0:\ r_i \leq r + \delta \qquad \textbf{testR}\ H_0:\ r_i \leq r$$

Assuming[a] the following holds,

$$r + \delta + t_{\nu-1,\alpha}\frac{s}{\sqrt{\nu}} \geq \hat{\mu} \geq r + t_{\nu-1,\alpha}\frac{s}{\sqrt{\nu}}$$

then r_i is restricted to $[r, r+\delta]$ with high probability.

[a] for the mean $\hat{\mu}$ of the ν mean values

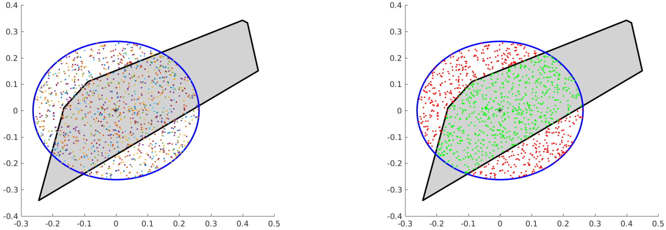
Initialization of Annealing Schedule

Given: polytope P, convex body C s.t. $C \cap P \neq \emptyset$, an interval $[q_{\min}, q_{\max}]$ and parameters $r, \delta, \alpha, \nu, N$.

- binary search for $q \in [q_{\min}, q_{\max}]$ s.t. both $\texttt{testL}(qC, qC \cap P)$ and $\texttt{testR}(qC, qC \cap P)$ are successful.
- sample from qC, rejection to P
- we call the successful qC, C'

1st iteration of the initialization step

- $[q_{\min}, q_{\max}] = [0.14, 0.38], \quad q = \frac{q_{\min}+q_{\max}}{2} = 0.26$
- $r = 0.8,\ \delta = 0.05,\ \alpha = 0.1,\ \nu = 10,\ \nu N = 1200 \Rightarrow N = 120$

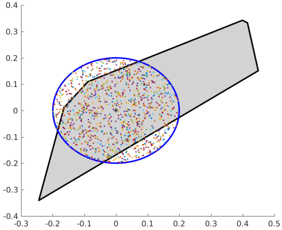


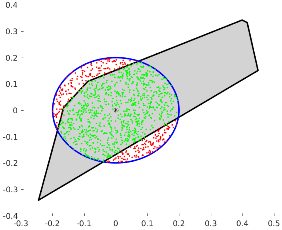
$$\hat{\mu} = \frac{r'^{(1)} + r'^{(2)} \dots + r'^{(10)}}{\nu} = \frac{684}{1200} = 0.57, \qquad s = 0.06$$

- testL$\rightarrow$ $H_0 : \frac{\text{vol}_q C \cap P}{\text{vol}(qC)} \le r + \delta \Rightarrow$ succeed $\quad$ (H_0 not rejected).
- testR$\rightarrow$ $H_0 : \frac{\text{vol}_q C \cap P}{\text{vol}(qC)} \le r \Rightarrow$ failed $\quad$ (H_0 not rejected).
- qC of 1st iteration is too big (the ratio is too small).

2nd iteration of the initialization step

- $[q_{\min}, q_{\max}] = [0.14, 0.26], \quad q = \frac{q_{\min}+q_{\max}}{2} = 0.20$
- $r = 0.8,\ \delta = 0.05,\ \alpha = 0.1,\ \nu = 10,\ \nu N = 1200 \Rightarrow N = 120$



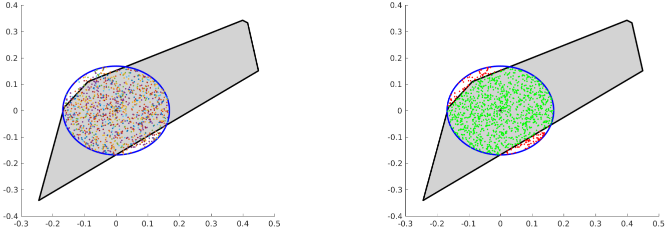


$$\hat{\mu} = \frac{r'^{(1)}+r'^{(2)}\cdots+r'^{(10)}}{\nu} = \frac{914}{1200} = 0.76, \qquad s = 0.04$$

- `testL`$\rightarrow \ H_0 : \frac{\text{vol}_{qC\cap P}}{\text{vol}(qC)} \leq r + \delta \Rightarrow$ succeed $\quad (H_0$ not rejected).
- `testR`$\rightarrow \ H_0 : \frac{\text{vol}_{qC\cap P}}{\text{vol}(qC)} \leq r \Rightarrow$ failed $\quad (H_0$ not rejected).
- qC of 2nd iteration is too big (the ratio is too small).

3rd iteration of the initialization step

- $[q_{\min}, q_{\max}] = [0.14, 0.20], \quad q = \frac{q_{\min}+q_{\max}}{2} = 0.17$
- $r = 0.8,\ \delta = 0.05,\ \alpha = 0.1,\ \nu = 10,\ \nu N = 1200 \Rightarrow N = 120$

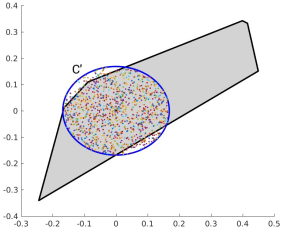


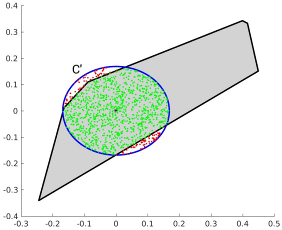
$$\hat{\mu} = \frac{r'^{(1)} + r'^{(2)} \cdots + r'^{(10)}}{\nu} = \frac{1076}{1200} = 0.90, \qquad s = 0.02$$

- testL$\rightarrow$ $H_0 : \frac{\text{vol}_{qC \cap P}}{\text{vol}(qC)} \leq r + \delta \Rightarrow$ failed $\quad (H_0$ rejected).
- testR$\rightarrow$ $H_0 : \frac{\text{vol}_{qC \cap P}}{\text{vol}(qC)} \leq r \Rightarrow$ succeed $\quad (H_0$ rejected).
- qC of 3rd iteration is too small (the ratio is too big).

4th iteration of the initialization step

- $[q_{\min}, q_{\max}] = [0.17, 0.20], \quad q = \frac{q_{\min}+q_{\max}}{2} = 0.185$
- $r = 0.8,\ \delta = 0.05,\ \alpha = 0.1,\ \nu = 10,\ \nu N = 1200 \Rightarrow N = 120$





$\hat{\mu} = \frac{r'^{(1)} + r'^{(2)} \cdots + r'^{(10)}}{\nu} = \frac{993}{1200} = 0.83, \qquad s = 0.04$

- testL$\rightarrow$ $H_0 : \frac{\text{vol}_{qC \cap P}}{\text{vol}(qC)} \leq r + \delta \Rightarrow$ succeed $\quad$ (H_0 not rejected).
- testR$\rightarrow$ $H_0 : \frac{\text{vol}_{qC \cap P}}{\text{vol}(qC)} \leq r \Rightarrow$ succeed $\quad$ (H_0 rejected).
- Set $C' = qC$.

Annealing Schedule

Iteration & Stopping criterion

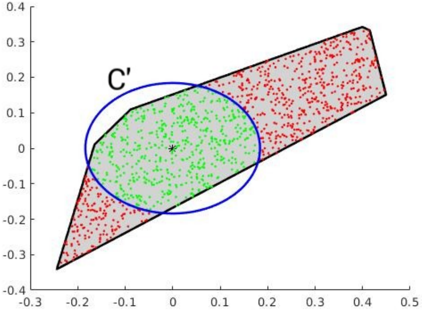
- In step i of the annealing schedule sample νN points from P_i to perform $\texttt{testR}(P_i, C' \cap P)$.
- The schedule stops if volume ratio $\text{vol}(C' \cap P)/\text{vol}(P_i)$ is large enough according to `testR`.

Stop in step i if $\texttt{testR}(P_i, C' \cap P)$ succeeds.
Set $m = i + 1$ and $P_m = C' \cap P, C_m = C'$.

- Otherwise, construct P_{i+1}: binary search for qC s.t. succeed $\texttt{testL}(P_i, P_{i+1})$, $\texttt{testR}(P_i, P_{i+1})$ using the same sample in P_i

Check stopping criterion

- $i = 0$, $P_0 = P$, sample νN points from P_0.
- $r = 0.8,\ \delta = 0.05,\ \alpha = 0.1,\ \nu = 10,\ \nu N = 1200 \Rightarrow N = 120$

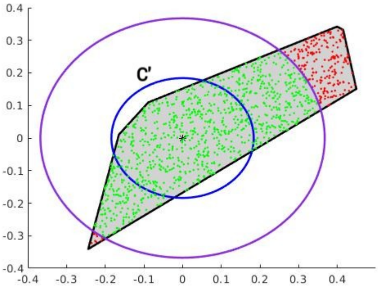


$$\hat{\mu} = \frac{r_0^{(1)} + r_0^{(2)} \cdots + r_0^{(10)}}{\nu} = \frac{535}{1200} = 0.45, \qquad s = 0.08$$

- testR$\rightarrow$ $H_0 : r_0 \leq r \Rightarrow$ failed $\quad$ (H_0 not rejected).
- Define P_1.

Define next convex body in MMC

- $i = 0$, $P_0 = P$, binary search for $q \in [0.185, 0.38]$
- $r = 0.8,\ \delta = 0.05,\ \alpha = 0.1,\ \nu = 10,\ \nu N = 1200 \Rightarrow N = 120$

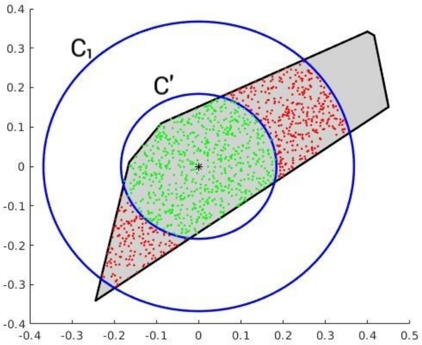


$q = 0.36, \quad \hat{\mu} = \frac{r_0^{(1)} + r_0^{(2)} \cdots + r_0^{(10)}}{\nu} = \frac{998}{1200} = 0.83, \quad s = 0.07$

- testL$\rightarrow$ $H_0 : r_0 \leq r + \delta \Rightarrow$ succeed (H_0 not rejected).
- testR$\rightarrow$ $H_0 : r_0 \leq r \Rightarrow$ succeed (H_0 rejected).
- Set $C_1 = qC$.

Check stopping criterion

- $i = 1$, $P_1 = C_1 \cap P$, sample νN points from P_1.
- $r = 0.8,\ \delta = 0.05,\ \alpha = 0.1,\ \nu = 10,\ \nu N = 1200 \Rightarrow N = 120$

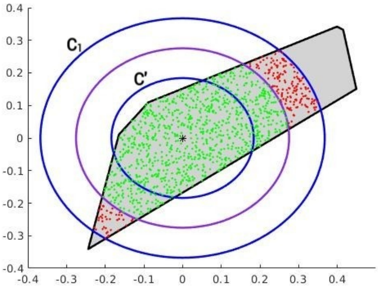


$$\hat{\mu} = \frac{r_1^{(1)} + r_1^{(2)} \cdots + r_1^{(10)}}{\nu} = \frac{535}{1200} = 0.57, \qquad s = 0.07$$

- testR$\rightarrow$ $H_0 : r_1 \leq r \Rightarrow$ failed (H_0 not rejected).
- Define P_2.

Define next convex body in MMC

- $i = 1$, $P_1 = C_1 \cap P$, binary search for $q \in [0.185, 0.36]$
- $r = 0.8,\ \delta = 0.05,\ \alpha = 0.1,\ \nu = 10,\ \nu N = 1200 \Rightarrow N = 120$

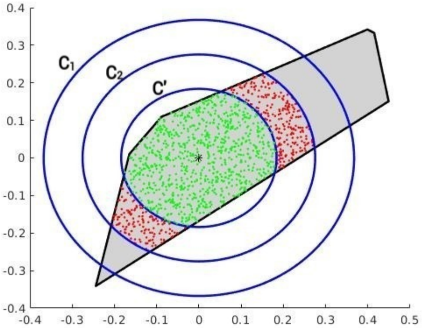


$$q = 0.28, \quad \hat{\mu} = \frac{r_1^{(1)} + r_1^{(2)} \cdots + r_1^{(10)}}{\nu} = \frac{998}{1200} = 0.82, \qquad s = 0.06$$

- testL$\rightarrow$ $H_0 : r_1 \leq r + \delta \Rightarrow$ succeed (H_0 not rejected).
- testR$\rightarrow$ $H_0 : r_1 \leq r \Rightarrow$ succeed (H_0 rejected).
- Set $C_2 = qC$.

Check stopping criterion

- $i = 2$, $P_2 = C_2 \cap P$, sample νN points from P_2.
- $r = 0.8,\ \delta = 0.05,\ \alpha = 0.1,\ \nu = 10,\ \nu N = 1200 \Rightarrow N = 120$

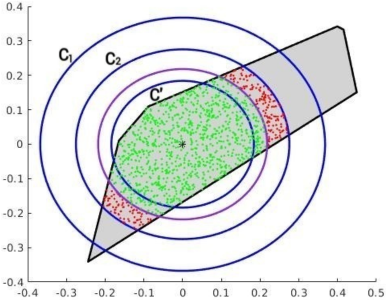


$$\hat{\mu} = \frac{r_2^{(1)} + r_2^{(2)} \cdots + r_2^{(10)}}{\nu} = \frac{833}{1200} = 0.69, \qquad s = 0.08$$

- testR$\rightarrow$ $H_0 : r_2 \leq r \Rightarrow$ failed $\quad (H_0$ not rejected$)$.
- Define P_3.

Define next convex body in MMC

- $i = 2$, $P_2 = C_2 \cap P$, binary search for $q \in [0.185, 0.28]$
- $r = 0.8,\ \delta = 0.05,\ \alpha = 0.1,\ \nu = 10,\ \nu N = 1200 \Rightarrow N = 120$

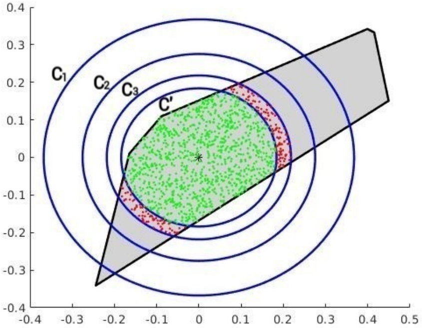


$$q = 0.21, \quad \hat{\mu} = \frac{r_2^{(1)} + r_2^{(2)} \cdots + r_2^{(10)}}{\nu} = \frac{976}{1200} = 0.81, \qquad s = 0.05$$

- testL$\rightarrow$ $H_0 : r_2 \leq r + \delta \Rightarrow$ succeed (H_0 not rejected).
- testR$\rightarrow$ $H_0 : r_2 \leq r \Rightarrow$ succeed (H_0 rejected).
- Set $C_3 = qC$.

Check stopping criterion

- $i = 3$, $P_3 = C_3 \cap P$, sample νN points from P_3.
- $r = 0.8,\ \delta = 0.05,\ \alpha = 0.1,\ \nu = 10,\ \nu N = 1200 \Rightarrow N = 120$



$$\hat{\mu} = \frac{r_3^{(1)} + r_3^{(2)} \cdots + r_3^{(10)}}{\nu} = \frac{1026}{1200} = 0.86, \quad s = 0.04$$

- `testR`$\rightarrow$ $H_0 : r_3 \le r \Rightarrow$ succeed $(H_0$ rejected).
- Set $m = 4$, $C_4 = C'$.

Estimate Ratio

Sliding window

For $i = 1, \ldots, m$ (m computed by schedule) estimate $r_i = \text{vol}(P_{i+1})/\text{vol}(P_i)$.

- Generate points in P_i. For each new point keep the value of r_i.
- Store the last n values in a queue called sliding window W.
- Update W for each new sample point, by inserting the new value of r_i and by popping out the oldest ratio value.

$\cdots$	r_i^j	$\cdots$	r_i^{j+n-1}

Convergence criterion

- W is a set of n values following Gaussian distribution
- Consider the mean value $\hat{\mu}$, the st.d. s of W and $Pr = \sqrt[m+1]{3/4}$. For each r_i assign a value ϵ_i (is the maximum allowed error for r_i), s.t. $\sum_i \epsilon_i^2 = \epsilon^2$.
- Using $p = (1 + Pr)/2$ and the interval $[\hat{\mu} - z_p s, \hat{\mu} + z_p s]$:

$$\text{if } \frac{(\hat{\mu} + z_p s) - (\hat{\mu} - z_p s)}{\hat{\mu} + z_p s} = \frac{2 z_p s}{\hat{\mu} + z_p s} \leq \epsilon_i/2, \text{ then declare convergence.}$$

Convergence criterion

- W is a set of n values following Gaussian distribution
- Consider the mean value $\hat{\mu}$, the st.d. s of W and $Pr = \sqrt[m+1]{3/4}$. For each r_i assign a value ϵ_i (is the maximum allowed error for r_i), s.t. $\sum_i \epsilon_i^2 = \epsilon^2$.
- Using $p = (1 + Pr)/2$ and the interval $[\hat{\mu} - z_p s, \hat{\mu} + z_p s]$:

$$\text{if } \frac{(\hat{\mu} + z_p s) - (\hat{\mu} - z_p s)}{\hat{\mu} + z_p s} = \frac{2 z_p s}{\hat{\mu} + z_p s} \leq \epsilon_i/2, \text{ then declare convergence.}$$

Remark:

If we assume (perfect) uniform sampling from P_i:

1. There is $n = O(1)$ s.t. $r_i \in [\hat{\mu} - z_p s, \hat{\mu} + z_p s]$ with $Pr = \sqrt[m+1]{3/4}$.
2. method estimates vol(P) with probability $3/4$ and error $\leq \epsilon$.

In practice it is open how to bound n

The volume algorithm

Algorithm 1 `VolumeAlgorithm` $(P, \epsilon, r, \delta, \alpha, \nu, N, n)$

Construct $C \subseteq \mathbb{R}^d$ s.t. $C \cap P \neq \emptyset$ and set $[q_{\min}, q_{\max}]$
$\{P_0, \ldots, P_m, C_m\}$=`AnnealingSchedule`$(P, C, r, \delta, \alpha, \nu, N, q_{\min}, q_{\max})$
$\epsilon_m = \epsilon/2\sqrt{m+1}$, $\epsilon' = \epsilon\sqrt{4(m+1)-1}/2\sqrt{m+1}$
Set $\epsilon_i = \epsilon'/\sqrt{m}$, $i = 0, \ldots, m-1$
for $i = 0, \ldots m$ **do**
  if $i < m$ **then**
    $r_i =$ `EstimateRatio` $(P_i, P_{i+1}, \epsilon_i, m, n)$
  else
    $r_m =$ `EstimateRatio` $(C_m, P_m, \epsilon_m, m, n)$
  end if
end for
return $\text{vol}(C_m)/r_0/\cdots/r_{m-1} \cdot r_m$

Bounds on #phases

- Let the power of a i-th t-test
  $pow_i := \Pr(H_0 \text{ false} \mid \text{reject } H_0) = 1 - \beta_i$.

Proposition

Given $P \subset \mathbb{R}^d$ and cooling parameters r, δ s.t. $r + \delta < 1/2$ and a s.l. α, let m be the number of balls in MMC. Then $m < d \lg(d)$ with probability $p \geq 1 - \max\{\beta_i\} = \min\{pow_i\},\ i = 0, \ldots, m$.

Proposition

Given P, the body C that minimizes the expected number of phases in MMC, for given cooling parameters r, δ and s.l. α is the one that maximizes vol$(C' \cap P)$ in the annealing schedule initialization

Implementation

VolEsti: sampling and volume estimation

- `C++`, `R`-interface, `python` bindings (in review)
- open source
- since 2014, CGAL (not any more), Eigen, LPSolve
- design: mix of obj oriented + templates, C++03
- main developers: VF, Tolis Chalkis
- `https://github.com/GeomScale/volume_approximation`
- Contributions welcome!

C++ VolEsti example

```cpp
template <typename NT, class RNGType, class Polytope>
void test_volume(Polytope &HP)
{
    // Setup the parameters
    int n = HP.dimension();
    int walk_len=10 + n/10;
    int nexp=1, n_threads=1;
    NT e=1, err=0.0000000001;
    int rnum = std::pow(e,-2) * 400 * n * std::log(n);
    unsigned seed = std::chrono::system_clock::now()
                    .time_since_epoch().count();
    RNGType rng(seed);
    boost::normal_distribution<> rdist(0,1);
    boost::random::uniform_real_distribution<>(urdist);
    boost::random::uniform_real_distribution<> urdist1(-1,1);

    vars<NT, RNGType> var(rnum,n,walk_len,n_threads,err,e,0,0,0,0,
        rng,urdist,urdist1,-1.0,false,false,false,false,false,fals

    //Compute chebychev ball and volume//
    typedef typename Polytope::PolytopePoint Point;
    std::pair<Point,NT> CheBall;
    CheBall = HP.ComputeInnerBall();
    std::cout << volume(HP,var,var,CheBall) << std::endl;
}
```

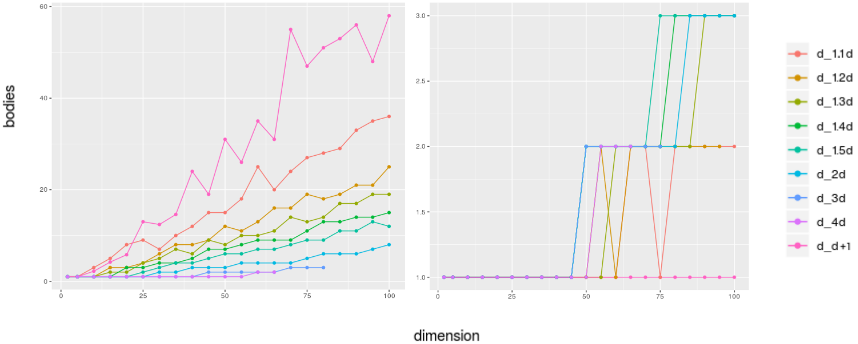
Experiments

Experimentally determine the parameters

- **Anneling schedule:** set $r = 0.1$ and $\delta = 0.05$ in order to define $\mathrm{vol}(P_{i+1}$ to be about 10% of $\mathrm{vol}(P_i)$.
  Set $\nu N = 1200 + 2d^2$ and $\nu = 10$.
- **Sliding window:** set window length $n = 2d^2 + 250$.
- **Hit-and-Run:** step equal to 1.

Zonotopes

Number of phases



Two types of bodies: balls (left) symmetric H-polytopes (right)

- For low order (i.e. $\#generators/d$) zonotopes, ≤ 4, the number of bodies is smaller than the case of using balls
- For balls the number of phases reduces as the order increases for a fixed dimension

Zonotopes

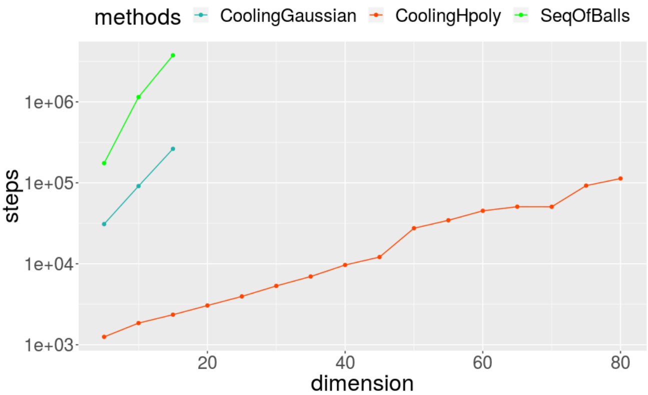
Performance

Experimental results for zonotopes.						
z-d-k	*Body*	*order*	*Vol*	*m*	*steps*	*time*(sec)
z-5-500	Ball	100	4.63e+13	1	0.1250e+04	22.26
z-20-2000	Ball	100	2.79e+62	1	0.2000e+04	1428
z-50-65	Hpoly	1.3	1.42e+62	1	1.487e+04	173.9
z-100-130	Hpoly	1.3	1.37e+138	3	17.19e+04	6073
z-50-75	Hpoly	1.5	2.96e+66	2	1.615e+04	253.6
z-100-150	Hpoly	1.5	2.32+149	3	15.43e+04	10060
z-70-140	Hpoly	2	8.71e+111	2	5.059e+04	2695
z-100-200	Hpoly	2	5.27e+167	3	15.25e+04	34110

- *z-d-k*: random zonotope in dimension d with k generators; *Body*: the type of body used in MMC; m: number of bodies in MMC
- Used to evaluate zonotope approximation methods in engineering [Kopetzki'17]

Comparison with state-of-the-art

Zonotopes



The number of steps for random zonotopes of order 2

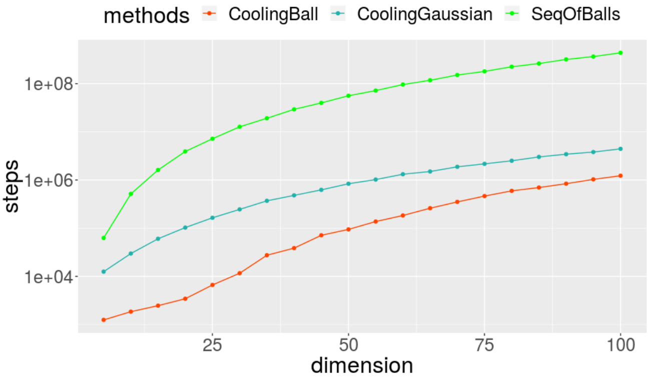
- Our method is asymptotically better
- `CoolingGaussian` and `SeqOfBalls` takes $> 2hr$ for $d > 15$.
- for higher orders the difference is larger

V-polytopes

P	Vol	m	steps	error	time	ex. Vol	ex. time
cross-60	1.27e-64	1	20.6e+03	0.08	60.1	−−	−−
cross-100	1.51e-128	2	94.2e+03	0.11	406	−−	−−
Δ-60	1.08e-82	10	77.4e+04	0.1	1442	1.203-82	0.02
Δ-80	1.30e-119	13	187e+04	0.07	6731	1.39e-119	0.07
cube-10	1052.4	1	1851	0.03	54.39	−−	−−
cube-13	7538.2	1	2127	0.08	2937	−−	−−
rv-10-80	3.74e-03	1	0.185e+04	0.08	3.35	3.46e-03	6.8
rv-10-160	1.59e-02	1	0.140e+04	0.06	5.91	1.50e-03	59
rv-15-30	2.73e-10	1	0.235e+04	0.02	3.46	2.79e-10	2.1
rv-15-60	4.41e-08	1	0.235e+04	−−	6.46	−−	−−
rv-20-2000	2.89e-07	1	0.305e+04	−−	457	−−	−−
rv-80-160	5.84e-106	3	11.3e+04	−−	3250	−−	−−
rv-100-200	1.08e-141	4	24.5e+04	−−	13300	−−	−−

time: the average time in seconds; ex. Vol: the exact volume; ex. time: the time in seconds for the exact volume computation i.e. `qhull` in `R` (package `geometry`); m is the number of phases. −− implies that the execution failed due to memory issues or exceeded 1 hr.

H-polytopes



- The number of steps for unit cubes in H-representation
- for $d = 100$: > 2 faster than `CoolingGaussian`and ~ 100x than `SeqOfBalls`

Future directions

- Better (practical) sampling methods (candidate: Hamiltonian walk) + implementation

Future directions

- Better (practical) sampling methods (candidate: Hamiltonian walk) + implementation
- Study convex body families for MMC

Future directions

- Better (practical) sampling methods (candidate: Hamiltonian walk) + implementation
- Study convex body families for MMC
- Applications:
  - zonotope approximation in engineering
  - finance [Cales, Chalkis, Emiris, F, SoCG'18]

Future directions

- Better (practical) sampling methods (candidate: Hamiltonian walk) + implementation
- Study convex body families for MMC
- Applications:
  - zonotope approximation in engineering
  - finance [Cales, Chalkis, Emiris, F, SoCG'18]
- Google Summer of Code '19 call for internships! (contact me :)

Future directions

- Better (practical) sampling methods (candidate: Hamiltonian walk) + implementation
- Study convex body families for MMC
- Applications:
  - zonotope approximation in engineering
  - finance [Cales, Chalkis, Emiris, F, SoCG'18]
- Google Summer of Code '19 call for internships! (contact me :)

GSOC19: Project 1. Sampling scalability

- empirical study of random walks for convex polytopes (mainly given by a set of linear inequalities).
- Currently variations of hit-and-run random walks are used but there are methods with better mixing time; e.g. hamiltonian walk https://arxiv.org/pdf/1710.06261.pdf.
- expectation: dramatic effect in the scaling of the underlying algorithms
- ultimate goal: "scale from a few hundred dimensions to a few thousand"

We can divide the coding project in the following steps:

- Understand the code structure and design of VolEsti.
- Create prototypes for new sampling algorithms (the main focus will be in Hamiltonian walk but we may investigate others too).
- Implement the best representatives from the previous step in VolEsti and create R interfaces.
- Write tests and documentation.

GSOC19: Project 2. Sampling and volume of spectahedra

- non-linear extension for VolEsti
- Spectahedra are feasible regions of semidefinite programs and are known to be convex.
- important role in optimization: the “next more understandable” convex objects after polytopes
- algorithms for sampling and volume computation will shed more light towards their study.

We can divide the coding project in the following steps:

- Understand the code structure and design of VolEsti. Understand the basics for spectahedra from bibliography.
- Implement a new convex body type and boundary oracle for spectahedra.
- Work on extensions of the problem such as replacing spectahedra by a spectahedral shadow.
- Write tests and documentation.

GSOC19: Project 3. Convex optimization with randomized algorithms

Design and implementation of optimization algorithms (available in relevant bibliography) in VolEsti that utilize sampling (already available in the library) as a main subroutine.
We can divide the coding project in the following steps:

- Understand the code structure and design of VolEsti.
- Implement optimization algorithms. A good place to start is Dabbene "A Randomized Cutting Plane Method with Probabilistic Geometric Convergence" Siam JOPT 2010
- Test implementations with various random walks available in VolEsti
- Write tests and documentation