Generalized permutahedral tropical sampling

Michael Borinsky, ETH Zürich - Institute for Theoretical Studies

July 10-15, SIAM Conference on Applied Algebraic Geometry 2023

Based on arXiv:2008.12310
Annales de l'Institut Henri Poincaré D

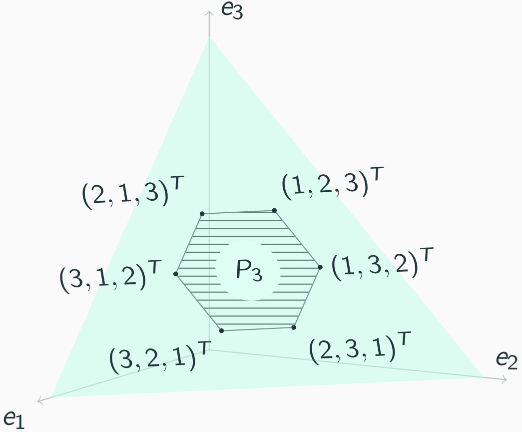
Extension to toric varieties and statistics arXiv:2204.06414
with Anna-Laura Sattelberger, Bernd Sturmfels, Simon Telen
SIAGA

Extension to Feynman Integrals in Minkowski space arXiv:2302.08955
with Henrik Munch and Felix Tellander
preprint

Generalized permutahedra

Permutahedra

$$P_n = \mathrm{Conv}\{(\sigma_1, \ldots, \sigma_n)^T \in \mathbb{R}^n : \sigma \in \mathbb{S}_n\}$$



Suppose a (co)vector $y \in (\mathbb{R}^n)^*$ is given.

Question (linear program)

For which point $u \in P_n$ is $y \cdot u = y_1 u_1 + \ldots + y_n u_n$ maximal?

Answer

For any $\sigma \in \mathbb{S}_n$ that respects the ordering of the y-components:

$$y_{\sigma_1} \leq \cdots \leq y_{\sigma_n}$$

The product $y \cdot u$ is maximized over $u \in P_n$ if $u = U(\sigma)$ given by

$$U_{\sigma_k}(\sigma) = k$$

because

$$y_{\sigma_1} U_{\sigma_1}(\sigma) + \cdots + y_{\sigma_n} U_{\sigma_n}(\sigma) = y_{\sigma_1} 1 + y_{\sigma_2} 2 + \cdots + y_{\sigma_n} n$$

Definition

For generalized permutahedra the answer to linear programming problem only depends on the ordering of the components of $y \in (\mathbb{R}^n)^*$.

Surprise feature

The linear programming problem has an efficient solution for generalized permutahedra.

I.e. $U(\sigma)$ can be computed in polynomial time in n.

Literature

- Initial definitions, volumes, invariants, etc:
  Postnikov 2005; **Postnikov-Reiner-Williams 2006**
- Connections linear programming, matroids, Hopf algebras, etc:
  Aguiar-Ardila 2017

... generalized permutahedra are interesting because

- Cool combinatorics (e.g. Lorentzian polynomials)
- Fancy algebra (e.g. Hopf algebras)
- Nice physics (e.g. Feynman integrals)

Application to integration

Let $f(x) \in \mathbb{R}[x_1, \ldots, x_n]$.

We want to (numerically) evaluate

$$\int_{\mathbb{R}_+^n} \frac{\mathrm{d}x_1 \cdots \mathrm{d}x_n}{f(x)}$$

Intermezzo: Monte Carlo integration

Suppose we have a probability measure

$$\mu = \frac{\mathrm{d}x_1 \cdots \mathrm{d}x_n}{w(x)} > 0 \quad \text{with} \quad \int_{\mathbb{R}_+^n} \mu = 1,$$

and a (efficient) way to sample points $x^{(1)}, \ldots, x^{(N)} \in \mathbb{R}_+^n$ from it.

Then we can (try to) estimate using

$$\int_{\mathbb{R}_+^n} \frac{\mathrm{d}x_1 \cdots \mathrm{d}x_n}{f(x)} = \int_{\mathbb{R}_+^n} \frac{\mathrm{d}x_1 \cdots \mathrm{d}x_n}{w(x)} \frac{w(x)}{f(x)} \approx \frac{1}{N} \sum_{i=1}^{N} \frac{w(x^{(i)})}{f(x^{(i)})}$$

We need to choose $w(x)$ such that

1. $w(x)$ is 'similar enough' to $f(x)$ on $\mathbb{R}_+^n$.
2. we can quickly sample from

$$\mu = \frac{\mathrm{d}x_1 \cdots \mathrm{d}x_n}{w(x)}.$$

Tropical approximation

Idea: a 'tropicalization' of $f(x)$ as weight function $w(x)$.

$\mathbf{N}[f]$ is the *Newton polytope* of $f(x)$.

Let

$$\mathrm{Trop}_f : y \mapsto \max_{u \in \mathbf{N}[f]} y \cdot u$$

and

$$w(x) = \exp\left(\mathrm{Trop}_f(\log x)\right),$$

where $\log x = (\log x_1, \ldots, \log x_n)$.

Theorem MB 2020: $w(x)$ does a decent job approximating $f(x)$.

Remaining problem: How can we sample from

$$\mu = \frac{\mathrm{d}x_1 \cdots \mathrm{d}x_n}{w(x)}?$$

Tropical sampling

Stochastic (in)version of the linear programming question:

Given a polytope P

draw a random covector $y \in (\mathbb{R}^n)^*$ with probability proportional to

$$\exp\left(-\max_{u \in P} y \cdot u\right)$$

(solves the sampling problem for

$$\mu = \frac{\mathrm{d}x_1 \cdots \mathrm{d}x_n}{w(x)}$$

with $P = \mathbf{N}[f]$)

Doable for general polytopes P **MB 2020**.
But computationally demanding (P has to be triangulated).

There is a ‘fast’ way to solve the ‘stochastic linear programming problem’ if P is a generalized permutadron

More on generalized permutahedra

A boolean function $z : 2^{[n]} \to \mathbb{R}$
(i.e. a function from all subsets of $[n] = \{1, \ldots, n\}$ to $\mathbb{R}$)
is supermodular if

$$z(A) + z(B) \leq z(A \cap B) + z(A \cup B) \text{ for all } A, B \subseteq [n].$$

Theorem Aguiar-Ardila 2017

Supermodular boolean functions are in 1-to-1 correspondence to generalized permutahedra.

‘Fast’ tropical sampling algorithm for gen. permutahedra P_z

MB 2020

Preprocessing:

Define a boolean function $J : 2^{[n]} \to \mathbb{R}$ recursively by $J(\emptyset) = 1$ and

$$J(A) = \sum_{e \in A} \frac{J(A \setminus e)}{z(A \setminus e)} \quad \text{for all } A \subset [n]$$

Algorithm:

Start with $A = [n]$ and $\kappa = 0$

1. Draw a random $e \in A$ with probability $p_e = \frac{1}{J(A)} \frac{J(A \setminus e)}{z(A \setminus e)}$
2. Remove e from A.
3. Set $y_e = \kappa$.
4. Draw $\xi \in [0, 1]$ uniformly and set $\kappa \to \kappa + \frac{1}{z(A)} \log(\xi)$.
5. If $A \neq \emptyset$, go back to 1.

Result: A sample $y \in (\mathbb{R}^n)^*$ distributed as

$$\exp\left(-\max_{u \in P_z} y \cdot u\right)$$

Gives a 'fast' integration algorithm

$$\int_{\mathbb{P}^n_{>0}} \frac{f(x)}{g(x)}\Omega$$

Theorem

If the Newton polytopes of $f(x)$ and $g(x)$ are gen. permutahedra.
And $g(x)$ is completely non-vanishing on $\mathbb{P}^n_{>0}$.
And the integral exists.

Then it can be evaluated up to δ relative accuracy in time

$$\mathcal{O}(n2^n + n^2 F(n)\delta^{-2}),$$

where $F(n) =$ [time to evaluate $f(x)/g(x)$ for one values of x].

Summary

Generalized permutahedral tropical sampling is orders of magnitude faster than the naive way.

$\Rightarrow$ Fastest algorithm to evaluate Feynman integrals.

Open question

Is there a polynomial time algorithm?

I.e. improve preprocessing runtime of $n2^n$ in $\mathcal{O}(n2^n + n^2 F(n)\delta^{-2})$