Noname manuscript No.
(will be inserted by the editor)

Tensor polytopes, Hom polytopes, Hyper-matrices and data clustering

Andrew Dynneson M.A.

Received: date / Accepted: date

Abstract Expected quantities are presented for the facets of the Tensor-polytope and the number of vertices of the Hom-polytope, for regular polygons as inputs. A cluster analysis was performed on the resulting data. The clustering scripts are provided to aid in reproducing the method and furthering the results. The concept of the standard matrix is extended to arbitrary dimensionality resulting in Hyper-matrices. The Hypercube matrix is used to represent Tensors, and has numerous potential applications in physics and various fields of research such as finance and statistics.

Keywords Polytope · Matrix · Tensor · Hom · Linear Algebra · Relativity · Gravitation · Geometry · Hypercube · Cluster analysis

PACS 04.20.-q · 02.10.Xm · 02.10.Yn · 02.40.Ft · 02.40.Ky · 02.60.Cb · 89.20.Ff

Mathematics Subject Classification (2010) 15Bxx · MSC 65D99 · 52B12 · 51P05 · 68W99 · 83C05

1 Introduction

The Hom-polytope and Tensor-polytope are on the one-hand at the most fundamental for the study of the category of polytopes. And, while the facets of the Hom-polytope and the vertices of the Tensor are inherent from their definition, very little is known about the quantities of the vertices of the Hom-polytope nor the facets of the Tensor product. These are difficult to derive, often numbering in the thousands, and the slightest perterbation of vertex risks miscounting the number of objects in the resulting structure.

Andrew Dynneson
University of Texas at El Paso
Tel.: +18312722441
E-mail: awdynneson@utep.edu

This paper rests heavily on cluster analysis, since intuitively, if extra objects are created by the error of computation, then the extra objects should be "nearby" enough to the actual object to form well-defined and stable clusters. The clustered data can also provide valuable information regarding the relative maginitude distance of the vertices, and the angles of the facets. The software Polymake [3] was used to collect information for the approximate calculations, using rationally-approximated regular-polygons, triangle through decagon, for both Hom and Tensor. Then a thourough cluster-analysis was performed on this data using the script written in Perl, via the use of six personal computers running simultaneously. All of the clustering data is presented, as well as expected values for the quantities of vertices and facets.

Next, the concept of the standard matrix is extended to arbitrary dimensionality, adding layers of computation, whose purpose is to construct a graphical representation of a Tensor. The idea of the hypercube-matrix arose naturally as the result of an in-depth examination of the Riemann Tensor from Einstein's Theory of Relativity, which is used to describe the curvature of space-time. As such, I will be using as a motivating source Misner, Thorne, and Wheeler's Gravitation [6]. The Hypercube matrix was invaluable in aiding the visualization of the Riemann Tensor, as well as relating it to topics in linear algebra. The Tensor is a prevalent mechanism in physics. Furthermore, its connections to linear algebra are unmistakable. When viewed as a transformation, a hypercube matrix is an n-linear function with n vectors as inputs, and the desired number of outputs, as vectors. This new type of matrix is capable of adding "layers" of computation to the flat matrices to form a cube-matrix. Then, layers of cube-matrices are added to form hypercube matrices, and so-on, capable of addressing as many input and as many output vectors as desired.

The idea of adding dimensionality and generality to the concept of matrix could prove invaluable, not only in the applied realm of Physics; also in other fields. For example, the hypercube matrix could be used to analyze multiple aspects of risk in finance at one time, beyond the mere two variables often used at present. Similarly, one could analyze the multiple-correleation instead of compairing correlations pair-wise in statistics.

This article builds on concepts from Abstract and Linear Algebra, Polytopal Geometry, and General Relativity.

1.1 Preliminaries

For the background information as it pertains to Convex Polyhedral Geometry, the reader is referred to Polytopes, Rings, and K-theory [4]. For a concise overview of background, see [2]. For a good exposition of Tensor algebra, the reader is referred to Dummit and Foote [5].

The polytope which is generated as the convex-hull of a set of points is denoted $\mathbf{conv}(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$.

The affine-hull of a set S is $\mathbf{aff}(S) := \{\sum\limits_{i=1}^{n} \lambda_i \mathbf{x}_i : \lambda_i \in \mathbb{R} \wedge \sum_{i=1}^{n} \lambda_i = 1\}$

An affine space $\mathbb{A}$ is a subset of a vector-space V such that it is closed under affine-combinations, that is, if $\mathbb{A} = \mathbf{aff}(\mathbb{A})$. I claim that this is equivalent to $\mathbb{A}$ being equal to a translated subspace U of V, that is, there exists $x \in V$ and $\exists U < V$ such that $\mathbb{A} = U + x$. The dimension is denoted $\dim \mathbb{A} = \dim U$.

Theorem 1 *The following are equivalent, and each type of object can be expressed as the other.*

(1) Let U be a vector-subspace of V, and $x \in V$. Then $x + U := \{x + u | u \in U\}$;

(2) The affine hull, $\mathbf{aff}(X)$, for $X \subset V$.

Proof (1) $\Rightarrow$ (2) : Begin with a space of the form $x + U$. Let $\sum\limits_{i=1}^{n} \lambda_i(x + u_i)$ be an affine combination of elements of $x + U$. Then:

$$\sum_{i=1}^{n} \lambda_i(x + u_i) = \left(\sum_{i=1}^{n} \lambda_i\right) x + \sum_{i=1}^{n} u_i = x + \sum_{i=1}^{n} u_i \in x + U.$$

Hence $x + U$ is closed under affine combinations.

(2) $\Rightarrow$ (1): Now assume that $x \in X$ and without loss of generality, $0 \notin \mathbf{aff}(X)$. Let $U = \mathbf{lin}(X)$. It is shown that $\mathbf{aff}(X) = x + U$.

($\subseteq$): Let $\sum\limits_{i=1}^{n} \lambda_i x_i \in \mathbf{aff}(X)$. Then:

$$\sum_{i=1}^{n} \lambda_i x_i = x + \left(\sum_{i=1}^{n} \lambda_i x_i\right) - \left(\sum_{i=1}^{n} \lambda_i\right) x = x + \left(\sum_{i=1}^{n} \lambda_i (x_i - x)\right) \in x + U.$$

($\supseteq$): For $\sum\limits_{i=1}^{n} \mu_i u_i \in \mathbf{lin}(X)$, each $u_i = a(x_i - x) \exists a \in \mathbb{R}, x_i \in X$. Let $\gamma_i = a\mu_i$. Then:

$$x + \sum_{i=1}^{n} \mu_i u_i = x + \sum_{i=1}^{n} \gamma_i (x_i - x) = 1 \cdot x + \left(\sum_{i=1}^{n} \gamma_i x_i\right) - \left(\sum_{i=1}^{n} \gamma_i\right) x \in \mathbf{aff}(X).$$

The join of two polytopes in their ambient vector-spaces, $P \subset V$ and $Q \subset W$, is $\mathbf{join}(P, Q) = \mathbf{conv}[(x, 0, 0), (0, 1, y) | x \in P, y \in Q] \subset V \oplus \mathbb{R} \oplus W$. In Section 3, I will prove a result utilizing multiple join-layers.

P, Q, and R will denote polytopes unless otherwise specified. The n-dimensional standard simplex is $\Delta_n := \mathbf{conv}(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{n+1})$. The set of facets of a polytope is denoted $\mathbb{F}(P)$, and the set of vertices is $\mathbf{vert}(P)$. Define the dimension of a polytope, $\dim P(\mathbf{aff})$. The number of elements in a set S is denoted $\#S$.

The standard regular polygon is $P_n := \mathbf{conv}(1, \zeta_n, \zeta_n^2, \ldots, \zeta_n^{n-1}) \subset \mathbb{C} \cong \mathbb{R} \oplus \mathbb{R}$, where $\zeta_n = \cos(2\pi/n) + i \sin(2\pi/n)$.

2 Cluster Analysis

Definition 1 Let $\varepsilon > 0$ be given. $V = \{v_0, v_1, \ldots, v_n\}$ is a finite collection of vectors. $\mathit{\Gamma} = \{P_0, P_1, \ldots, P_m\}$ is a finite collection of sets. Then $\mathit{\Gamma}$ is an *ε-clustering* of V if it satisfies the following properties:

1. $\bigcup\limits_{i=0}^{m} P_i = V.$
2. $\forall P, Q \in \mathit{\Gamma}, P \cap Q = \emptyset.$
3. $\forall P \in \mathit{\Gamma}, \forall x \in P, \forall y \in V, ||x - y|| < \varepsilon$ if and only if $y \in P$

If a clustering $\mathit{\Gamma}$ exists, then it is unique. There may not be a valid clustering $\mathit{\Gamma}$ for a given set V and $\varepsilon > 0$. As far as this project is concerned, one issue is that no valid clustering may exist for a given ε, and another issue is that a valid clustering may or may not accurately represent the geometric objects in question.

If additional objects are created by virtue of imprecision, it makes sense that the artifacts would be nearby, or "clustered" about the actual object. If clustering vertices, then the extra points will appear in balls, or clusters around the actual vertex. Similarly, if artificial facets are created because of imprecision, then the normal vectors will be nearby the normal vectors for the actual facet. Clustering therefore should reveal this if it occurs.

Additionally, clustering can give us some information regarding the distance, or spacing of the vertices, or the angles of the facets, of the objects if some vertices or facets are nearby eachother enough to cluster, while the others are not.

Below is the ε-clustering algorithm I used, and it is better than previous methods which might have been used in [1] since it automatically tests for stability over the full-range of possible values for ε.

Begin with $P_0 = B_\varepsilon(v_0) \cap V. \forall x, y \in P_0 \backslash v_0, \forall z \in V \backslash P_0$, if $||x - y|| > \varepsilon$ or $||x - z|| \leq \varepsilon$, then ε does not determine a valid cluster.

At step m, the index set $I_m \subset \{1, \ldots, n\}$ is defined as follows: Let $i \in I_m$, if $v_i \in V \backslash \left(\bigcup\limits_{i=0}^{m-1} P_i \right)$. Let $k = \min_<(I_m)$, and $P_m = B_\varepsilon(v_k) \cap V$. Verify: $\forall x \in P_m \backslash v_k$, $\forall y \in V \backslash P_m$, that $||x - y|| \leq \varepsilon$ if and only if $x, y \in P_m$.

Continue until V is exhausted or determine that ε does not determine a valid clustering.

The above describes the main clustering algorithm. To determine which ε's to attempt for a clustering, let $\varepsilon_{\text{min}} = \min(||x - y|| : x, y \in V)$, and similarly, $\varepsilon_{\text{max}} = \max(||x - y||)$. For a reasonable number of **increments**, test each $\varepsilon_{\text{min}} \leq \varepsilon \leq \varepsilon_{\text{max}}$, for which:

$$\varepsilon = \varepsilon_{\text{min}} + n \cdot \frac{\varepsilon \max - \varepsilon \min}{\textbf{increments}}, n \in \mathbb{N}.$$

This will reveal clusters which are stable for ranges of ε. The program I wrote (in Perl) is given in the section "coding."

3 Tensor Polytope

Definition 2 [1] Let $P \subset V$ and $Q \subset W$ be polytopes in their ambient vector-spaces. Then, the *Tensor polytope* is defined as follows:
$P \otimes Q := \mathbf{conv}(\{(x \otimes y, x, y) | x \in P, y \in Q\}) \subset (V \otimes W) \oplus V \oplus W$.

Proposition 1 $\#\mathbb{F}(\varDelta_n \otimes P) = (n+1) \cdot \#\mathbb{F}(P)$.[1]

Proof Let $P^{\mathbf{join}^2} := \mathbf{join}(P, P)$, and $P^{\mathbf{join}^{n+1}} := \mathbf{join}(P^{\mathbf{join}^n}, P)$.

It is clear that $\varDelta_n \cong \{*\}^{\mathbf{join}^{n+1}}$, where $*$ is a single point.

From [1], we know that $P \otimes \mathbf{join}(Q, R) \cong \mathbf{join}(P \otimes Q, P \otimes R)$.

Then, one sees that $\varDelta_n \otimes P \cong P \otimes \{*\}^{\mathbf{join}^{n+1}} \cong (P \otimes \{*\})^{\mathbf{join}^{n+1}} \cong P^{\mathbf{join}^{n+1}}$.

Observe that $\mathbf{join}(P, Q)$ has $\#\mathbb{F}(P) + \#\mathbb{F}(Q)$ facets. Assume that:

$\#\mathbb{F}(P^{\mathbf{join}^n}) = n \cdot \#\mathbb{F}(P)$. Then $P^{\mathbf{join}^{n+1}} = \mathbf{join}(P^{\mathbf{join}^n}, P)$, which has $n \cdot \#\mathbb{F}(P) + \#\mathbb{F}(P) = (n+1) \cdot \#\mathbb{F}(P)$ facets. □

Corollary 1 $\mathbb{R}^2 \otimes \mathbb{R}^2 \cong \mathbb{R}^4$ *in the vector-sense by* $(a, b) \otimes (c, d) \mapsto (ac, ad, bc, bd)$.[2]

3.1 Results

The following data was acquired using an approximation of P_m, P_n. Then the normal vectors for the facets of $P_m \otimes P_n$ were calculated in Polymake [3]. The vectors were all normalized, and then the clusters were analyzed. Since $\varepsilon_{\min} > 10^{-4}$ is fairly large, this indicates that it is likely that the number Polymake returns is accurate for the number of facets. If a stable clustering occurs nearby $\varepsilon_{\min}$ then this indicates that the cluster is more likely to be meaningful than a stable clustering that is far-distant. It is also possible (unlikely) that the count is an underestimate.

The results of this section are summarized in the following table. Note the symmetry of the table indicated by the slashes (The results in italics come from experimental results, whereas the results in normal type are from theoretical results):

[1] This conjecture was given to me as “definitely true,” but without proof, by Prof. Gubeladze.

[2] This is a restriction of the well-known fact from Algebra.

$\otimes$	3	4	5	6	7	8	9	10
3	9	12	15	18	21	24	27	30
4	/	*24*	*60*	*144*	*308*	*592*	*1044*	*1720*
5	/	/	*135*	*630*	*1295*	*2440*	*4185*	*4470*
6	/	/	/	*552*	*3738*	*6000*	*9666*	*15900*
7	/	/	/	/	*2415*	*14840*	*23625*	*36433*
8	/	/	/	/	/	*8656*	*45576*	*63680*
9	/	/	/	/	/	/	*30231*	*115110*
10	/	/	/	/	/	/	/	*59520*

Table 1 Polymake expected values for $\mathbb{F}(P \otimes Q)$

All of the data gathered on the clusters for the $P \otimes Q$ objects is presented below. Note a count of zero facets indicate that there were points outside the ball that should be inside, or that there are points within the ball that are further than ε away from one another, i.e. not a true clustering for those epsilons. $P_n \otimes P_{10}$ was too large to cluster in the time that I had for $n = 8, 9, 10$. $P_8 \otimes P_9$ took about one month to perform a cluster-analysis with the use of five separate machines.

	$P_8 \otimes P_{10}$	$P_9 \otimes P_{10}$	$P_{10} \otimes P_{10}$
$\varepsilon_{\min}$	.00138732359204731	.000797084921839763	.00542680509773397

$P_5 \otimes P_6$	ε−min=0.0383328864892567 ε-max=1.68205636874513	Step-size: 0.00164372348225587
#Facets	ε-range	#steps
630		
622	0.0383328864892567	1
570	0.0399766099715126 to 0.0416203334537685	2
510	0.0432640569360244 to 0.140243742389121	60

$P_5 \otimes P_7$	ε−min= 0.0407936726679516 ε-max=1.64713823746133	Step-size: 0.00160634456479338
#Facets	ε-range	#steps
1295		
1291	0.0407936726679516	1
1155	0.042400017232745 to 0.0456127063623318	3

$P_5 \otimes P_8$	ε-min: 0.0450047826869534 ε-max=1.63781178762196	Step-size: 0.00318561400987001
#Facets	ε-range	#steps
2440		
2439	0.0450047826869534	1
2280	0.0481903966968234 to 0.0513760107066935	2
0	0.0545616247165635 to 0.0800465367955236	9
1960	0.0832321508053936 to 0.0959746068448736	5

$P_5 \otimes P_9$	ε-min=0.0301225372572608 ε-max=1.64816724197138	Step-size: 0.00161804470471412
#Facets	ε-range	#steps
4185		
4181	0.0301225372572608	1
4005	0.0317405819619749 to 0.0349766713714032	3

$P_6 \otimes P_8$	ε-min=0.0163436463610595 ε-max=1.65276163564236	Step-size: 0.0016364179892813
#Facets	ε-range	#steps
6000		
5968	0.0163436463610595	1
5808	0.0179800643503408 to 0.0228893183181847	4
0	0.024525736307466 to 0.0539812601145295	19
4368	0.0556176781038108	1

$P_6 \otimes P_9$	ε-min=0.00711292873183818 ε-max=1.62442465524836	Step-size: 0.00161731172651652
#Facets	ε-range	#steps
9666		
9658	0.00711292873183818	1
9450	0.0087302404583547 to 0.0103475521848712	2

$P_6 \otimes P_{10}$	ε-min= 0.017650476323911 ε-max=1.64333514750996	Step-size: 0.000406421167796512
#Facets	ε-range	#steps
15900		
15884	0.017650476323911	1
15660	0.0180568974917075 to 0.018463318659504	2
15420	0.0188697398273005	1
15300	0.019276160995097 to 0.0204954244984866	4

$P_7 \otimes P_7$	ε-min=0.0891378230057438 ε-max=1.6459131980079	Step-size: 0.00155677537500216
#Facets	ε-range	#steps
2415		
2407	0.0891378230057438	1
2219	0.0906945983807459 to 0.103148801380763	9
2023	0.104705576755765 to 0.120273330505787	11

$P_7 \otimes P_8$	ε-min=0.003094540237951 ε-max=1.65834000577096	Step-size: 0.000662098186213202
#Facets	ε-range	#steps
14840		
14838	0.003094540237951	
14728	0.0037566384241642 to 0.0057429329828038l	4
14504	0.00640503116901702	1
14280	0.00706712935523022 to 0.00772922754144342	2

$P_7 \otimes P_9$	ε-min=0.00464412345966292 ε-max=1.64349512595472	Step-size: 0.000409712750623765
#Facets	ε-range	#steps
23625		
23621	0.00464412345966292	1
0	0.00505383621028668 to 0.00546354896091045	2
23121	0.00587326171153421 to 0.00669268721278174	3

$P_7 \otimes P_{10}$	ε-min=0.00032227625071186 ε-max=1.63597432199261	Step-size: 0.0016356520457419
#Facets	ε-range	#steps
36433		
36425	0.00032227625071186	
35593	0.00195792829645376	1
35313	0.00359358034219565	1
34476	0.00522923238793755	1

$P_8 \otimes P_8$	ε-min=0.0434146529645682 ε-max=1.67437215161989	Step-size: 0.00081547874932766
#Facets	ε-range	#steps
8656		
8592	0.0434146529645682	
8400	0.0442301317138958 to 0.0458610892125511	3
8144	0.0466765679618788 to 0.0491230042098618	4

$P_8 \otimes P_9$	ε-min=0.0010008709135947 ε-max=1.6490067843067	Step-size: 0.00082400295669655
#Facets	ε-range	#steps
45576		
45560	0.0010008709135947	
44856	0.00182487387029125 to 0.0026488768269878	2
44136	0.00347287978368435	1
43848	0.0042968827403809	1
0	0.00512088569707745	1
42984	0.005944888653774	1

$P_9 \otimes P_9$	ε-min=0.00433288203872405 ε-max=1.64100254203561	Step-size: 0.00163666965999689
#Facets	ε-range	#steps
30231		
30227	0.00433288203872405	
28935	0.00596955169872094	1
28287	0.00760622135871783 to 0.0108795606787116	3

4 Hom-polytope

Definition 3 A map $f : P \to Q$ is *affine* if it preserves barycentric coordinates:

$$f\left(\sum_{i=1}^{n} \lambda_i \mathbf{x}_i\right) = \sum_{i=1}^{n} \lambda_i f(\mathbf{x}_i) : \sum_{i=1}^{n} \lambda_i = 1, \lambda_i \in \mathbb{R}, \mathbf{x}_i \in P.$$

The *Hom-polytope* is $\mathbf{Hom}(P, Q) := \{f : P \to Q | f \text{ is affine}\}$.

Also, let $\mathbf{aff}(P, Q) := \{f : \mathbf{aff}(P) \to \mathbf{aff}(Q) | f \text{ is affine}\}$

It is not difficult to see that for vector spaces V and W, $\mathbf{aff}(V, W)$ is a vector-space. For affine spaces $\mathbb{A} \subset V$ and $\mathbb{B} \subset W$, one sees that $\mathbf{aff}(\mathbb{A}, \mathbb{B})$ is an affine subspace of $\mathbf{aff}(V, W)$, whose dimension is $\dim(\mathbf{aff}(\mathbb{A}, \mathbb{B})) = \dim \mathbb{A} \cdot \dim \mathbb{B} + \dim B$.

Proof Let $\mathbb{A} = U + x_0$ and $\mathbb{B} = W' + y_0$, where U, W' are subspaces of V and W respecitvely.

Let $\pi_U : V \to U$ be the linear projection mapping, and $\pi = \pi_U + x_0 : V \to \mathbb{A}$ be the affine projection (Notice that $\pi^2 = \pi$). Also let $\iota : \mathbb{B} \to W$ be the identity embedding. Then the embedding $\mathbf{aff}(\mathbb{A}, \mathbb{B}) \to \mathbf{aff}(V, W)$ defined by $f \mapsto \iota \circ f \circ \pi$ makes $\mathbf{aff}(\mathbb{A}, \mathbb{B})$ into an affine subspace of $\mathbf{aff}(V, W)$ [1].

Let $\mathcal{L}(U, W')$ be the set of linear maps $U \to W'$, and $\mathcal{B}$ a basis of $\mathcal{L}(U, W')$. Let $\mathcal{C}$ be a basis for W'. Then, let Ω be the vector space generated by $\mathcal{B} \cup \mathcal{C}$. We want to show that $\Omega + y_0 = \mathbf{aff}(\mathbb{A}, \mathbb{B})$

$(\subseteq)$: Let $f(u) + (w + y_0) \in \Omega + y_0$, f extends to a linear map $V \to W$, and $w + y_0 \in W$, hence $f(u) + (w + y_0) \in \mathbf{aff}(\mathbb{A}, \mathbb{B})$.

$(\supseteq)$: Let $f(x) + b \in \mathbf{aff}(\mathbb{A}, \mathbb{B})$. Then $f(x) + b$ is the restriction of an affine map $V \to W$, $f(x) + b = f(u + x_0) + b = f(u) + (f(x_0) + b)$. Since $f(0) = 0$, $f(x_0) + b \in \mathbb{B}$, hence $f(x) + b \in \Omega + y_0$.

Recall that $\dim \mathcal{L}(U, W) = \dim U \cdot \dim W = \dim \mathbb{A} \cdot \dim \mathbb{B}$.

Then $\#(\mathcal{B} \cup \mathcal{C}) = \dim \mathbb{A} \cdot \dim \mathbb{B} + \dim \mathbb{B}$, as desired.

In particular, $\dim(\mathbf{aff}(P, Q)) = \dim P \cdot \dim Q + \dim Q$.

Theorem 2 *[1]* $\boldsymbol{Hom}(P, Q)$ *is a polytope in* $\boldsymbol{aff}(P, Q)$*, the facets are described by:*

$$\mathbb{F}(\boldsymbol{Hom}(P, Q)) = \{H(v, F) | v \in \boldsymbol{vert}(P), F \in \mathbb{F}(Q), H(v, F) = \{f \in \boldsymbol{Hom}(P, Q) | f(v) \in F\}\}.$$

Also, $\dim \boldsymbol{Hom}(P, Q) = \dim P \dim Q + \dim Q$*.*

Proof $\forall F \in \mathbb{F}(Q), \exists \varphi_F \in \mathbf{Hom}(Q, \mathbb{R}_+)$ such that $\ker \varphi_F = F$ (see Figure 1) [1].

Let $\varphi_{F,v} : \mathbf{aff}(P, Q) \to \mathbb{R}$ defined by $f \mapsto (\varphi_F \circ f)(v), \forall v \in \mathbf{vert}(P)$ [1].

It is not difficult to see that $\varphi_{F,v}$ is affine.

We begin by showing that $\mathbf{Hom}(P, Q) = \bigcap_{F,v} \{f \in \mathbf{aff}(P, Q) | \varphi_{F,v}(f) \geq 0\}$.

$\subseteq$: Let $f \in \mathbf{Hom}(P, Q), F \in \mathbb{F}(Q), v \in \mathbf{vert}(P)$. If $f(v) \in F$, then $\varphi_{F,v}(f) = \varphi_F(f(v)) = 0$. If $f(v) \in Q \backslash F$, then $\varphi_{F,v}(f) > 0$.

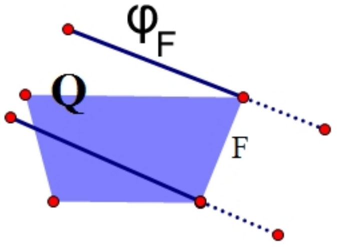


Fig. 1 Example construction of φ_F

$\supseteq$: Let $f \in \mathbf{aff}(P,Q)$ such that $\forall F \in \mathbb{F}(Q), \forall v \in \mathbf{vert}(P), \varphi_{F,v}(f) \geq 0$. Assume by way of contradiction that $\exists v \in \mathbf{vert}(P)$ such that $f(v) \notin Q$ (see Figure 2).

Let $y \in \mathbf{Interior}(Q)$. Then the line-segment $[f(v), y]$ passes through a facet F of Q. φ_F is affine, non-negative on $[f(v), y] \cap Q := [y', y]$, and zero on $[f(v), y] \cap F$. Then, φ_F is negative on $[f(v), y')$, and since $0 > \varphi_F(f(v)) = \varphi_{F,v}(f)$, this is a contradiction.

Since all of $\mathbf{vert}(P)$ maps into Q by f, we have $f(P) \subset Q$, and $f \in \mathbf{Hom}(P,Q)$.

Next, since for a support hyperplane H, $H \cap P$ is a facet of P if and only if it is not contained in any other proper face of P, it remains to show that for each particular (F, v), there is an element unique to that facet.

Let $P \overset{\text{onto}}{\to} [0,1]$ be a mapping such that $v \mapsto 0$ and $P \backslash v \overset{\text{onto}}{\to} (0,1]$.

Also, let $[0,1] \overset{\text{embeds}}{\to} Q$ such that $0 \to \mathbf{Interior}(F)$ and $(0,1] \to \mathbf{Interior}(Q)$.

Let f be the composition of these two maps. Then $f : P \to Q$ such that $f(v) \in \mathbf{Interior}(F)$ and $f(w) \in \mathbf{Interior}(Q)$ $\forall w \in \mathbf{vert}(P) \backslash v$.

Hence, $f \in \ker \varphi_{F,v} \backslash \bigcup_{\mathbb{F}(Q)\backslash F, \mathbf{vert}(P)\backslash v} \ker \varphi_{G,w}$ [1].

To see that $\dim(\mathbf{Hom}(P,Q)) = \dim P \dim Q + \dim Q$, $\mathbf{Interior}(\mathbf{Hom}(P,Q)) = \bigcap_{F,v} \{f \in \mathbf{aff}(P,Q) | \varphi_{F,v}(f) > 0\} \neq \emptyset$ and is open. Hence $\mathbf{Hom}(P,Q)$ is full-dimensional [1], and $\dim(\mathbf{Hom}(P,Q)) = \dim(\mathbf{aff}(P,Q)) = \dim P \dim Q + \dim Q$.

4.1 Results

I was able to reproduce the vertex quantities found in [1] for $n, m = 3, \ldots, 8$, as well as extend them to $n, m \leq 9, 10$. In [1], the authors find stability for $10^{-3} \leq \varepsilon \leq 10^{-4}$. However, in my results, the $\varepsilon_{\min}$ was invariably greater than 10^{-2} for $n, m \leq 8$, meaning the points were spaced further apart than the clusters found in [1]. It is unclear if my method exactly mirrors the method that

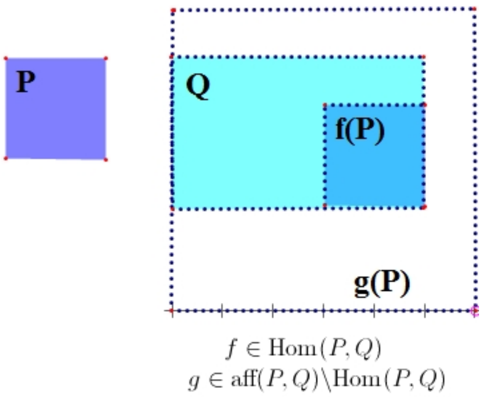

P
Q
f(P)
g(P)
$f \in \mathrm{Hom}(P, Q)$
$g \in \mathrm{aff}(P, Q) \backslash \mathrm{Hom}(P, Q)$

Fig. 2 Some examples of affine images.

was used there. It is possible that stability was found because there was only one point within each cluster, a sort of false-positive result. It is also possible that the rational approximations I used are simply more precise, and so the objects were not viewed as being distinct when the polygons were placed into polymake, and the resulting vertices were placed into the matrix. Regardless, it is clear that the results presented here represent a marked improvement when compared with the original results.

$\mathbf{Hom}(P_m, P_n)$	$n = 3$	4	5	6	7	8	9	10
$m = 3$	27	64	125	216	343	512	729	1000
4	15	36	125	138	371	360	801	750
5	63	144	*165*	*696*	*987*	*1408*	*1809*	*1480*
6	33	64	*305*	*180*	*1141*	*1040*	*1665*	*1960*
7	129	256	*915*	*1308*	*1001*	*3312*	*1336*	*3114*
8	75	100	*485*	*414*	*2079*	*1144*	*1502*	*1951*
9	237	400	*1355*	*1824*	*3913*	*1535*	*1590*	*1787*
10	153	144	*525*	*756*	*2597*	*3248*	*7389*	*3080*

Table 2 Polymake expected values for $\mathbf{vert}(\mathbf{Hom}(P, Q))$

The vertices are then clustered using the clustering algorithm, and results are presented below:

$\mathbf{Hom}(P_4, P_8)$	ε−min=0.507305936153988 ε-max=2.8284271248	Step-size: 0.00232112118864601
#Vertices	ε-range	#steps
360		
328	0.507305936153988	1
296	0.509627057342634 to 0.583902935379307	32
264	0.586224056567953 to 0.662821055793272	33

$\mathbf{Hom}(P_4, P_9)$	ε−min=0.20573706381088 ε-max=2.78545696138793	Step-size:0.00257971989757705
#Vertices	ε-range	#steps
801	0.203157343913303	1
785	0.20573706381088	1
657	0.208316783708457 to 0.252172021967267	17

$\mathbf{Hom}(P_4, P_{10})$	ε−min=0.330792269092958 ε-max=2.82842712481603	Step-size:0.00499526971144614
#Vertices	ε-range	#steps
750	0.325796999381512	1
734	0.330792269092958	1
670	0.335787538804405 to 0.38074496620742	9
630	0.385740235918866 to 0.530603057550804	29

$\mathbf{Hom}(P_5, P_9)$	ε−min=0.0340145455377996 ε-max=2.70873731833382	Step-size: 0.00534944554559204
#Vertices	ε-range	#steps
1809	0.0286650999922076	1
1805	0.0340145455377996	1
0	0.0393639910833916 to 0.0982078920849 04	11
1269	0.103557337630496 to 0.194497911905561	17

$\mathbf{Hom}(P_6, P_7)$	ε−min=0.01519146347549 ε-max=2.65792796852489	Step-size: 0.0026427365050494
#Vertices	ε-range	#steps
1141		
1133	0.01519146347549	1
1057	0.0178341999805394 to 0.0733316665865768	21
973	0.0759744030916262 to 0.0997590316370708	10

$\mathbf{Hom}(P_6, P_8)$	$\varepsilon-$min=0.171213249503796 ε-max=2.71163385580301	Step-size: 0.00254042060629921
#Vertices	ε-range	#steps
1040		
1036	0.171213249503796	1
0	0.173753670110095 to 0.239804605873875	26
752	0.242345026480174 to 0.249966288299072	4

$\mathbf{Hom}(P_7, P_6)$	$\varepsilon-$min=0.0263622373165557 ε-max=2.53431645299447	Step-size: 0.0100318168627117
#Vertices	ε-range	#steps
1308	0.016330420453844	
3300	0.0263622373165557	
1224	0.0363940541792674 to 0.0865531384928256	6

$\mathbf{Hom}(P_7, P_8)$	$\varepsilon-$min=0.0135340558993126 ε-max=2.69453400350058	Step-size: 0.00268099994760127
#Vertices	ε-range	#steps
3312		
3308	0.0135340558993126	1
0	0.0162150558469138 to 0.0242580556897176	4
2864	0.0269390556373189 to 0.0644730549037367	14

$\mathbf{Hom}(P_7, P_{10})$	$\varepsilon-$min=0.010596164571308 ε-max=2.70759126132689	Step-size: 0.00539399019351116
#Vertices	ε-range	#steps
3114	0.00520217437779688	
3113	0.010596164571308	
0	0.0159901547648192 to 0.0321721253453527	4
2634	0.0375661155388638 to 0.075324046893442	8
2574	0.0807180370869531 to 0.0861120272804643	2

$\mathbf{Hom}(P_8, P_7)$	$\varepsilon-$min=0.0179855033449858 ε-max=2.61356423524859	Step-size: 0.0012977893659518
#Vertices	ε-range	#steps
2079		
2063	0.0179855033449858	1
1967	0.0192832927109376 to 0.0218788714428413	3
1855	0.0231766608087931 to 0.0244744501747449	2
0	0.0257722395406967 to 0.0452390800299737	15
1631	0.0465368693959255 to 0.0543236055916364	7

$\mathbf{Hom}(P_8, P_9)$	$\varepsilon-$min=0.0166180319420024 ε-max=2.71893208974682	Step-size: 0.00540462811560963
#Vertices	ε-range	#steps
1502	0.0112134038263927	
1486	0.0166180319420024	1
1470	0.022022660057612	1
1454	0.0274272881732216 to 0.0436411725200505	4

$\mathbf{Hom}(P_8, P_{10})$	$\varepsilon-$min=0.0645885690432305 ε-max=2.7393655462301	Step-size: 0.00534955395437373
#Vertices	ε-range	#steps
1951	0.0592390150888568	
1949	0.0645885690432305	1
0	0.0699381229976043 to 0.075287676951978	2
1855	0.0806372309063517	1

$\mathbf{Hom}(P_9, P_3)$	$\varepsilon-$min=0.195210464007939 ε-max=1.78671407978536	Step-size: 0.00318300723155485
#Vertices	ε-range	#steps
237	0.192027456776384	
233	0.195210464007939	1
183	0.198393471239493 to 0.312981731575468	37

$\mathbf{Hom}(P_9, P_6)$	$\varepsilon-$min=0.0941939428330602 ε-max=2.48727707044292	Step-size: 0.00478616625521972
#Vertices	ε-range	#steps
1824	0.0894077765778405	
1816	0.0941939428330602	1
1716	0.09898010908828 to 0.122910940364379	6
0	0.127697106619598 to 0.161200270406136	8
1644	0.165986436661356 to 0.180344935427015	4

$\mathbf{Hom}(P_9, P_{10})$	$\varepsilon-$min=0.0941939428330602 ε-max=2.48727707044292	Step-size: 0.00478616625521972
#Vertices	ε-range	#steps
1787	0.0377804871635288	
1783	0.0404618235078826	
0	0.0431431598522364 to 0.0619125142627131	8
1607	0.0645938506070669 to 0.0672751869514207	2

$\mathbf{Hom}(P_{10}, P_7)$	$\varepsilon-$min=0.0404883634274083 ε-max=2.56518854471618	Step-size: 0.00504940036257755
#Vertices	ε-range	#steps
2597	0.0354389630648307	
2589	0.0404883634274083	
2457	0.0455377637899858 to 0.065735365240296	5

$\mathbf{Hom}(P_{10}, P_9)$	$\varepsilon-$min=0.00414489157101909 ε-max=2.6996916909301	Step-size: 0.00134777339967954
#Vertices	ε-range	#steps
7389	0.00279711817133955	
7381	0.00414489157101909	
7209	0.00549266497069863 to 0.00684043837037816	2
6849	0.0081882117700577 to 0.0122315319690963	4

5 The Hypercube Matrix

5.1 Definitions

Let $n \in \mathbb{N}$ be a natural number and let I be the interval $[1, n]$. Define the hypercube by the Cartesian product of m of these intervals:

Hypercube$= I \times \cdots \times I = \prod_{j=1}^{m} I$.

Define the hypercube matrix by introducing a lattice-structure **Hypercube** $\bigcap \mathbb{Z}^m$, and label each of the entry-coordinates $(\alpha_1 \ldots \alpha_m)$, where $\alpha_j = 1, \ldots, n$ for $j = 1, \ldots, m$. Each coordinate entry will have a value $A(\alpha_1 \ldots \alpha_m)$, often $A(\alpha_1 \ldots \alpha_m) \in \mathbb{R}$ is real, but it could also be abstracted to complex or field etc. This matrix is denoted $[\![A(\alpha_1 \ldots \alpha_m)]\!]$, or equivalently, $[\![A(\alpha_1 \ldots \alpha_m)]\!]_{\alpha_1 \ldots \alpha_m}$, when the index is not immediately clear.

One multiplies by an n-vector in the j-th direction:

$$[\![A(\alpha_1 \ldots \alpha_m)]\!] \otimes_j [u^1 u^2 \ldots u^n] = \left[\!\!\left[\sum_{\alpha j} A(\alpha_1 \ldots \alpha_m) \cdot u^{\alpha_j} \right]\!\!\right]_{\alpha_1 \ldots \overline{\alpha_j} \ldots \alpha_m}$$

Here the overline on the α_j indicates that the jth direction has collapsed, that is, the resulting index does not have a j-th coordinate.

You can also take the abstract analogue of a rectangular-prism. Let the intervals: $I_1 = [1, n_1], I_2 = [1, n_2], \ldots, I_m = [1, n_m]$, where $n_i \in \mathbb{N}$ are natural numbers for $i = 1, 2, \ldots, m$. Define the prism polytope by the Cartesian product of these m intervals:

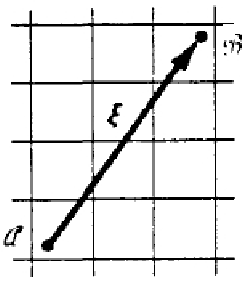
$$\textbf{Prism} = I_1 \times I_2 \times \cdots \times I_m = \prod_{j=1}^{m} I_j$$

Let $n = \max(n_1, n_2, \ldots, n_m)$, and construct the Hyper-matrix (h-matrix) by creating a lattice **Prism** $\bigcap \mathbb{Z}^n$, and labeling each entry $(\alpha_1 \alpha_2 \ldots \alpha_m)$ for $\alpha_j = 1, 2, \ldots n_j$ and $j = 1, 2, \ldots, m$. The result will be an $n_1 \times n_2 \times \cdots \times n_m$ matrix!

H-matrix multiplication can be defined $\otimes(i, j)$ for h-matrices, provided they match along certain dimensions. I am leaving this remaining work to the reader.

5.2 Motivating Example

The Metric Tensor of Box 1.3.IV.B of [6], this result is seen in elementary Linear algebra because it can be represented by a traditional matrix. In usual Euclidean coordinates, start with a separation vector ξ pointing from point $\mathcal{A}$ to $\mathcal{B}$.



Then the separation coordinates are:

$$\xi^1 = x^1(\mathcal{B}) - x^1(\mathcal{A})$$
$$\xi^2 = x^2(\mathcal{B}) - x^2(\mathcal{A})$$

As we know from Pythagoras, the metric traditionally used to measure the length of ξ in Euclidean coordinates is $|\xi|^2 = (\xi^1)^2 + (\xi^2)^2 = \xi \cdot \xi$ (the dot-product). In matrices, this is:

$[\xi^1\ \xi^2] \cdot \begin{bmatrix} 1\ 0 \\ 0\ 1 \end{bmatrix} \cdot \begin{bmatrix} \xi^1 \\ \xi^2 \end{bmatrix}$

The Kronecker-Delta: $\delta_{\alpha\beta} = \begin{cases} 1 & \alpha = \beta \\ 0 & \alpha \neq \beta \end{cases}$

So, in linear-equation, the above metric tensor is:

$$\xi \cdot \xi = \sum_{\alpha,\beta=0,1} \delta_{\alpha\beta}\xi^\alpha\xi^\beta$$

In localized Lorentz space-time coordinates (x^0, x^1, x^2, x^3), where x^0 is the time coordinate, and the X^1, X^2, X^3 are space coordinates, the distance metric is given: $\langle\xi, \xi\rangle = -(\xi^0)^2 + (\xi^1)^2 + (\xi^2)^2 + (\xi^3)^2$. In matrices:

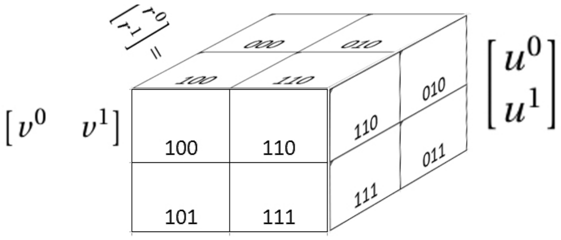
$$[\xi^0\ \xi^1\ \xi^2\ \xi^3] \cdot \begin{bmatrix} -1\ 0\ 0\ 0 \\ 0\ \ 1\ 0\ 0 \\ 0\ \ 0\ 1\ 0 \\ 0\ \ 0\ 0\ 1 \end{bmatrix} \cdot \begin{bmatrix} \xi^0 \\ \xi^1 \\ \xi^2 \\ \xi^3 \end{bmatrix}$$

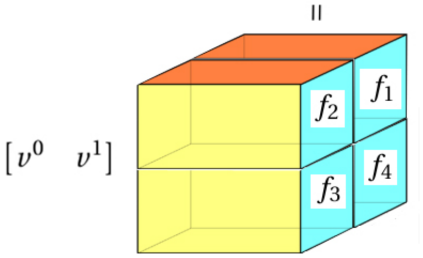
Here $\eta_{\alpha\beta} = \begin{cases} -1 & \alpha = 0 = \beta \\ \delta_{\alpha\beta} & \text{otherwise} \end{cases}$, and the index $I = \{\alpha, \beta = 0, 1, 2, 3\}$, the linear equation for the Lorentz metric tensor is:

$$\langle\xi, \xi\rangle = \sum_I \eta_{\alpha\beta}\xi^\alpha\xi^\beta$$

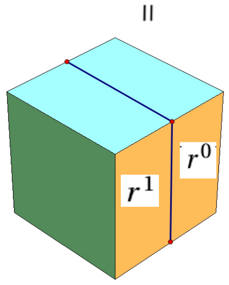
5.3 Example: Basic cube-matrix

In this example, I use the cube-matrix the represent a Tensor with two vector inputs, and a two-dimensional output: $\mathbf{g}(\mathbf{u},\mathbf{v}) = [r^0, r^1]$. This matrix has three axis, α, β, γ, and its entries are represented equivalently by $A_{\alpha\beta\gamma} = \alpha\beta\gamma = (\alpha\beta\gamma)$. To keep this motivating example simple, I will be using the index $I = \{\alpha, \beta, \gamma = 0, 1\}$, one can see that this example can be extended to a larger index (depths by columns by rows).





$$f_1 = (000)\,u^0 + (010)\,u^1$$
$$f_2 = (100)\,u^0 + (110)\,u^1$$
$$f_3 = (101)\,u^0 + (111)\,u^1$$
$$f_4 = (001)\,u^0 + (011)\,u^1$$



$$r_0 = (000)u^0v^0 + (010)u^1v^0 + (001)u^0v^1 + (011)u^1v^1$$
$$r_1 = (100)u^0v^0 + (110)u^1v^0 + (101)u^0v^1 + (111)u^1v^1$$

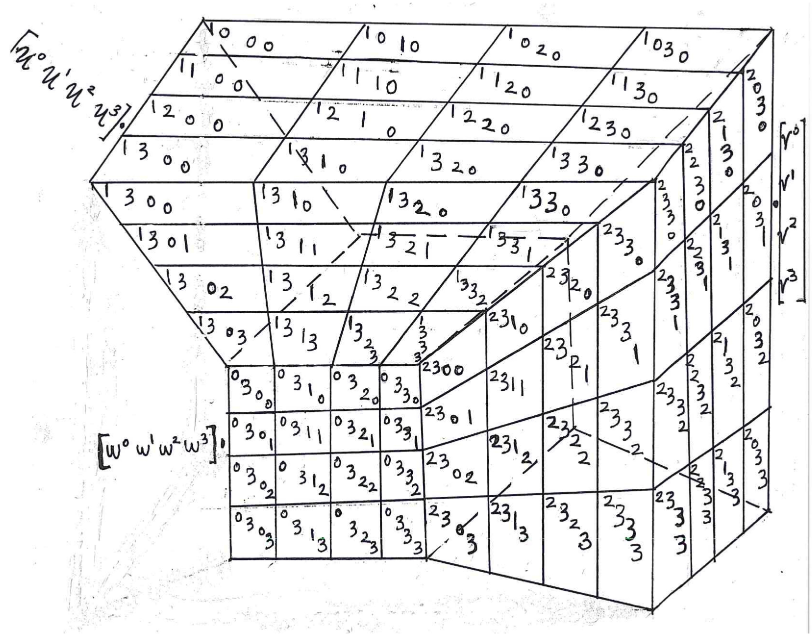
In bi-linear-equation form this is $\mathbf{g}(\mathbf{u},\mathbf{v}) = [r^0, r^1]$, and $r^\alpha = \sum\limits_{\beta,\gamma} A_{\alpha\beta\gamma}u^\beta v^\gamma$.

5.4 Hypercube matrix applied to The Riemann Tensor

As can be seen in equation (1.10) of §1.6 of [6], let the index $I = \{\alpha, \beta, \gamma, \delta = 0, 1, 2, 3\}$. Then, there is the Riemann Tensor, which has three 4-vectors as inputs, a 4-vector as outputs, and can be represented as trilinear functions:

$$\mathbf{r} = \mathbf{Riemann(u,v,w)} \Leftrightarrow r^\alpha = \sum_{\beta,\gamma,\delta} R^\alpha_{\beta\gamma\delta}u^\beta v^\gamma w^\delta$$

The 4-cube matrix I have constructed to represent this equation has four axis, with each entry $R^\alpha_{\beta\gamma\delta} = (\alpha\beta\gamma\delta)$. The cube matrix meant to represent r^3 I have chosen to put on the back, so all of its entries are obscured. Also, many of the entries on the other three cubes are obscured. There are a total of $4 \times 4 \times 4 \times 4 = 256$ entries, meant to describe the curvature of space-time. Vector $\mathbf{u}$ is multiplied in the β direction, $\mathbf{v}$ in the γ direction, and $\mathbf{w}$ in the δ direction. The α direction is the only one of the four that does not collapse, and the result is an output of a 4-vector $\mathbf{r}$. Each component of this result is represented trilinearly by the expression in equation (1.10).



6 coding

My method for sections 3 and 4 is as follows: First run these scripts within Polymake [3]. The first produces approximations for the Tensor-polytope, the second produces approximations for the Hom-polytope.

```
use application "polytope";

# Takes user input for which polygons to Tensor.

use Math::Complex; use Math::Trig;

print "Will now calculate Tensor(Pn,Pm):\n";

declare $count=1;

declare $n;
declare $m;
print "Enter n=";
chomp($n=<>);
print "Enter m=";
chomp($m=<>);

print "Now calculating Tensor(P$n,P$m)\n";

# Calculates vertices of the Polygon.

sub poly {
declare $zeta=Math::Complex->new(cos 2*pi/$_[0],sin 2*pi/$_[0]);
declare $zeta2=$zeta;

declare @polygon=(1);
for (declare $i=1;$i<$_[0];$i++){
push(@polygon,$zeta2);
$zeta2=$zeta2*$zeta;
}
print "Polygon $count=conv(@polygon)\n";
return(@polygon);
}

declare @poly1=poly($n);
$count++;
declare @poly2=poly($m);

# Calculates vert(PnPm)={(vw,v,w)|vvert(Pn),wvert(Pm)} under the identification R^2R^2R^4 by (a,b)(c
# Also note that for some reason polymake needs an extra dimension, which accounts for the 1 as the
```

```
declare @vertices;
declare $v=new Vector<Float>([1,0,0,0,0,0,0,0,0]);

for (declare $j=0;$j<$n;$j++){
for ($i=0;$i<$m;$i++){
$v=[1,Re($poly1[$j])*Re($poly2[$i]),Re($poly1[$j])*Im($poly2[$i]),Im($poly1[$j])*Re($poly2[$i]),Im($
push(@vertices,$v);
}
}

# Now put the vertices into a matrix

declare $matrix=new Matrix<Float>(\@vertices);

# Now create the polytope object PnPm, and look at its facets
# P10P10 takes about half an hour and returns 59520 facets before clustering.

declare $p=new Polytope<Float>(POINTS=>$matrix);
print "Number of facets before applying the theoretical epsilon-clustering method: ";
print $p->N_FACETS . "\n";

$matrix=$p->FACETS;

open FILE, ">matrix.txt" or die $!;
print FILE $matrix;
close FILE;

use application "polytope";

# Takes user input for which polygons to take the Hom.

use Math::Complex; use Math::Trig;

print "Will now calculate Hom(Pn,Pm):\n";

declare $count=1;

declare $n;
declare $m;
print "Enter n=";
chomp($n=<>);
print "Enter m=";
chomp($m=<>);

print "Now calculating Hom(P$n,P$m)\n";
```

```
# Calculates vertices of the Polygon in Complex.

sub poly {
declare $zeta=Math::Complex->new(cos 2*pi/$_[0],sin 2*pi/$_[0]);
declare $zeta2=$zeta;

declare @polygon=(1);
for (my $i=1;$i<$_[0];$i++){
push(@polygon,$zeta2);
$zeta2=$zeta2*$zeta;
}
print "Polygon $count=conv(@polygon)\n";
return(@polygon);
}

declare @poly1=poly($n);
$count++;
declare @poly2=poly($m);

declare $v=new Vector <Float>(1,0,0);
declare @array;
for (my $i=0;$i<$n;$i++) {
$v=new Vector <Float>(1,Re($poly1[$i]),Im($poly1[$i]));
push(@array,$v);
}

declare $matrix=new Matrix <Float>(\@array);

declare $p=new Polytope <Float> (POINTS=>$matrix);

@array=();

for (my $i=0;$i<$m;$i++) {
$v=new Vector <Float>(1,Re($poly2[$i]),Im($poly2[$i]));
push(@array,$v);
}

$matrix=new Matrix <Float>(\@array);

declare $q=new Polytope <Float> (POINTS=>$matrix);

declare $r=mapping_polytope($p,$q);

$p=new Polytope <Float> ($r);
$matrix=$p->VERTICES;
```

```
open FILE, ">Hom.txt" or die $!;
print FILE $matrix;
close FILE;

# apply clustering algorithm
```

The next step is to normalize the output vectors from Polymake:

```
#!/usr/bin/perl

#puts the text file into a matrix

open FILE, "matrix.txt" or die $!;

while (defined (my $line = <FILE>)) {
   my @tmp = split(' ', $line);
   push(@facet, [@tmp]); }

close FILE;

#normalizes the vectors

use Math::Orthonormalize qw(:all);

$n=scalar(@facet);

for (my $i=0;$i<$n;$i++) {
my $vector=$facet[$i];
$vector=normalize($vector);
my @tmp=@$vector;
push(@normalized,[@tmp]);
}

# saves normalized matrix to text

open FILE, ">normalized.txt" or die "cannot create normalized";

for (my $j=0;$j<$n;$j++) {
for (my $i=0;$i<9;$i++){
print FILE $normalized[$j][$i] . "\t";
}
print FILE "\n";
}
close FILE;
```

Finally, the main clustering algorithm is given below, and the data is collected:

```
#!/usr/bin/perl


open FILE, "normalized.txt" or die $!;


while (defined (my $line = <FILE>)) {
   my @tmp = split(' ', $line);
   push(@facet, [@tmp]); }

close FILE;

print "*************************************\n";
print "Welcome to Epsilon-Cluster\n";
print "By Andrew Dynneson\n";
print "*************************************\n";

print "\n\nEnter number of dimensions:\n";
$d=<>;

$n=scalar(@facet); # number of vectors

sub distance {
my $x=$_[0];
my $y=$_[1];
my $distance=0;
for (my $i=0;$i<$d;$i++) {
my $a=$facet[$x][$i];
my $b=$facet[$y][$i];
my $c=$a-$b;
$distance=$distance+$c*$c;
}
$distance=sqrt($distance);
return($distance);
}

print "Enter how fine you like the epsilon increments to be:";
$increment=<>;

print "Would you like to enter the Epsilon-Range manually? [no]:";
$char=<>;
chomp($char);
```

```
if (lc($char) eq ’y’) {
print "Epsilon Min:";
$emin=<>;
print "Epsilon Max:";
$emax=<>;
}

else {
$emax=0;
$emin=distance(0,1);

for (my $i=0; $i<$n-1;$i++) {
for (my $j=$i+1;$j<$n;$j++) {
my $dis=distance($i,$j);
if ($dis>$emax) {$emax=$dis;}
if ($dis<$emin) {$emin=$dis;}
}
}
}

open FILE, ">result.txt" or die "cannot create result";
$ninc=$increment+1;
print FILE "Number of vectors to cluster=$n\t Number of increments=$ninc\n";
print FILE "eMin=$emin;\t eMax=$emax\n";

my $a=$emax-$emin;
$increment=$a/$increment;

print FILE "Step-size=$increment\n\n";

$epsilon=$emin;

# Skip checking emin? (i.e. already checked it on the first pass)
# $epsilon=$emin+$increment;


# print FILE "# Facets\t Epsilon\n";
print FILE "# Vertices\t Epsilon\n";

$bound=$emax+$increment;
while ($epsilon<$bound) {
close FILE;
open FILE, ">>result.txt" or die "cannot create result";

$k=0;  # placeholder that determines where to begin the next partition.
$counter=0; # number of partitions (result)
```

```
@P=(); # Partitions

$l=0; # holds the outcome of the tests
while ($k<$n) {
@Q=($k);
@C=(); # Complement
for (my $j=0;$j<$n;$j++) {
if ($j!=$k) {
my $dis=distance($k,$j);
if ($dis<=$epsilon) {push(@Q,$j);}
else {push(@C,$j);}
}
}

for (my $h=1;$h<scalar(@Q);$h++) { # tests to see if there are points outside the ball that should b
my $x=$Q[$h];
for (my $j=0;$j<scalar(@C);$j++) {
my $y=$C[$j];
my $dis=distance($x,$y);
if ($dis<=$epsilon) {$l=2;}
last if $l==2;
}
last if $l==2;
}

if ($l==2) {
print $epsilon . ": invalid partition; points lying outside ball\n";
print FILE "-1\t";
}
last if $l==2;

if ($l==0) { # tests to see if there are two points inside the ball that are too far away from eacho
for (my $i=1;$i<scalar(@Q)-1;$i++) {
my $x=$Q[$i];
for (my $j=$i+1;$j<scalar(@Q);$j++) {
my $y=$Q[$j];
my $dis=distance($x,$y);
if ($dis>$epsilon ) {$l=3;}
last if $l==3;
}
last if $l==3;
}
}
if ($l==3) {
print $epsilon . ": invalid partition; points inside ball are further than epsilon\n";
print FILE "-2\t";
}
last if $l==3;

push(@P,@Q);
$counter++;


# Find the next k

my $t=0;
my $y=$counter;
while ($t==0 && $y<$n) {
$t=1;
foreach (@P) {
my $x=$_;
if ($x==$y) {$t=0;}
last if $x==$y;
}
if ($t==0) {$y++;}
}
$k=$y;
}

if ($l==0) {
print "$epsilon: $counter Facets\n";
print FILE $counter . "\t";
#if ($counter==1) {$one=1;}
}
print FILE $epsilon . "\n\n";
$epsilon=$epsilon+$increment;

# This ends the scan at the first instance of the entire data set being clumped into a single cluste
# last if $one==1;

}

close FILE;
```

References

1. Contois Bogart and Gubeladze, Hom-Polytopes, Math. Z., Vol. 273, pp.1267-1296 (2013)
2. Andrew Dynneson, Tensor product of polygons, Thesis: San Francisco State University (2012)
3. Ewgenij Gawrilow and Michael Joswig, polymake: a framework for analyzing convex polytopes, Polytopes-Combinatorics and Computation (Gil Kalai and Gunter M. Ziegler, eds.), Birkauser, pp. 43-74 (2000)

4. Winfried Bruns and Joseph Gubeladze, Polytopes, rings, and K-theory. Springer Monographs in Mathematics, Springer, Dordrecht (2009)
5. David S. Summit and Richard M. Foote, Abstract algebra, third ed., John Wiley and Sons Inc., Hoboken, NJ (2004)
6. Misner, Thorne, Wheeler, Gravitation. W.H. Freeman and Company (1973).