Classification of Reflexive Polytopes Using Machine Learning

Group 46

June 2022

Abstract

The aim of this research project is to investigate the plausibility of using Machine Learning techniques to distinguish equivalence classes of polytope based on change of basis operation. We attempted to adopt various algorithms within the spectrum of machine learning. At the end of the three-week project, we have generated a graph neural network program with a classification accuracy above 80 percent on dimension 3 cases, which indicates the feasibility of employing Machine Learning to examine equivalence relation between polytopes. We believe our result shows the application of machine learning in polytope classification is promising and provides inspirations to future research.

1 Introduction

1.1 Motivation

In order to check whether two polytopes are equivalent to one another up to change of basis, as mathematicians, we instantly think of whether a change of basis matrix M that sends one polytope to the other exists. To do so, we may need to consider every possible way of linearly transforming vertices of the former polytope to those of the latter one.

Despite of being familiar to mathematicians, above operations can be overly tiring and time-consuming for human brains to conduct. It, then, comes natural to people to try to build a tool completing the computations instead, for instance, computer programs. In fact, there are programs capable of doing polytope equivalence class identification already available online for everyone to download. Nevertheless, the existing algorithms are fast for dimension 2, acceptable for dimension 3, and excessively slow for dimension 4, as with increasing dimension, the computation quantity skyrockets and becomes stressing for computers to run within an appropriate period of time.

Therefore, in this research project, we are interested in making progress towards a new class of algorithms, which could be suitable for fast (and possibly approximated) equivalence class recognition of polytopes in dimension 4, utilizing deep learning. Additionally, it is also thrilling to combine a field of mathematics with one of the most cutting-edge technologies in computer sciences.

1.2 Overview of Process

Firstly, we started by investigating certain properties of polytopes and the spaces they lie in, for we can find a suitable way of representing polytopes in a computer program and create a more effective and efficient training logic for the machine learning code afterwards. We looked through lattices, dual operations, reflexive polytopes, plucker coordinates, toric varieties, protective geometry, and connections between polytopes and polynomials. This leads us to a solid foundation of classifying equivalence relationship, up to change of basis, of polytopes.

Secondly, we initiated to build a classifier, which is an algorithm that categorizes data into different classes automatically. To obtain a more precise and efficient classifier, we have explored different types of algorithms, including Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), and Graph Neural Network (GNN), in which the last one outperformed the former two in our experiments. Moreover, considering the difference of the learning path of an Artificial Intelligence to that of a human being, we attempt to write classifiers with little mathematical concept, although it turned out that the algebraic content assisted the code to gain a better accuracy in our case.

Further in this article, we will illustrate about the field of pure algebraic probed, machine learning related content applied, the experiments executed, and, eventually, the outcomes achieved.

2 Toric Varieties

2.1 Background

In this part of the paper we will try and motivate our polytope classification problem by exploring how we can obtain abstract variety from polytopes using the theory of fans and toric varieties. Before we can dive into that we must introduce some basic concepts in algebraic geometry, for simplicity we will work over the field of complex numbers $\mathbb{C}$.

2.1.1 Affine Varieties

Definition (Affine Varieties). Given polynomials $f_1, ..., f_s$ with variables $x_1, ..., x_n$ over the complex field $\mathbb{C}$ (**Notation:** $f_1, ..., f_s \in \mathbb{C}[x_1, ..., x_n]$), *we get the* ***affine variety***

$$\mathbf{V}(f_1, ..., f_s) \;=\; \{a \in \mathbb{C}^n \mid f_1(a) = \cdots = f_s(a) = 0\}.$$

If $I \subset \mathbb{C}[x_1, ..., x_n]$ is an ideal, we define

$$\mathbf{V}(I) \;=\; \{a \in \mathbb{C}^n \mid f(a) = 0 \; \forall f \in I\}.$$

Remark. As $\mathbb{C}[x_1, ..., x_n]$ is a Noetherian ring, I is always finitely generated by $f_1, ..., f_s$. Therefore $\mathbf{V}(I)$ can be expressed as the common zeros of the finite set of polynomials $f_1, ..., f_s$.

Definition (Irreducible Variety). An affine variety, V, is irreducible if $\nexists V_1, V_2$ such that $V = V_1 \cup V_2$ and $V \notin \{V_1, V_2\}$

Proposition 2.1 (Noether-Lasker). *Every variety V is the union of finitely many irreducible varieties.*

Definition (Vanishing Ideal). Let $W \subset \mathbb{C}^n$, then we write

$$I(W) = \{f \in \mathbb{C}[x_1, \ldots, x_n] \mid \; f(a) = 0, \;\; \forall a \in W]\}$$

Called the **vanishing ideal** of W

Clearly for any subset, $W \subset \mathbb{C}^n$, $I(W)$ is an ideal in $\mathbb{C}[x_1, \ldots, x_n]$

Definition (Coordinate Ring). Let $V \subset \mathbb{C}^n$ be a variety, then the **coordinate ring** of V, denoted $\mathbb{C}[V]$ is,

$$\mathbb{C}[V] = \mathbb{C}[x_1, \ldots, x_n]/I(V)$$

Intuitively, we can think of $\mathbb{C}[V]$ as the set of all distinct functions on V, since if we have functions $f, g \in \mathbb{C}[x_1, \ldots, x_n]$ such that $\forall x \in V$, $f(x) = g(x)$, this implies that $f - g \in I(V)$.

Definition (Radical). The radical of an ideal $I \subset \mathbb{C}[x_1, ..., x_n]$, $\sqrt{I}$, is defined as

$$\sqrt{I} = \{f \in \mathbb{C}[x_1, ..., x_n] \mid f^m \in I \text{ for some } m \geq 1\}.$$

Theorem 1 (Hilbert's Nullstellensatz). Let K be an algebraically closed field and let J be an ideal in $K[x_1, ..., x_n]$. Then,

$$(I(\mathbf{V}))(J) = \sqrt{J}$$

Remark. Notice that every maximal ideal M of $\mathbb{C}[x_1, ..., x_n]$ corresponds to a minimal irreducible closed subset of $\mathbb{C}^n$, which is just a point $P = (a_1, ..., a_n)$, ergo every maximal ideal is of the form $\langle x_1 - a_1, ..., x_n - a_n \rangle$, where $a_i \in \mathbb{C}$. We call such points closed points.

Lemma 3. *V is irreducible $\iff$ $I(V)$ is prime.*

So just as we have a correspondence between maximal ideals and closed points, we have a correspondence between prime ideals and irreducible varieties. It follows that if we have an irreducible variety V, then $\mathbb{C}[V]$ is an integral domain. Thus we are able to define a *function field* denoted $\mathbb{C}(V) = \text{Frac}\ \mathbb{C}[V]$ which is just the field of fractions defined in the usual way. If $f, g \in \mathbb{C}[x_1, \ldots, x_n]$ and $g \notin I(V)$, then f/g represents an element of $\mathbb{C}(V)$ and determines a map,

$$\phi : V \backslash \mathrm{V}(g) \to \mathbb{C}$$

We say that ϕ is a rational function and $x \in V$ is a regular point if we can find $f, g \in \mathbb{C}[x_1, \ldots, x_n]$ such that $\phi = f/g$ and $g(P) \neq 0$.

Definition (Local Ring). The local ring at a point $P \in V$, where V is an irreducible variety, is defined as the set regular rational functions at P,

$$\mathcal{O}_{V,P} = \{f \in \mathbb{C}(V) \mid \text{f is regular at P}\}$$

A key property of local rings it that they have a unique maximal ideal,

$$\mathfrak{m}_{V,P} = \{f \in \mathcal{O}_{V,P} \mid f(P) = 0\}$$

A property that will later become important once we develop the theory of toric variety is that of *normality.* However, before we can define it we need to introduce the notion of an *integrally closed domain.*

Definition (Integrally Closed Domain). Let R be a ring and let $K = \text{Frac}(R)$, then we say R is an integrally closed domain if every element in K which is integral - i.e. is a root of a monic polynomial in $R[x]$ - is also in R.

Now with this we can introduce normality for varieties,

Definition (Normal Variety). A variety V is normal if it is irreducible and $\forall\ P \in V$ the local ring $\mathcal{O}_{V,P}$ is an integrally closed domain.

It will later turn out that there is a bijective correspondence between normal toric varieties and fans (up to isomorphism).

3.1 Projective Varieties

First let's introduce projective n-space, as that is the space upon which we define projective varieties.

Definition (Projective Space). We define **n-dimensional projective space** as:

$$\mathbb{P}^n = (\mathbb{C}^{n+1} \setminus \{0\}) / \sim,$$

where the equivalence relation $\sim$ is defined as

$$(a_0, ..., a_n) \sim (b_0, ..., b_n) \iff \exists\, \lambda \in \mathbb{C}^* \text{ such that } (a_0, ..., a_n) = \lambda(b_0, ...b_n)$$

Remark. $(a_0, ...a_n)$ is called a *homogeneous coordinate*, and is unique up to multiplication by elements of $\mathbb{C}^*$.

It is clear that polynomials are not well-defined on $\mathbb{P}^n$, however if we restrict our discussion to only the zeroes of homogenous polynomials, we find that they are well-defined, since if $a \in \mathbb{C}^n$ and $b = ta$, for some $t \in \mathbb{C}^\times$,

$$f(a) = 0 \iff f(b) = 0$$

Given any homogenous polynomial f.

Definition (Projective Varieties). Consider **homogeneous polynomials** of degree d: $f_1, ..., f_s \in \mathbb{C}[x_0, ..., x_n]$ defined as

$$f(\lambda x_0, ..., \lambda x_n) = \lambda^d f(x_0, ..., x_n),\ \lambda \in \mathbb{C}^*$$

Then the **projective variety** is defined as

$$\mathbf{V}(f_1, ..., f_s) = \{a \in \mathbb{P}^n \mid f_1(a), ..., f_s(a) = 0\} \subset \mathbb{P}^n.$$

Remark. An ideal I is *homogeneous* if it is generated by homogeneous polynomials, and it defines the projective variety

$$\mathbf{V}(I) = \{a \in \mathbb{C}^n \mid f(a) = 0,\ \forall\ f \in I\}$$

3.1.1 Thoughts on Zariski topology on a variety

We have now introduced the notion of a variety, however let's say that we are interested in the spaces of solutions of polynomial equations over $\mathbb{C}^n$. The classical Euclidean topology is too fine to for our study of polynomials, so we would like something different. Since we are mainly concerned with zero-sets could we somehow construct a topology on $\mathbb{C}^n$ from affine varieties?

Let's start with some well-known facts we can verify by elementary algebra:

Lemma 4. *Let $a, b \subset \mathbb{C}[x_1, \ldots, x_n]$ be ideals. Then we have,*

1. $\mathbf{V}(0) = \mathbb{C}^n$
2. $\mathbf{V}(\mathbb{C}[x_1, \ldots, x_n]) = \emptyset$
3. $\mathbf{V}(ab) = \mathrm{V}(a) \bigcup \mathrm{V}(b)$
4. $\mathbf{V}(\sum_{i \in I} a_i) = \bigcap_{i \in I} \mathrm{V}(a_i)$, *where $\{a_i\}_{i \in I}$ is any collection of ideals*

Now we can see that affine varieties satisfy the axioms for closed sets of a topology on $\mathbb{C}^n$. It is called Zariski topology.

One can see that closed sets in Zariski topology on $\mathbb{C}^n$ are closed in usual topology of $\mathbb{C}^n$, since polynomials are continuous. So the Zariski and Euclidean topologies are comparable, and it is clear the Zariski topology is coarser.

Example 4.1. *Consider $\mathbb{C}$. Even if it is one of the simplest examples conceivable, it provides a bigger picture. Zariski closed sets in it are just a collection of roots of polynomial equations. So it is a cofinite topology on $\mathbb{C}$. This provides us intuition that Zariski topology identifies roots of polynomials as 'essential parts'.*

Note that closed sets are generated via $B = \{V(f) \mid f \in \mathbb{C}[x_1, ..., x_n]\}$, a set of varieties generated by irreducible polynomials. This in turn implies that the varieties are irreducible, by the previous lemmas. Irreducible varieties in themselves are rather strange. Let some irreducible variety have an induced Zariski topology, then any two open sets that are non-empty intersect. This in turn implies that every open set is dense as well as non-separable. Here we have a rare example of a non-Hausdorff space that is actually useful!

Then a new question arises, whether we can 'convert' Zariski topology into something algebraically meaningful? The answer is evidently yes! To summarise what we have seen so far we have the following one-to-one correspondences,

$$\text{radical ideals} \leftrightarrow \text{subvarieties}$$

$$\text{prime ideals} \leftrightarrow \text{irreducible subvarieties}$$

$$\text{maximal ideals} \leftrightarrow \text{closed points}$$

Also in our case it is important that:

Lemma 5. *A $\mathbb{C}$-algebra R is the coordinate ring of some affine variety $\iff$ R is finitely generated with no nilpotent elements.*

Because of this we can consider varieties as a pair (X, R) where X is a variety and R is its corresponding coordinate ring.

Continuing our journey of connecting varieties with algebra we define:

Definition (Spectrum of a Ring). The spectrum of a ring R, denoted $\mathrm{Spec}(R)$, is the set of all prime ideals of R.

For any ideal $I \subset R$ we denote V_I to be the set of all prime ideals which contain I. We can put a topology on $\operatorname{Spec}(R)$, by defining the set of closed sets to be:

$$\{V_I \mid I \text{ is an ideal of } R\}$$

We call this topology the *Zariski topology on a spectrum.*

Example 5.1. *Let's consider a classical example. Take* $\mathbb{C}[x,y]$. *Closed sets of* $\operatorname{Spec}(\mathbb{C}[x,y])$ *can be described in the following manner. Since* $\mathbb{C}[x,y]$ *is Noetherian we get that any maximal ideal corresponds to a point in* $\mathbb{C}^2$ *namely* $\langle X-x,\ Y-y\rangle$ *corresponds to* (x,y). *Also since any closed set is a union of irreducibles, we can just describe the correspondence with irreducibles. Namely if* $f(x,y)$ *is an irreducible polynomial, then the corresponding closed set is given by* $f=0$ *with a prime ideal generated by* f. *Note as well the closure is the whole* Spec.

Take a variety and coordinate ring pair as defined above, $(X,R)=(V,\mathbb{C}(V))$ then by the Nullstellensatz points of V correspond with maximal ideals of R that contain $I(V)$. Let's denote the collection of maximal ideals of R as $\operatorname{MaxSpec}(R)$, then it can be proved that the correspondence $V \mapsto \operatorname{MaxSpec}(\mathbb{C}[V])$ is a homeomorphism. In general we can define an association $R \mapsto \operatorname{Spec}(R)$ that determines a functor from **Ring** to **Top** by sending the homomorphism $f: R \mapsto S$ to $f^{-1}: \operatorname{Spec}(S) \mapsto \operatorname{Spec}(R)$. This is true in general, however if we are consider maximal ideals we have to be a bit more careful.

Lemma 6. $\mathbb{C}$-*algebra homomorphisms between coordinate rings send maximal ideals to maximal ideals.*

So we get a functor $\phi : \mathbf{Ring} \to \mathbf{Top}$, $\mathbb{C}[V] \mapsto \operatorname{MaxSpec}(\mathbb{C}[V])$ and by using the functor $\psi : \mathbf{Ring} \to \mathbf{AffineVariety}$, $\mathbb{C}[V] \mapsto V$ and Nullstellensatz we get an morphism of functors. So, we can treat varieties as the MaxSpec of its coordinate ring. By abuse of notation we identify the variety with the spectrum of its coordinate ring as that gives us additional information such as irreducible varieties correspondence.

6.0.1 Toric varieties

In $\mathbb{R}^n$ we define the n-dimensional torus as,

$$\mathbb{T}^n = \underbrace{S^1 \times \cdots \times S^1}_{\text{n times}}$$

But note that we can express S^1 as,

$$\mathrm{V}(x^2+y^2-1) = S^1$$

But if we instead consider the polynomial, x^2+y^2-1, over $\mathbb{C}$ then the variety becomes,

$$\mathrm{V}(x^2+y^2-1) = \mathbb{C}^\times$$

Thus we arrive upon the definition,

Definition (Algebraic Torus). An algebraic torus of rank 1 is defined as $\mathbb{G}_m := \mathbb{C}^\times$, and the rank n algebraic torus is given by $\mathbb{G}_m^n := (\mathbb{C}^\times)^n$.

Now consider the variety $\mathbb{P}^n$ with homogeneous coordinates $x_0, ..., x_n$ and a map

$$\phi : \mathbb{G}_m^n \to \mathbb{P}^n,\ (t_1, ..., t_n) \mapsto (1, t_1, ..., t_n)$$

which implies that we can identify $\mathbb{G}_m^n$ with the Zariski open subset $\mathbb{P}^n \setminus V(x_0, ..., x_n)$. We may also define:

$$(t_1, ..., t_n) \cdot (a_0, ..., a_n) = (a_0, t_1 a_1, ..., t_n a_n)$$

It can be shown that this definition is an action of the algebraic torus on $\mathbb{P}^n$. Such varieties are called toric. In general we define:

Definition (Toric Variety). A toric variety X is an irreducible variety that contains $\mathbb{G}_m^n$ as a dense open subset and such that the multiplication map defined in the usual way extends to a group action on X.

Note since X is irreducible, non-empty and open implies dense. Consider another example:

Example 6.1. *Consider cuspidal curve* $C = \mathrm{V}(y^2 - x^3) \subset \mathbb{C}^2$. *Notice it is a toric variety. We consider map* $t \mapsto (t^2, t^3)$ *which gives an open torus* $T = C \setminus \mathrm{V}(x, y^2 - x^3)$. *Action associated to it is* $t \cdot (x, y) = (t^2 x, t^3 y)$

Now we should start thinking how one can represent toric varieties in a simple manner. Since varieties as mentioned above can be related to polynomials, and polynomials can be related to lattices, we start doing the latter one.

Lattice in general is a free Abelian group of finite rank. Consider a triplet (N, M, b) which is a pairing over a field $\mathbb{Z}$ where N is a lattice and $M = \mathrm{Hom}_{\mathbb{Z}}(N, \mathbb{Z})$ is its dual with bilinear map associated the pair. For our purposes we choose $\mathbb{Z}$-basis for N, then $M \sim \mathbb{Z}^n$ for some n, and b is a classical dot product.

The question now standing is how we relate points $a = (a_1, ..., a_n) \in \mathbb{Z}^n$.

Laurent Monomials

Laurent monomial is defined as a character of algebraic torus into C^*, or in other words(considering our lattice identifications above) it is a map $\Xi^a : \mathrm{G}_n \mapsto \mathbb{C}^\times$ as

$$(t_1, ..., t_n) \mapsto (t_1^{a_1} ... t_n^{a_n})$$

Note this **a** belongs to dual lattice M. These will be our 'building' blocks to coordinate rings associated to toric varieties.

Remark. Note that if we fix some $t \in G_n$ then Ξ will be semi-group homomorphism with respect to lattice. Such homomorphisms later will give us correspondance to lattice points and new visualisation of a variety.

1-parameter subgroup of $\mathbb{G}_m^n$

For $b = (b_1, ..., b_n) \in N$ let's define $\lambda^b : \mathbb{C} \mapsto G_m^n$ as $t \mapsto (t^{b_1}, ..., t^{b_n})$.

Remark. One might recreate toric variety using 1-parametric groups. Informally if we will consider $\lim_{t \mapsto 0} \lambda^u(t)$ we will obtain orbit generators for torus orbits. Then by considering such limit as a relation we will obtain a fan.

Remark. One could also note that given a character Ξ^m and 1-parameter subgroup λ_n one gets $\Xi^m \circ \lambda_n$: $\mathbb{C}^\times \mapsto \mathbb{C}^\times]$ is a character denoted as $t \mapsto t^{<m,n>}$. This is no coincidence, it is the definition of our bilinear map.

Let's start constructing our first examples with association to the lattice.

Consider an example:

Example 6.2. *Note that a map* $t \mapsto (\Xi^2(t), \Xi^3(t)) = (t^2, t^3) \subset \mathbb{C}^2$. *Note it is an embedding of a 1-torus in* $\mathbb{C}^2$ *where Zariski closure is* $V(y^2 - x^3)$.

This example generalizes quite nicely. Consider a subset $B = \{m_1, ..., m_l\} \subset M$ where M is dual lattice. Then by passing from a lattice to varieties we can define a map $\Phi_B : G_m^n \mapsto \mathbb{C}^\times$ as $t \mapsto (\Xi^{m_1}(t), ..., \Xi^{m_l}(t))$ We denote $Y_B = \mathrm{cl}(\mathrm{Im}\Phi_B)$ in Zariski topology. This is a variety and also it is an affine toric variety.

Proposition 6.1. Y_B *is an affine toric variety.*

Remark. Later this proposition will be useful in showing that specific gluing of toric varieties will be toric.

For continuity let's briefly connect this toric variety representation with algebra. Consider the solution space for this variety in $\mathbb{C}[x_1, ..., x_n]$, Namely vanishing ideal $I(Y_B) \subset \mathbb{C}[x_1, ..., x_n]$. Question is how does this ideal look like. Note that character are semi-group homomorphisms. That means that the key information is the values at the generations B. If $B = \{m_1, ..., m_s\}$ and $\Sigma_i \alpha_i m_i = \Sigma_i \beta_i m_i$ then $\Pi_i \Xi^{\Sigma_i \alpha_i m_i} = \Pi_i \Xi^{\Sigma_i \beta_i m_i}$. In general any such binomial belongs to the vanishing ideal of Y_B. If we represent vectors as a matrix $A \in M^{m \times n}$ where n is a number of elements in B we can prove that

Proposition 6.2. $I(Y_B) =< x^{k_+} - x^{k_-} >$ *where* k_+ *and* k_- *are non-negative and* $k_+ - k_- \in ker(A)$.

Cox page 15.

Remark. If we will denote $H = \{m \in S_\sigma \mid \text{m is irreducible}\}$ where by irreducibility we mean that m can not be expressed as the sum of two non zero elements from S_σ. This is known as the Hilbert basis, which in our case contains minimal generators of S_σ. It follows from that we can embed affine toric variety $\text{Spec}(\mathbb{C}[S_\sigma])$ via the above monomial map into $|H|$ dimensional affine space.

Ideals in the last proposition are named toric ideals. Namely

Definition. $I \subset \mathbb{C}[x_1, ..., x_n]$ is called toric ideal if it is a prime ideal generated by binomials.

For completion let's consider an example:

Example 6.3. *Consider* $B = \{(1,0), (0,1), (1,1)\}$*, then* $I(Y_B) = \langle xy - z \rangle$.

Remark. It can be developed further by considering embedding of $\mathbb{C}^n$ in $\mathbb{P}^n$ and then considering projective toric varieties. Ones who are interested should check [5]

6.0.2 Cones

We will now introduce some convex geometry,

Definition (Cones). A polyhedral convex cone is a subset of $\mathbb{R}^n$

$$\sigma = \text{Cone}(S) = \{\sum_{v \in S} \lambda_v v \mid \lambda_v \geq 0\}$$

Where $S \subset \mathbb{R}^n$, is finite. We say that σ is the cone generated by S

If $S \subset \mathbb{Z}^n$, then we say σ is *rational*. As the name suggests σ is convex.

Definition (Polytopes). A polytope is a subset of $\mathbb{R}^n$

$$P = \text{Conv}(S) = \{\sum_{v \in S} \lambda_v v \mid \sum_{v \in S} \lambda_v = 1 \ ; \ \lambda_v \geq 0\}$$

We say that P is the *convex hull* of S.

We can also define cones as the intersection of finitely many half-spaces. Every generator v, has an associated half-space, $H_v = \{u \in \mathbb{R}^n | \langle v, u \rangle \geq 0\}$. Then the cone, σ, generated by $S \subset \mathbb{R}^n$ is given by

$$\sigma = \bigcap_{v \in S} H_v$$

Definition (Dual Cone). Let σ be a cone. Its dual, denoted $\sigma^\vee$, is given by,

$$\sigma^\vee = \{u \in (\mathbb{R}^n)^* | \langle u, v \rangle \in \sigma\}$$

Where $(\mathbb{R})^*$ is the dual space of $\mathbb{R}^n$

Definition (Cone). [Let σ be a cone and let $\lambda \in \sigma^\vee \cap M$ then,

$$\tau = \sigma \cap \lambda^\perp = \{v \in \sigma | \langle \lambda, v \rangle = 0\}$$

is called a face of σ, and we write $\tau < \sigma$

Clearly the face of rational convex polyhedral cone is also a rational convex polyhedral cone. We will now list state some useful properties.

Proposition 6.3. *Let σ be a cone, and let $\lambda \in \sigma^\vee$*

- $\tau < \sigma \implies \sigma^\vee \subset \tau^\vee$
- *Let $\sigma = \sigma_1 + \sigma_2$, then $\sigma^\vee = \sigma_1^\vee \cap \sigma_2^\vee$*
- *If $\tau = \sigma \cap \lambda^\perp$ is a face of σ, then*
$$\tau^\vee = \sigma^\vee + \mathbb{R}_{0\geq}(-\lambda)$$

Given a cone, σ, let us define, $S = \sigma^\vee \cap M$. Similarly if we have a rational convex polyhedral cone σ with face, $\tau = \sigma \cap \lambda^\perp$, then

$$\tau^\vee = S_\sigma + \mathbb{Z}_{0\geq}(-\lambda)$$

Definition (Fans). A fan Σ is a collection of rational convex polyhedral cones which satisfies the following properties:

- If $\sigma \in \Sigma$ has a face τ, then $\tau \in \Sigma$
- If $\sigma_1, \sigma_2 \in \Sigma$ then $\sigma_1 \cap \sigma_2$ is a face of both σ_1 and σ_2

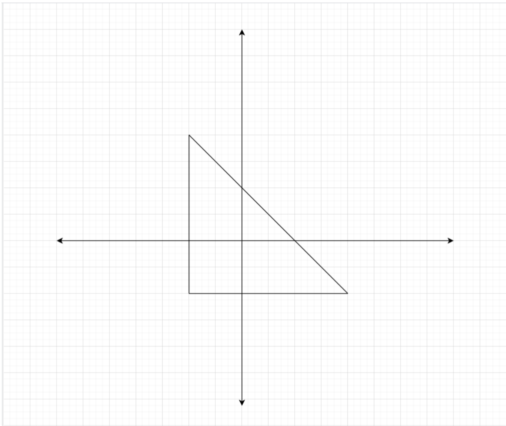
Proposition 6.4. *For each face τ of our polytope P, we associate it to a cone*

$$\sigma_\tau = \{v \in N_\mathbb{R} \mid \langle u, v \rangle \leq \langle u', v \rangle \ \forall u \in \tau \text{ and } u' \in P\}.$$

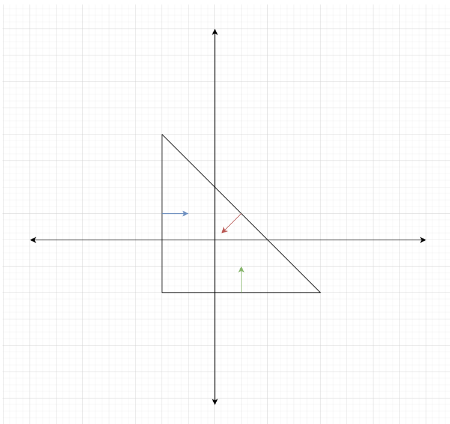
Then we have

- *The cones σ_τ form a fan Σ_P*
- *If $\{0\} \in \mathrm{Int}(P)$, then Σ_P is made of the cones based on the faces of of the polar polytope, P°*

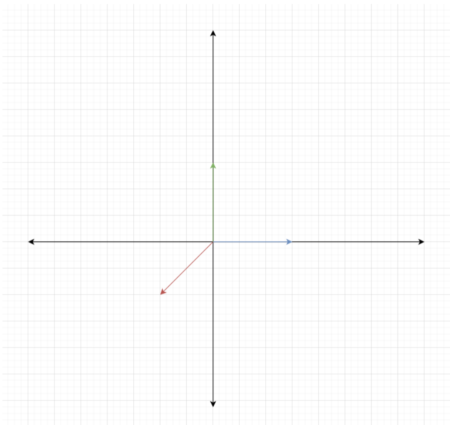
Let's now go through an example of how we might obtain a fan given a polytope. Consider the following polytope,



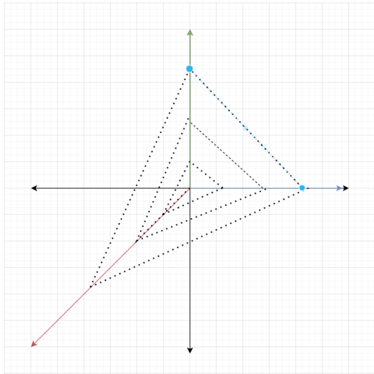
We then look at the inward normal of the faces of the polytope.



If we take the normal rays and place them all at the origin,



Finally, we can see that the three rays when extended forms three cones which form a fan!



It's easy to apply this method to construct fans out of other polytopes. Here is another example

Example 6.4. *Here is an example of a normal fan Σ_P made of cones σ_0, σ_1, and σ_2, from the original vertices of the triangle in $\mathbb{R}^2$.*

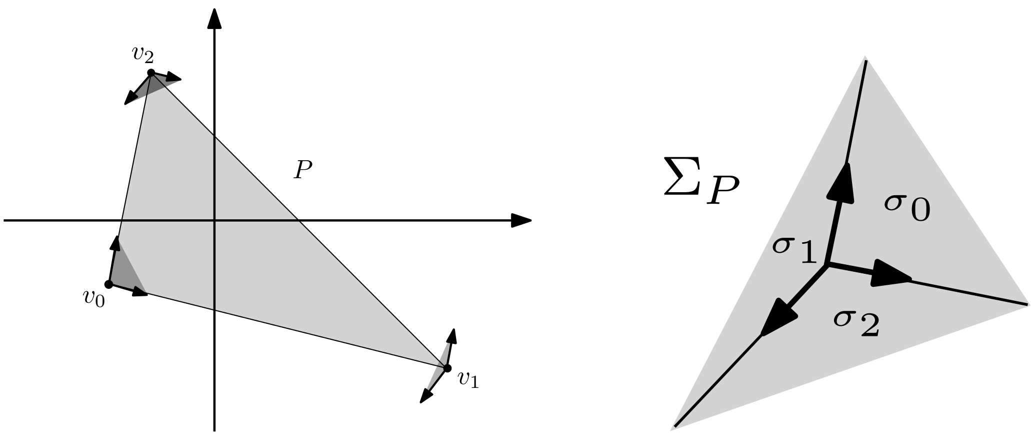


Figure 1: Triangle $P \in \mathbb{R}^2$ and its normal fan Σ_P

This method generalises to higher dimension as well, if instead we are in 3D,

Example 6.5. *Here is an example of a normal fan of a lattice tetrahedron in $\mathbb{R}^3$. The arrows pointing from each vertex v_i are the normal vectors of the facets containing v_i and the shaded cone σ_i they generate.*

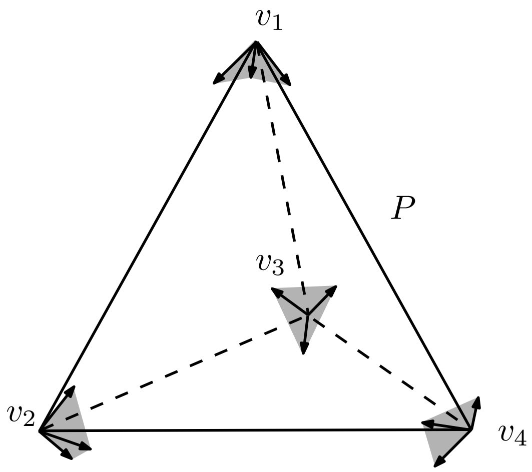


Figure 2: Tetrahedron in $\mathbb{R}^3$

An important concept we need in the development of toric varieties is that of the *dual* cone, which is the corresponding geometric structure of a cone in dual space.

Proposition 6.5. *Let $P \subseteq M_{\mathbb{R}}$ be a full dimensional lattice polytope of dimension n and consider the cones σ_Q in the normal fan Σ_p of P. Then:*

- $dim\, Q + dim\, \sigma_Q = n \;\forall$ *faces* $Q < P$.
- $N_{\mathbb{R}} = \bigcup_{v \text{ vertex of } P} \sigma_v = \bigcup_{\sigma_Q \in \Sigma_P} \sigma_Q$.

Remark. Note that a fan satisfying the second condition of the proposition is called *complete*. A normal fan of a lattice polytope is always complete.

Before we can state an important result, we need to introduce the notion of a semi-group

Definition (Semi-Group). A Semi-Group is a set S together with an associative binary relation, "$\cdot$", that is

$$\forall\, a, b, c \in S,\; (a \cdot b) \cdot c = a \cdot (b \cdot c)$$

We can now state Gordan's lemma,

Lemma 7 (Gordan's Lemma). *The semi-group of integral points of a rational polyhedral convex cone, $\sigma \cap N$, is finitely generated.*

Proof can be found in Cox.

Lemma 8 (Seperation Lemma). *If σ_1, σ_2 are convex polyhedral cones and $\tau = \sigma_1 \cap \sigma_2$ is a face of each, then for all m in the relative interior of $\sigma_1^\vee \cap (-\sigma_2^\vee)$,*

$$\tau = \sigma_1 \cap m^\perp = \sigma_2 \cap m^\perp$$

8.1 Toric Varieties of Fans

After providing background to cones, we start converting cones into algebras and then converting everything to **Top** category.

Let $\sigma \subset \mathbb{R}^n$ be a rational cone. Then consider $S_\sigma = \sigma^\vee \cap M$ where $M \sim \mathbb{Z}^n$ is a dual lattice. Gordon's theorem implies that S_σ is semi-group what in itself implies that $\mathbb{C}[S_\sigma]$ is a finitely generated $\mathbb{C}$-algebra. For motivation why it is important remember the beginning of our discussion, this allows us to go from **Ring** category to **Top** category where it is meaningful to restrict maps of prime spectra to maximal spectra.

Remark. Here $\mathbb{C}[S_\sigma]$ is a linear combinations of characters related to S_σ. In other words, finite sums $\Sigma a_i \Xi^{m_i}$ where $m_i \in S_\sigma, a_i \in \mathbb{C}$ and $\Xi^m \Xi^{m'} = \Xi^{m+m'}$

Notice we obtain another representation of algebraic torus as a spectrum of $\mathbb{C}[t_1, t_1^{-1}, ..., t_n, t_n^{-1}]$. In addition to that note that $C[S_\sigma]$ is an integral domain it makes sense to consider $\mathbb{C}[S_\sigma] \simeq \frac{\mathbb{C}[x_1,...,x_h]}{I}$ where I is prime. Now considering spectrum of later one, we get that points in corresponding variety corresponds to maximal ideals that contains I then such variety can be identified to V(I) and by the discussion in the beginning, V(I) is irreducible since I is prime.

Remark. Remember that when we have a spectrum of a coordinate ring, we can express that coordinate ring as a $\frac{\mathbb{C}[x_1,...,x_n]}{I}$ where I is so ideal. Then our variety of interest is isomorphic to $V(I)$.

Assume that we are considering strongly convex cones $\sigma \subset N$, then $\{0\} \subset \sigma$ what implies that there in an injection $\mathbb{C}[S_\sigma] \subset \mathbb{C}[S_{\{0\}}] = \mathbb{C}[t_1, t_1^{-1}, ..., t_n, t_n^{-1}]$. Passing to **Spec** subcategory of **Top** we get an injection(embedding) $\mathbb{G}_n \simeq \mathrm{Spec}(\mathbb{C}[t_1, t_1^{-1}, ..., t_n, t_n^{-1}]) \mapsto \mathrm{Spec}(\mathbb{C}[S_\sigma])$. This gives us intuition to define the following:

Proposition 8.1. *The algebraic torus $\mathbb{G}_m^n$ is congruent to* $\mathrm{Spec}(\mathbb{C}[t_1, t_1^{-1}, ..., t_n, t_n^{-1}])$ *and* $\mathrm{Cone}(\{0\})$.

Since all cones contain $\mathrm{Cone}(\{0\})$, then that implies that all cones contain the torus as an open subset. Let's consider a few examples.

Example 8.1. *Consider a cone* $\sigma = \mathrm{Cone}(e_1, ..., e_n)$ *and let* $S_\sigma = \mathbb{Z}_{0\leq}$ *and* $\mathbb{C}[S_\sigma] = \mathbb{C}[x_1, ..., x_n]$. *Then we get the correspondence* $\mathrm{Spec}(\mathbb{C}[S_\sigma]) = V_\sigma = \mathbb{C}^n$

Consider another, less trivial example:

Example 8.2. *Let* $\sigma = \mathrm{Cone}(e_1, e_2, 2e_1 - e_2)$, *note it is convex. Then we have* $\sigma = H^+_{e_2} \cap H^+_{e_1} \cap H^+_{e_1+2e_2}$ *which implies that* $\sigma^\vee = \mathrm{Cone}(e_2, e_1, e_1 + 2e_2)$. *Now let's consider the map* $\mathbb{C}[x, y, z] \mapsto \mathbb{C}[x_2, x_1, x_1x_2^2]$ *with basis going to basis. The map is surjective, so it provides us an isomorphism* $\mathbb{C}[S_\sigma] \sim \frac{\mathbb{C}[x,y,z]}{<yx^2-z>}$. *Note* $V_\sigma = \mathrm{Spec}(\mathbb{C}[S_\sigma]) = V(yx^2 - z) \subset \mathbb{C}^3$. *In other words, it gives us all points whose maximal ideals contains the ideal* $< yx^2 - z >$.

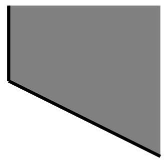


Figure 3: Cone σ

However we might get a feel that we are not utilising characters to their full extent. To this point our discussion about toric group actions has been coordinate related, however note that the characters on the torus are. Consider a semi-group S that belongs to a lattice, then the variety associated to it is $\mathrm{Spec}(\mathbb{C}[S])$. Note the evaluation maps gives different maps at different points from $\mathbb{C}[S]$ to $\mathbb{C}$. We claim that this gives us a correspondence to points in the variety.

Theorem 2. Let S be a semi-group and let $X = \mathrm{Spec}(C[S])$ be it's corresponding toric variety then there is a bijective correspondence between points $p \in X$ and semi-group homomorphisms $S \mapsto \mathbb{C}$.

Consider a semi-group homomorphism $x : S \mapsto \mathbb{C}$. Note it induces $\mathbb{C}$-algebras map $X : \mathbb{S} \mapsto \mathbb{C}$ in an obvious way which is surjective. Since it is a surjective and the image is a field, we get that a kernel is maximal, ergo we get a correspondence to a point by above sections. From 'coordinates' perspective we get a point $(x(m_1), ..., x(m_n))$ where m_i's a Hilbert basis. On the other hand for a point p we can just associate $m \mapsto \Xi^m(p)$. For full proof consult [22....]

We also know that action of torus induces an action on semi group homomorphisms, namely

Proposition 8.2. *If $x_p : S \mapsto \mathbb{C}$ is a semi-group homomorphism corresponding to a point $p \in V$ then an element $t \in \mathbb{G}_m^n$ is a map $(m \mapsto x_p(m)) \mapsto (m \mapsto \Xi^m(t)x_p(m))$.*

Same paper as above...

Remark. Note that if we take

$$m \mapsto \begin{cases} \text{m} = 0 & 1 \\ \text{otherwise} & 0 \end{cases}$$

we get a fixed point of the variety. Moreover it is a 0 representation in the variety, and only such fixed point of the action. So, to conclude we will have a fixed point if and only if 0 is in the variety.

Remark. In general we could define a distinguished point of a cone σ as $x_\sigma(u) = 1$ iff $u \in \sigma^\perp$ otherwise it is zero. It corresponds to the orbits of a face in a fan. For further developments, interested reader could consult [Brass] for light exposition.

8.1.1 Normality of Fans

Now we will discuss a few properties of varieties. One of the nicest properties that a variety can have is smoothness and the next best thing is normality. Remember that the variety V is normal, if its coordinate ring $\mathbb{C}[V]$ is integral over its field of fraction $\mathbb{C}(V)$. For an example of non-normality take,

Example 8.3. *Consider $V = V(y^2 - x^3)$. It is not normal since $\mathbb{C}[V] \simeq \frac{\mathbb{C}[x,y]}{\langle y^2 - x^3 \rangle}$. Note that $\frac{y}{x}$ solves $X^2 - x = 0$ but it is not in a coordinate ring.*

Here we will reinterpret smoothness and normality geometrically as well as describe differential perspective of it..

First of all notice that all smooth varieties will be normal ones,

Proposition 8.3. *Let X be a toric variety,*

$$X \text{ is smooth} \implies X \text{ is normal}$$

The converse doesn't hold. It can be verified that the following example is normal but not smooth.

Example 8.4. *V(xy-zw)*

Imagine that we are in a lattice $M \sim \mathbb{Z}$, and we chose a semi-group $S = \{2, 3\} \subset M$. This provides a cupsidal curve which by above is not normal; however if we will take let's say $S = \{e_1, e_2\}$ we will get normal variety. There is a fundamental difference between them, in the first one we can find a natural number such that division by it gives us a lattice point not belonging to a semi-group S. That motivates us to explore this property more.

Definition (Saturation). Let $L \subset \mathbb{Z}^m$ be a sublattice, then the saturation of L is the lattice is,

$$\mathrm{Sat}(L) = \{\alpha \in \mathbb{Z}^m \mid d\alpha \in L \text{ for some } d \in \mathbb{Z}^+\}$$

If $\mathrm{Sat}(L) = L$ we say that L is saturated.

One can see that if we have a non-saturated semi-group associated to the variety, then that element which does not belong to a semi-group but whose multiple does will give us an element not belonging to a coordinate ring but being a root of a polynomial in ring of fractions, and so is not normal.

That motivates us to state

Theorem 3. Let X be a toric variety then the following are equivalent:

- X is normal
- $X = \mathrm{MaxSpec}(\mathrm{C}[S])$ for a saturated semi-group $S \subset M$
- $X \cong U_\sigma = \mathrm{MaxSpec}(\mathbb{C}[S_\sigma])$ for some strongly convex rational cone $\sigma \subset N_\mathbb{R}$

Proof can be found in Telen.
Let's get back smoothness context. Point in a variety in general is said to be smooth if variety dimension agrees with tangent space dimension at that point.

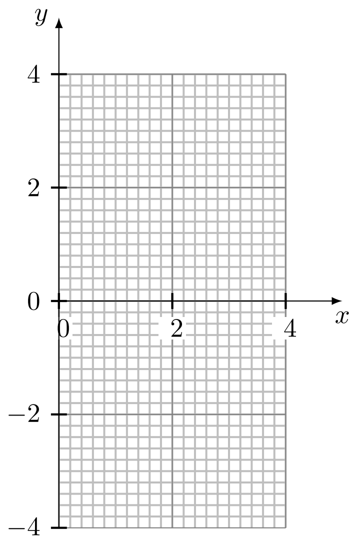
A dimension for a variety in general is defined as a Krull dimension, meaning that it is

$$dim(V) = max_d(V_0 \subset ... \subset V_d)$$

where V_i's are non-empty irreducible varieties.

Remark. Note Krull dimension is defined for a commutative ring as a length minus 1 of a maximal chain of prime ideals. In our case we have a correspondence with irreducible subvarieties. Also since coordinate ring is Noetherian, we get a finiteness of such dimensions.

Example 8.5. *Note tangent space of cupsidal curve is 2 dimensional at the origin, so it is not smooth.*



There is an analogy of tangent spaces in the algebraic context. For a point $p \in V$ of some variety, Consider local ring $R = \mathcal{O}_{V,p}$. This gives us a unique maximal ideal associated to the point p in the local ring, namely $\mathfrak{m}_{V,P} = \{f \in \mathcal{O}_{V,P} \mid f(P) = 0\}$. Now we consider $\frac{m_p}{m_p^2}$. This is called the Zariski cotangent space and the Zariski tangent space is its dual Hom$(\frac{m_p}{m_p^2}, \mathbb{C})$.

But note the Zariski cotangent space is essentially just a linearisation! We can derive it by considering taylor expansions. The maximal ideal, in fact, corresponds to the higher order terms of their expansion. Also one might have notice considering above example that in case of smoothness the notions corresponds, namely

Example 8.6. *Note that Zariski cotangent space for a cusp is* $\frac{\langle x,y\rangle}{\langle x^2,xy,y^2\rangle}$ *at point* $(0, 0)$.

In conclusion here,normality is useful due to the fact that it bounds the dimension of singular points variety, to be more precise $dim(V_s) \leq n - 2$ if $dim(V) = n$. That is why it is next best thing to smoothness. It will have singular points however they will not be very wild. For further references check out [Wan]. Later we see that normal torics can all be constructed in one way.

8.1.2 Constructing Varieties from Fans

Above we expressed normal toric varieties as the spectra of $\mathbb{C}$-algebras, now we will explore how we could build new varieties in the methods discussed above.

We start this section with an exploration of complex projective plane $\mathbb{P}^2$. It can be realised as a gluing of three $\mathbb{C}^2$. Let's take homogeneous coordinates (x_0, x_1, x_2) for $\mathbb{P}^2$ and consider coordinate charts

$$U_i = \{(\frac{x_0}{x_i} : \frac{x_1}{x_i} : \frac{x_2}{x_i})|x_i \neq 0\}$$

which are isomorphic to $\mathbb{C}^2$. It is clear that they are glued with maps induced via action on projective space that gives the complex plane. Also U_i corresponds to $\mathbb{C}[U_i]$ which is the subring of regular functions. Then $\mathbb{C}[U_i] = \mathbb{C}[\frac{x_j}{x_i}, \frac{x_k}{x_i}]$ where i,j,k is a permutation of $(0, 1, 2)$. Let $X = \frac{x_1}{x_0}$ and $Y = \frac{x_2}{x_0}$ then we obtain

$$\mathbb{C}[U_0] = \mathbb{C}[X, Y], \mathbb{C}[U_1] = \mathbb{C}[X^{-1}, X^{-1}Y], \mathbb{C}[U_2] = \mathbb{C}[Y^{-1}, XY^{-1}]$$

. Note each ring corresponds to the cones $\sigma_0, \sigma_1, \sigma_2$ generated by $\{(1, 0), (0, 1)\}$, $\{(0, 1), (-1, -1)\}$ and $\{(1, 0), (-1, -1)\}$.

Speaking less formally each face of a fan has a $\mathbb{C}$-algebra associated to it which in turn gives a variety. Consider how one should glue them. Let's take two faces $\sigma_1, \sigma_2 \in \Sigma$. Consider $\tau = \sigma_1 \cap \sigma_2$. Taking the dualization of the cones, reverses the inclusions and that gives an inclusion between $\mathbb{C}$-algebras. Let's take $\mathbb{C}[S_{\sigma_1}] \subset \mathbb{C}[\tau]$. We know that going to specs we can flip the arrows and get a map $V_\tau - > V_\sigma$. We see that $S_\tau = S_\sigma \cap H_m$ for some $m \in \sigma^\vee \cup M$ here M is a lattice. That in turn gives an expression for S_τ, namely $S_\tau = S_\sigma + (-m)\mathbb{Z}$. That means

$$\mathbb{C}[S_\sigma] \subset \mathbb{C}[S_\sigma]_{\Xi^m}$$

.

Note from here we get an open immersion map shown above.

Lemma 9. *If S is the mutiplicative set of regular functions of a ring A then the homomorphism $A \mapsto S^{-1}A$ induces a map* $\mathrm{Spec}(S^{-1}A) \mapsto \mathrm{Spec}(A)$ *where images are such $p \in \mathrm{Spec}(A)$ such that $p \cap S = \emptyset$*

Noting that it is an homeomorphism on the image, provides us with identification that $U_\tau \sim U_\sigma \backslash \{u_k = 0\}$ where u_k is a coordinate associated with Ξ^m.

Here we obtain an abstract variety structure on a fan where $f_1 : U_\tau \mapsto U_{\sigma_1}$ and $f_1 : U_\tau \mapsto U_{\sigma_2}$ provides and isomorphism $\mathrm{Im} f_1 \mapsto \mathrm{Im} f_2$. That is so called gluing of spaces.

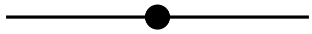


Figure 4: Corresponds to $\mathbb{P}^1$.

Let's look at the example

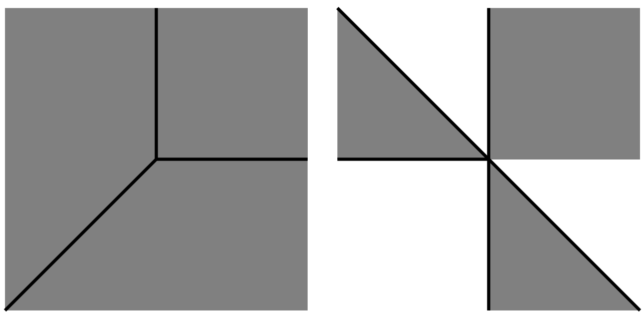
Example 9.1. *Let's take a cones* $\mathrm{Cone}(e_1)$, $\mathrm{Cone}(-e_1)$ *and* $\{0\}$ *in* $\mathbb{R}$. *That gives us a picture:*

Note we have a following inclusions $\mathbb{C}[t] \supset \mathbb{C}[t,t^{-1}] \subset \mathbb{C}[t^{-1}]$. *That gives us identification* $t \mapsto t^{-1}$ *which implies that we have a* $\mathbb{P}^1$. *To elaborate more, note that* $\mathbb{C}[t,t^{-1}] \sim \frac{\mathbb{C}[x,y]}{<xy-1>}$. *Then element in corresponding spec is a maximal ideal* $\frac{<x-a,y-b>}{xy-1}$. *In one image we get an element* a *and in other we get* $b = a^{-1}$ *due to relation* $xy-1$ *where* $a \neq 0$. *Also varieties corresponding to non-zero cones here is just* $\mathbb{C}$. *That concludes our observation*

Theorem 4. The construction above is a **normal toric variety** denoted by X_Σ. [Cox]

We conclude our section with following examples:

Example 9.2. *Look at the dual fan of cone described below. We can see that their three cones* $\sigma_0 = \mathrm{Cone}(e_1,e_2)$, $\sigma_1 = \mathrm{Cone}(e_1, e_1+e_2)$ *and* $\sigma_2 = mathrmCone(e_2, e_1+e_2)$ *gives us a three* $\mathbb{C}^2$ *described above. Also their gluings are specifically the same as in our construction of* $\mathbb{P}^2$ *fan. Also the face* $\sigma = \{0\}$ *identifies all tori inside those* $\mathbb{C}^2$ *and so we get a toric variety* $\mathbb{P}^2$.



(a) Fan of $\mathbb{P}^2$ (b) Dual of $\mathbb{P}^2$

Let's thing about slightly different example:

Example 9.3. *Consider cones in* $\mathbb{R}^2$, *namely a fan with maximal cones being* $\sigma_1 = \mathrm{Cone}(e_1, e_1+e_2)$, $\sigma_2 = \mathrm{Cone}(e_2, e_1+e_2)$ *and* τ. *We get a covering of our variety by two affine sets, namely* $U_{\sigma_1} = \mathrm{Spec}(\mathbb{C}[y, xy^{-1}])$ *and* $U_{\sigma_2} = \mathrm{Spec}(\mathbb{C}[x, x^{-1}y])$. *These correspond to* $\mathbb{C}^2$. *We glue along intersection of* σ_1 *and* σ_2, *namely* $\tau = \sigma_1 \cap \sigma_2 = \mathrm{Cone}(e_1+e_2)$. *We get a corresponding variety* $U_\tau = \mathrm{Spec}(\mathbb{C}[xy^{-1}, x^{-1}y, x, y]) = \mathrm{Spec}(\mathbb{C}[xy^{-1}, y]_{xy^{-1}}) = \mathrm{Spec}(\mathbb{C}[x^{-1}y, x]_{x^{-1}y})$. *We get the following identification:*

$$U_{\sigma_1} \leftrightarrow \mathrm{Spec}(\mathbb{C}[xy^{-1}, y]_{xy^{-1}}) \sim \mathrm{Spec}(\mathbb{C}[x^{-1}y, x]_{x^{-1}y}) \to U_{\sigma_2}$$

Note this variety is isomorphic to $\mathbb{C} \times \mathbb{C}^{\times}$ Let's think about maximal ideals. The form of the middle variety the maximal ideal will be $\langle x-a, y-b, x^{-1}y - a^{-1}b, y^{-1}x - b^{-1}a \rangle$. That implies that we have a correspondence $xt_0 = yt_1$ where x, y are complex and t_0, t_1 are homogeneous coordinates corresponding to $xy^{-1} \mapsto x^{-1}y$. This is a simple example of a fundamental concept in algebraic geometry called the blow up, in this case we are blowing-up $\mathbb{C}^2$.

Now we will conclude this pure part. One can see that actually we have a normal toric variety correspondence with a fan, namely:

Theorem 5. If X is a normal toric variety, we can associate a fan Σ to it such that $X_{\Sigma} \simeq X$, the reverse also holds.

Later one is more or less evident since strongly convex cones are normal and also gluing does not change normality.

In conclusion, normal toric varieties are nice despite having singularities and are of interest due to their correspondence with fans which the paper mentions can also by extension correspond to polytopes. If we had consider time to only consider the subset of reflexive polytopes, we would have been able to explore the rich family of Gorenstein-Fano toric varieties. However, this area will be explored in the context of establishing basis change equivalence classes using machine learning.

10 Convex Polytope Classification

Given two lattice polytopes $P, Q \subset \mathbb{Z}^n$, we want to find whether they are isomorphic:

Definition (Polytope Isomorphism)**.** Two lattice polytopes on the same ambient lattice $P, Q \subset \Lambda$ are **isomorphic** iff there exists an automorphism $\phi : \Lambda \to \Lambda$ such that $\phi(P) = Q$.

Equivalently, seeing the application of an automorphism as a "change-of-basis" operation, we can say that $P \cong Q$ if and only if there are some bases B, B' of Λ such that taking the vertices of P with respect to B and expressing them with respect to B' gives the vertices of Q. For simplicity, from now on we will use the word "polytope" to refer to "lattice polytopes over $\mathbb{Z}^n$", as this is the only class we are interested in.

Let $\mathcal{R}$ refer to the set of all reflexive polytopes, and . It is clear that $\cong$ is an equivalence relation on $\mathcal{R}$. We can define the isomorphism decision problem over reflexive polytopes as:

Definition (Reflexive Polytope Isomorphism (Rpi))**.** Given $P, Q \in \mathcal{R}$, output `true` iff $P \cong Q$ and `false` otherwise.

Now, the work in [KrSk] has found representatives of all isomorphism classes of reflexive polytopes of dimension up to 4, and thus this gives an important ancilliary problem:

Definition (Reflexive Polytope Classification (Rpc))**.** Given a tuple $(P_1, ..., P_N) \subset \mathcal{R}$ such that:

$$\forall i, j \in \{1, ..., N\}, \quad P_i \cong P_j \leftrightarrow i = j \qquad\qquad \forall P \in \mathcal{R}, \exists k \in \{1, ..., N\} \quad s.t. \quad P \cong P_k$$

Output the index into the tuple k for the arbitrary polytope $P \in \mathcal{R}$

It is clear that any oracle for deciding the reflexive polytope isomorphism problem immediately yields an algorithm for deciding the classification problem - simply iterate over the tuple, querying the oracle to determine whether $P \cong P_i$, for each i. However, this is a process with a time and space complexity in $O(N)$ (assuming constant-time oracle queries). In the 4-dimensional case, however, $N = 473, 800, 776$, thus this algorithm is unacceptably slow for any practical use-case.

10.1 Normal Forms

Given two polytopes P, Q, we wish to find a "normal form mapping" $f : \mathcal{R} \to ?$ such that $f(P) = f(Q)$ iff $P \cong Q$. If we can find this successfully, we will be able to - given an oracle for the isomorphism problem - be able to solve repeated queries of the classification problem to the same tuple in a much faster asymptotic time complexity. The standard use-case of polytope classification is to repeatedly index polytopes into the canonical PALP [PALP] database rather than changing the tuple on each query, thus this would be an incredibly useful improvement. Given a normal form mapping, f, by computing $f(P_i)$ for each $i \in \{1, ..., N\}$, indexing these using a fast-lookup system and then storing this as $\mathcal{D}$, we can compute the isomorphism class of P by simply finding the index of $f(P)$ in $\mathcal{D}$. Using a search tree would yield, for each query after the first, a complexity of $O(\log N)$, or a Hash Table to yield an amortized $\Theta(1)$ complexity. In both cases, the first query will require $O(N)$ to build $\mathcal{D}$, but this can be precomputed at any time.

Clearly, normal form mappings are in a one-to-one correspondence with injective mappings $\phi : \mathcal{R}/ \cong \to$?, and it is this conceptualisation that we will use, following the algorithm used in PALP. Since this is fundamentally a linear-algebraic question, our first step is to represent the polytope P as a matrix. We define a map:

$$V_P = [\mathcal{V}(P)] \in \mathbb{Z}^{n \times |\mathcal{V}(P)|}$$

i.e. the columns of $M(P)$ are the vertices of P. This is called Since lattice polytopes can be defined as the convex hull of their vertices, $M(P)$ is injective.

Remark. It is clear that if $P \cong Q$, $|\mathcal{V}(P)| = |\mathcal{V}(Q)|$ by the injectivity and linearity of the automorphism between them. Thus, since we are discussing the problem of algorithms to solve the isomorphism/classification problem, we will use n_v to denote the number of vertices of the polytopes involved, and it can be assumed that each polytope has the same number of vertices.

Now that we have expressed a polytope as a matrix, there is a clear candidate for our normal form map. The **Hessian Normal Form** is an analogous construction to the RREF, except on integer matrices. Letting $H(M)$ denote the HNF of M, we have the essential property that:

$$\forall B \in GL_n(\mathbb{Z}),\ \ H(M) = H(BM)$$

However, permuting $\mathcal{V}(P)$ will in turn permute the rows of $M(P)$ and thus change $H(M(P))$. Since we cannot guarantee that permuting vertices of P will lead to permutations of the rows or columns of $H(M(P))$, we introduce the **vertex-facet pairing matrix**:

Definition. Let $v_i := \mathcal{V}(P)_i$, with $\{w_i, c_i\} \in \Lambda^* \times \mathbb{Z}$ defining the supporting hyperplanes of P - each w_i is a primitive normal vector to the facet F_i of P, such that:

$$\forall v \in \mathcal{V}(P),\ \ \langle w_i, v_i \rangle = -c_i$$

The pairing matrix is thus defined:

$$PM(P)_{ij} := \langle w_i, v_j \rangle + c_i$$

i.e., PM_{ij} corresponds to the height in the lattice of v_j above F_i

By elementary results about lattice heights, this is invariant under the action of $GL(\mathbb{Z})$. Furthermore, it's quite clear that permuting the vertices of P will lead to a permutation of the columns of PM, with permuting the facets yielding a permutation of the rows.

The key observation now is that, since we have a matrix whose entries are invariant up to permutation under isomorphic transforms, and whose permutations are in one-to-one correspondence with permutations of the vertices of P, we can define a canonical ordering of V_P through finding the permutation which minimizes the lexicographic order of $PM(P)$. Since every $P' \cong P$ will have the same minimal lexicographic pairing matrix, this permutation is invariant under isomorphism class. We will not describe the precise algorithm here, but this, along with a more detailed description of the entire process, is available [Normal]. We define

$$NF(P) := H(\pi(V_P))$$

Where π is the canonical permutation, and conclude:

Theorem 6. For $P, Q \in \mathcal{R}$:

$$P \cong Q \leftrightarrow NF(P) = NF(Q)$$

10.2 Naive Machine Learning Classification

Note: All the code implementing methods described in this section is available at https://github.com/arcayn/M2Rfinal. Please check the first commit for the code as written before submission, as it may be cleaned up in the future

Now, it is clear that approaches to solve the problem of Reflexive Polytope Classification have a deep relationship to a number of important questions in mathematics, especially in linear algebra from our discussion of normal forms, but also to combinatorics as we will explore later. When approaching questions of this form, it is a natural approach to consider what insights may be accessible through the application of deep learning techniques to the classification problem, and this will constitute the remainder of our investigation. We have then two main objectives:

- To investigate machine learning architectures which have an affinity for deriving partial solutions to Rpc given only a non-normalised representation of the polytopes, and gather results which may be used as a starting point for new directions of future research in the fields of both ML and algebra. There is currently heavy precedent for the application of ML to algebraic problems leading to new intuition, with a groundbreaking paper published in Nature [Davies] illustrating the use of deep learning saliency maps to guide human mathematicians in solving problems in representation theory.
- To investigate alternatives to deterministic solutions to Rpc that require an (at present) unavoidable $O(N)$ space complexity for storing polytope data - whether in the form of a small reduction in asymptotic complexity or its constant coefficients.

Our first goal is to verify that there is some promise for ML approaches to work in this context, and to this end we will cover two important classifier constructs. Since we are working with a classification problem for highly discrete data, we will not investigate any deterministic regression-based approaches such as Support Vector Machines, and instead begin with approaches more suited to this specific task.

10.2.1 Multi-Layer Perceptrons

The quintessential neural network architecture is the feed-forward **Multi-Layer Perceptron (MLP)**, also known as a **dense** or **fully-connected** network. The intuition behind a perceptron is very simple, and is designed to mimic as closely as possible the functioning of a biological neuron. Much like a logic gate, it has a number of input wires and output wires. Each input wire X_i is assigned a weight, w_i, and the output of the perceptron is given by:

$$\phi\left(\sum_i X_i w_i\right)$$

Where ϕ is known as the **activation function**. In the original proposal for the perceptron (known as a *linear threshold unit*) ϕ is the Heaviside step function, but it may be chosen arbitrary, although most applications will use one of a handful and today Heaviside is rarely used. From the base component of the perceptron, we can build an intuitively simple emulation of the way the human brain is known to work. We simply create layers (rows) of perceptrons, with each perceptron in layer i receiving as input the output wires of each perceptron in layer $(i-1)$. This is from where the name *fully-connected* is derived - every "neuron" is connected to every other neuron in the layers on either side. The input of the MLP corresponds to the input of the first layer *(input layer)* and the output to the output of the final layer *(output layer)*.

With the framework of the MLP in place, it is clear that the natural application of ML to this architecture is to apply optimisation routines to "learn" the weights given to each wire. There are further parameters that can be trained, including adding constant "bias" neurons to each layer, and optimising the output values of these. This is done through a method known as *backpropagation*. We will not describe it in depth [OReillyML] but the basic principle is to apply gradient descent-based optimisation to a differentiable **cost function** which approximates (non-differentiable) prediction accuracy, and "propagating" this through the network, using descent on the outputs of each neuron to assess how the input weights should be adjusted. It is for this reason that Heaviside is rarely used as an activation function, since it cannot be used in conjunction with gradient optimizers. We introduce three important parameters:

- **Batch size:** To reduce overfitting and other undesirable effects, we split the training data into batches and the descent optimiser will collect its data by processing a batch before making adjustments. We want to achieve a middle-ground between very short-sighted optimisations based on small batches, or very imprecise optimisations that will be heavily susceptible to overfitting based on large batches.
- **Epoch count:** The optimiser processes every batch (i.e. the entire dataset) a certain number of times (*epochs*). Too few epochs will not allow for convergence, leading to underfitting, and too many epochs will result in very inflexible rules, leading to overfitting.
- **Learning rate:** The rate at which the optimizer will adjust parameters during training. Too large and the optimizer will bypass local extrema and diverge, but the smaller it is the more epochs are required for acceptable convergence (which will slow down the process)

We will not describe any implementation details of our experimentations on our problem with basic MLP architectures, for reasons that will become apparent in the next section. The most compelling result is a failure to obtain any better-than-random predictions using vertices which are permuted as well as mapped by isomorphism, but a 97% accuracy rate when the vertex data was augmented with connectivity data, and vertices were not allowed to be permuted. Our analysis of these results will come at the end of the report.

10.3 Convolutional Neural Networks

The main class of classifier neural network in active development today is the **Convolutional Neural Network**, a generalisation of the fully-connected MLP design. It's a very intuitive generalisation, and works groundbreakingly-well for image-recognition, the task for which it was originally designed, but have been extended to many more use-cases. Convolutional networks are fundamentally composed of three types of layer:

- **Convolutional:** These are the most important layers in CNNs. They take a tensor as input, but instead of being fully connected, each neuron in a convolutional layer is connected to a **filter** or **kernel** - a hypercuboid subset of the previous layer's neurons. Convolutions may be done in arbitrary dimension, but with 2-dimensional input this corresponds to a square. The intuition behind a convolutional neural network is that by specialising filters on certain areas of a (2-dimensional) image, we can develop a more effective hierarchy of input weightings and layering - each subset of neurons focuses on specific features of the image, and each layer becomes more "big-picture" focused than the last, since the early layers will take their inputs from very small "details" of the input, and the later layers will take their inputs from amalgamations of details which grow in size.

  Another important feature is the notion of *channels*. Again, this etymology comes from separating RGB channels in an image to be classified. Quite simply, data is of a dimension one greater than one might expect - e.g. $3 \times 256 \times 256$ for a 256x256 image. The output from the convolution layers is also separated into channels - each neuron in a channel uses the same filter (of course translated based on neuron position), whereas each channel uses a different filter. In this way, layers can process multiple filters at once, each encoding different features of the data.

  Now filters become another trainable parameter, and the optimization function will also adjust the shape and translation of filters to adjust the network's behaviour from, again, more of a structural, "big-picture" perspective.

- **Pooling:** Pooling layers are similar to convolutional layers in that they too employ filters, but they do not have weights or activation functions, rather the output of a neuron in a pooling layer is simply a constant aggregator function over the neurons in its filter - maximum, mean and sum are common. In addition, the filters from pooling neurons are generally disjoint, or have minimal overlap. In this way, they introduce **downsampling**, decreasing the output tensor size by, for example, averaging over a $3 \times 3 \times 3$ cube, and thus downscaling each dimension by 3. These are critical in CNNs, and allow for much more efficient feature mapping by adding a threshold whereby either node data is pooled at little loss of accuracy - since they only occur after one or more convolutional layers, which has already worked to assign priorities to features - or nodes which are not contributing heavily to the model are stripped away.

- **Dense:** In general, a CNN does not solve classification problems using convolutional layers alone. Instead, they can be seen as *feature mappings*. The convolutional layers take the input tensor and transform it into a tensor of different dimension where the features from the original input are combined, weighted, and in some cases eliminated, based on their influence in the output. The final step is to flatten the multi-dimensional tensor output of the convolutional layers, before using this as input into a standard MLP. This is the part of the network which actually performs the classification, and it of course made up exclusively of dense layers

However, applying convolutional networks to our approach yields no useful classification beyond random guessing. Why might this be? It is a well-researched phenomenon [IsoNN] that the problem of classifying polytopes or other "unordered" data (i.e. since there is no immediate ordering of the vertices consistent across the input data) is very difficult with traditional deep-learning architectures. The primary cause of this can be seen directly by considering the MLP. Consider an input vector:

$$\mathbf{v} = (v_1, ..., v_n) \in (-1, 1)^n$$

We denote by w_{ij} the weight given to the j-th neuron in the input layer by the i-th neuron in the first hidden layer. Consider that the output vector of the first layer is given by:

$$\begin{pmatrix} w_{11} & ... & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & ... & w_{mn} \end{pmatrix} \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix}$$

Where m is the number of neurons in the hidden layer. Now if we choose some random permutation of the input, we can represent this by an $n \times n$ permutation matrix P, and thus given instead the input vector Pv, the output of this layer will be given by:

$$(w_{ij})P\mathbf{v}$$

Which is precisely equivalent to permuting the columns of the weights vector. By expressing the permutation in these terms, we can obtain that attempting to learn classification on an MLP through random permutation of inputs is exactly equivalent to the problem of constructing a learning schema for an unstable MLP which essentially has unordered weights in the input row - that is to say, we cannot weight particular neurons, only choose the proportions of weights which will be assigned to any neuron. Furthermore, we can apply the exact same reasoning to a convolutional network, since they can be modelled as dense networks with particular weights set to a constant value of zero. Furthermore, CNNs are actually more susceptible to disruption by permutation, since the contiguous filters are split across the image in unpredictable ways

We can conjecture that there are a number of ways to attack the problem of permutation invariance:

- By using nonlinear normalisation functions after the first output layer, we can break the linear relationship which leads to this equivalence between input and weight permutation, and begin to recover some information about proportional weightings in the training data. We can see that this is equivalent to considering a new network, where the original input and first hidden layers are merged into a new input, however it may still preserve some salient feature-mapping. Using non-standard **feedback** neurons, where output from the second layer is used as input to the first may further aid this approach.
- Clearly, we can aggregate data over the entire input vector and set every weight in the first layer equal, which will completely subvert this problem and in fact supply *permutation invariance*, however this will only work with incredibly basic data.
- It may be possible to take advantage of the fact that the filters of the convolutional layers are scattered, and look for relationships be

We were able to achieve some better-than-random guessing accuracy results by using fragile permutation-invariant heuristics, including up to nearly 30% with a combination of lexicographic ordering and augmenting the vertex matrix with node adjacency data. However, it is clear that any attempt to make a permutation invariant network will inevitably lead to a significant departure from any currently-considered architecture.

10.4 Graph Neural Networks

Since their initial proposal in 2009 [GNN], the concept of a **Graph Neural Network** has gained significant traction in the Deep-Learning classifier research community. In general, the term GNN simply refers to any neural network specifically designed to operate on graphs, thus leading to networks which take two parameters as input:

- **Vertices:** The vertex (ordered) set of the graph is given by the tuple:

$$V = (\mathbf{v}_1, ..., \mathbf{v}_{v_n})$$

Each $\mathbf{v}_i \in \mathbb{R}^m$ are vectors of the same dimension, representing a vertex and the feature data assigned to them

- **Edges:** The edge set of the graph is given by the tuple:

$$E = ((x_1, y_1, \mathbf{e}_1), ..., (x_{n_e}, y_{n_e}, \mathbf{e}_{n_e}))$$

  In this case, each triplet represents an edge, with the (x_i, y_i) pair representing its endpoints, and $\mathbf{e}_i \in \mathbb{R}^m$ the vector of features assigned to each edge.

This is a specialisation of neural networks which accept arbitrary tensors, such that layers are constructed with the express purpose of maximising the utility of each individual feature set (edge and vertex), with added precautions to ensure that Graph neural networks have two main applications:

- **Node-Level Learning:** In the most domain-specific application, we take a **semi-supervised learning** approach; given a graph with some labelled nodes and some unlabelled nodes, deduce the missing labels. This approach differs from the **supervised** train/predict model, since we are given explicit connections between our training and testing data which can be exploited

- **Graph-Level Learning:** This is the direct analogue to the previous architectures we have considered - we are given training data in the form of graphs, and must classify them

10.4.1 Permutation invariance

We can apply the exact same principles as in a standard convolutional network to a graph network - conceptualising a standard convolutional network as a graph network which acts only on lattice graphs - i.e. multidimensional grids where each interior node has exactly $2n$ neighbours in the n-dimensional case. Instead of considering small hypercuboidal sections of the input data, we can instead consider small arbitrary subgraphs of the input data. In this case, we can conceive of GCNNs as analagous to special cases of convolutional networks, where vertex data and edge connectivity is separated into the channels of the convolutional layers. Since GCNNs are specifically designed with orderless data in mind, they exhibit incredibly strong, but not perfect, permutation invariance. Our final architecture is given by:

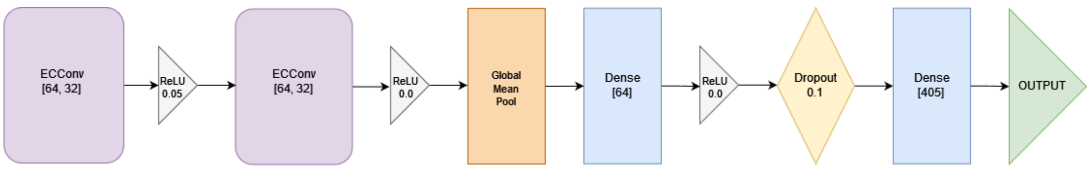


Figure 5: Graph Neural-Network architecture

This is based on the Edge-Conditioned Convolutional layer, denoted by ECConv and detailed in [ECC]. It consists of two graph convolutional layers, followed by an average pooling layer and a final, simple, dense network. After a number of test-train iterations, we managed to achieve a classification accuracy of 81.15%, and conjecture based on the performance of convergence that if resource limitations were lifted, we would be able to push this classification rate even higher.

10.4.2 Evaluation

Whilst we were initially excited at the incredibly high accuracy rate of the GNN approximating highly nontrivial functions and the implications which this may have for extending the learned feature mappings of the network beyond the scope of the project to general problems in combinatorics and linear algebra, we conjecture that the most likely learned techniques by the network is the following

- Graph Neural Networks, and the ECConv design in particular, are specifically designed with permutation invariance in mind. In this way, they attempt to probabilistically generalise the first step of creating a normal form - that is, the construction of the Vertex-Facet pairing matrix and derivation of a canonical vertex matrix permutation from this.

- When converting a polytope to a graph, a crucial piece of information preserved is the adjacency matrix of the polytope. This is clearly invariant under isomorphism, and in fact 94% of three-dimensional reflexive polytopes have distinct adjacency matrices, and of those which overlap, there is no adjacency matrix shared by more than two equivalence classes. This is thus, at least in three dimensions, a very good approximation of the Hermite Normal Form as an injective function constant over isomorphism classes. We note that 94% of equivalence classes having distinct adjacency matrices exactly matches up with our results from non-permuted vertices in MLPs, since the probability of success based on adjacency matrix alone is:

  $$0.94 * 1 + 0.06 * 0.5 = 0.97$$

  Since in the case where the adjacency matrix is distinct, it is exactly a choice between two classes. We thus hesitantly conjecture that the MLP was unable to derive any useful data from the vertices alone, and instead entirely weighted its predictions based on the adjacency data, since this is consistent with the expected accuracy value.

In conclusion, future work will certainly consist of investigating the possibility of extending the adjacency matrix-based pseudo-normal form to 4-dimensions, and the implications there, as well as any results that may indicate the possibility for neural network learning of data compression of the polytope database - since even in the most economical and minimal vertex encoding possible (without compression being employed), it still stretches into the range of 100+GB.

It would also involve applying Machine Learning techniques to polytope classification through the medium of toric varieties and other associated algebraic constructs immediately - seeking to perhaps deduce some novel connections which are missed by conversion into polytope representation for ease of deterministic calculation.

11 Conclusion

During the project, we have read through lattices, reflexive polytopes, dual operations, toric varieties, fans, and normality of fans. We have also investigated the basics of machine learning and various algorithms that may contribute to the project. Combining both aspects of knowledge, we came to accomplish a classifier with a notable accuracy of 82.15 percent.

Admittedly, the research on algebraic geometry went beyond the actual content we applied in our final coding, which is, indeed, a pity caused by the limited time frame. On the other hand, this difference matches the circumstance in a more general world of academic. Driven by enthusiasm, mathematicians have kept going beyond the boundary of applications for hundreds of years. Following this pattern of enthusiasm, we come to the final stage of being confident to claim that a never discovered connection between pure mathematics and machine learning is revealed. We believe this research project is inspirational to future study aiming to improve the classifier to a higher dimension with a greater accuracy and encouraging for ambitions of employing machine learning to other suitable areas of mathematics.

References

[1] Miles. Reid (1988) *Undergraduate Algebraic Geometry*, Cambridge University Press.

[2] Alexander M. Kasprzyk, Benjamin Nill (2011) *Reflexive Polytopes of Higher Index and the Number 12*, arXiv:1107.4945v1

[3] Simon Telen (2022) *Introduction to Toric Geometry*, arXiv:2203.01690v1

[4] Jean-Paul Brasselet (2006) *Introduction to Toric Varieties*, Institut de Mathematiques de Luminy

[5] David A. Cox, John Little, Hal Schenck (2010) *Toric Varieties*, https://www.mimuw.edu.pl/ jarekw/pragmatic2010/CoxLittleSchenckJan2010.pdf

[6] Bjorn Poonen, Fernando Rodriguez-Villegas (2000) *Lattice Polygons and the Number 12*, The American Mathematical Monthly, Vol. 107, No. 3

[7] Gunter M. Ziegler (1998) *Lectures on Polytopes*, Springer Graduate Texts in Mathematics

[8] Benjamin Nill, *Reflexive Polytopes - Combinatorics and Convex Geometry*, https://staff.math.su.se/nill/daten/talk_kreta.pdf

[9] John Terilla (2014), *Zariski Topology*

[10] Dongmin Roh (2016), *Reflexive Polygons and Loops*, https://math.uoregon.edu/wp-content/uploads/2017/10/RohThesis2016.pdf

[11] Frank Sottile (2003), *Toric ideals, real toric varieties, and the algebraic moment map*, Topics in Algebraic Geometry and Geometric Modeling

[12] Oscar Zariski (1939), *Some Results in the Arithmetic Theory of Algebraic Varieties*, American Journal of Mathematics, Vol. 61, No. 2

[13] Wanlong Zheng, *Integral closure and normalization*, https://www.dpmms.cam.ac.uk/ wz302/randDoc/intClosure.pdf

[14] Alexander M. Kasprzyk, Benjamin Nill (2011), *Reflexive Polytopes of Higher Index And The Number 12*, arXiv:1107.4945v1

[15] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini, "The Graph Neural Network Model," in IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61-80, Jan. 2009, doi: 10.1109/TNN.2008.2005605.

[16] Kreuzer, Maximilian, and Harald Skarke. "Complete Classification of Reflexive Polyhedra in Four Dimensions." arXiv.org, February 28, 2000. https://doi.org/10.48550/arXiv.hep-th/0002240.

[17] http://hep.itp.tuwien.ac.at/ kreuzer/CY/CYpalp.html

[18] Davies, A., Veličković, P., Buesing, L. et al. Advancing mathematics by guiding human intuition with AI. Nature 600, 70–74 (2021). https://doi.org/10.1038/s41586-021-04086-x

[19] https://doi.org/10.48550/arXiv.1704.02901

[20] https://doi.org/10.48550/arXiv.1301.6641

[21] GeÌ ron, Aureì lien. Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. 2nd ed., O'Reilly, 2019.

[22] https://doi.org/10.48550/arXiv.1907.09495