2017

Toric Ideals, Polytopes, and Convex Neural Codes

Caitlin Lienkaemper
Harvey Mudd College

Toric Ideals, Polytopes, and Convex Neural Codes

Caitlin Lienkaemper

Mohamed Omar, Advisor

Dagan Karp, Reader

Department of Mathematics

May, 2017

Abstract

How does the brain encode the spatial structure of the external world? A partial answer comes through place cells, hippocampal neurons which become associated to approximately convex regions of the world known as their place fields. When an organism is in the place field of some place cell, that cell will fire at an increased rate. A neural code describes the set of firing patterns observed in a set of neurons in terms of which subsets fire together and which do not. If the neurons the code describes are place cells, then the neural code gives some information about the relationships between the place fields–for instance, two place fields intersect if and only if their associated place cells fire together. Since place fields are convex, we are interested in determining which neural codes can be realized with convex sets and in finding convex sets which generate a given neural code when taken as place fields. To this end, we study algebraic invariants associated to neural codes, such as neural ideals and toric ideals. We work with a special class of convex codes, known as inductively pierced codes, and seek to identify these codes through the Gröbner bases of their toric ideals.

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I'd like to thank my advisor Mohamed Omar for his suggestions of research directions and his consistently helpful advice. I'd also like to thank Anne Shiu for introducing me to convex neural codes during the Algebraic Methods in Computational Biology REU at Texas A&M University during summer 2015.

Beyond this, I'd like to thank my friends at Mudd for supporting me through this semester. I'd like to thank Micah Pedrick for many kinds of support, and in particular for close proofreading. I'd also (of course) like to thank my parents for supporting me throughout life.

Chapter 1

Background

1.1 Neural Codes and Convexity

Suppose we have recorded the behavior of three neurons over a period of a few minutes. We assume that at each time, the state of a neuron can be described as either "on" or "off". This is a reasonable assumption since neural activity is well described by action potentials–when the voltage across the cell membrane crosses a certain threshold, it will rapidly spike to a higher voltage before rapidly returning to the baseline voltage, thus there is a clear yes/no answer to the question, "did this neuron fire in this one second interval?" We label our cells one, two, and three. At various times in our interval, we sometimes find that all three cells are firing, sometimes find that cells one and two are firing, sometimes find that cells one and three are firing, sometimes find that cell one or cell two is firing alone, and sometimes find that no cells are firing. We can record this behavior as a neural code. A codeword is a binary vector of length n that describes the states of n neurons (firing or not firing) at a given moment. A neural code is a collection of codewords which describes all states a collection of n neurons takes over a period of time. More formally:

Definition 1.1 (neural code)**.** *A codeword is a binary vector $c \in \mathbb{F}_2^n$. A neural code is a set of codewords, and therefore is a subset $\mathcal{C} \subset \mathbb{F}_2^n$.*

For instance, the neural activity described above corresponds to the neural code

$$\mathcal{C} = \{(1,1,1),(1,1,0),(1,0,1),(1,0,0),(0,1,0),(0,0,0)\} \subset \mathbb{F}_2^n.$$

The codeword (1,1,1) describes all thee neurons firing, the codeword (1,1,0) describes just neurons one and two firing, the codeword (1,0,0) describes neuron one firing alone, and the codeword (0,0,0) describes no neuons firing. For convenience, we typically discard vector notation for simpler binary notation, writing C instead as

$$C = \{111, 110, 101, 100, 010, 000\}.$$

Instead of defining the neural code in terms of binary vectors, it is sometimes convenient to describe the neural code in terms of sets of neurons which fire together.

Definition 1.2 (support)**.** *The support of a codeword* $c = (c_1, \ldots, c_n)$ *is the set* $\{i \mid c_i = 1\}$*. The support of a neural code* C *is* $\operatorname{supp}(C) = \{\operatorname{supp}(c) \mid c \in C\}$*.*

For the neural code above,

$$\operatorname{supp}(C) = \{123, 12, 13, 1, 2, \emptyset\}.$$

In the larger neuroscience literature, such codes are called combinatorial neural codes, since they discard precise data about the timing and rate of neural activity and retain only combinatorial information about which subsets of cells fire together.

In general, patterns of neural activity do not occur in a vaccum. Instead, our brains use these patterns to encode information about the external world and our relation to it. This thesis is motivated by neural codes arising from place cells, neurons which form part of the "neural GPS system". In 2014, John O'Keefe, Evard Moser, and May Britt Moser were awarded the Nobel Prize in Physiology or medicine for their discovery of the fascinating behavior of these cells and a related class of cells known as grid cells (Burgess (2014)). Taking advantage of technology which was revolutionary at the time, O'Keefe recorded the activity of single neurons in the hippocampus, the part of the brain involved in consolidation of memory and spatial navigation, in freely moving rodents. He discovered a class of neurons with surprising behavior–when the rodent was in a specific part of its enclosure, the neuron he was monitoring would fire at a high rate. Otherwise, it would fire slowly or not at all. Thus he named these neurons place cells. In familiar environments, these place cells become associated to regions of space known as their place fields or receptive fields. Since then, place cells have been observed in a number of species, including humans (Burgess and O'Keefe (2003)). If we, as researchers, have access to locations of place fields, then we can determine where the animal is in its environment by monitoring place cell activity. However, this is not what the brain is doing: since the brain does not have an external record of place field locations, all information the animal has about the environment and its place within it must be somehow recorded in patterns of neural activity (but not necessarily place cell activity). Thus, we are interested in determining how much spatial information can be recovered from the intrinsic structure of neural codes which arise from place cell activity. A more thorough account of place cells and their behavior can be found in Burgess (2014).

Experimentally, place fields are observed to be approximately convex. That is, any straight line between two points in a place field is fully contained within the place field. To avoid pathological behaviors such as two dimensional place fields which intersect at a single point or along a line, we generally require that place fields be open sets, though there has been some investigation into models with closed convex place fields. In general, we are trying to avoid codewords which correspond to sets of measure zero–this is because if there is a region corresponding to a set of measure zero, then the probability that the set of neurons corresponding to this region will fire together is zero, so we will not actually include this codeword in our neural code.

Given a collection of sets $U_1, \dots, U_n$, we can find the neural code C they would generate when taken as place fields as follows:

For each $\sigma \subset [n]$,

$$\bigcap_{i \in \sigma} U_i \setminus \bigcup_{i \in [n] \setminus \sigma} U_i \neq \emptyset,$$

then $c = (c_1, \dots, c_n)$ with $c_i = 1$ of and only if $i \in \sigma$ is in C. If C is the code a collection of sets $\mathcal{U} = \{U_1, \dots, U_n\}$, then we say $C = C(\mathcal{U})$. Another way to describe this is to label each region wth the set of place fields it is contained in, and then let the neural code be the set of all these labels. Inspired by this, we say that $U_1, \dots, U_n$ is a realization of a neural code C if C is the neural code $U_1, \dots, U_n$ generate. In other words,

Definition 1.3 (realization)**.** *Let C be a neural code. Then $U_1, \dots, U_n$ is a realization of C if $c \in C$ if and only if*

$$\bigcap_{c_i = 1} U_i \setminus \bigcup_{c_i = 0} U_i \neq \emptyset.$$

Since place fields are approximately convex, we are interested in determining which neural codes can be realized with convex sets. We define open and closed convex neural codes, beginning with definitions of convex, open, and closed sets.

Definition 1.4 (convex). *A set $A \subset \mathbb{R}^n$ is convex if for all points $a, b \in A$, the line segments between a and b stays within A.*

Figure 1.1 The right two sets (red) are convex. The left three sets (blue) are not, as the pictured line segments demonstrate.

Definition 1.5 (open). *A set $U \subset \mathbb{R}^n$ is open if for each $u \in U$, there exits an $\epsilon > 0$ such that if the distance $d(u, x) < \epsilon$, then $x \in U$.*

Definition 1.6 (closed). *A set $A \subset \mathbb{R}^n$ is closed if its complement is open.*

For instance, the set $(a, b) := \{c \in \mathbb{R} | a < c < b\}$ is open, the set $[a, b] := \{c \in \mathbb{R} | a \leq c \leq b\}$ is closed, and the set $(a, b] := \{c \in \mathbb{R} | a < c \leq b\}$ is neither open nor closed. Both the empty set and the whole real line are both open and closed when viewed as subsets of $\mathbb{R}$.

Definition 1.7 (convex neural code). *A neural code $\mathcal{C}$ is open convex with embedding dimension d if there exists a realization $U_1, \ldots, U_n$ of $\mathcal{C}$ such that $U_1, \ldots, U_n$ are open convex subsets of $\mathbb{R}^d$. Likewise, a neural code $\mathcal{C}$ is closed convex with embedding dimension d if there exists a realization $U_1, \ldots, U_n$ of $\mathcal{C}$ such that $U_1, \ldots, U_n$ are closed convex subsets of $\mathbb{R}^d$. In either case, the smallest such d minimal open (respectively, closed) convex embedding dimension of $\mathcal{C}$.*

For instance, we give a convex realization of the code $\{000, 100, 010, 110, 101, 111\}$ in Figure 1.2.

A number of natural questions in neuroscience and math arise from these definitions.

1. Given a neural code, is there an (efficient) algorithm to determine whether it is convex?

2. Is there a way to construct realizations of convex codes?

3. What are the algebraic and combinatorial signatures of convexity?

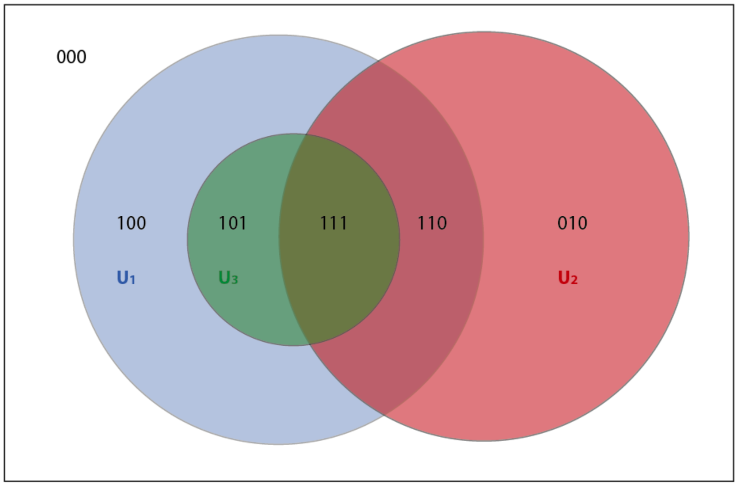

Figure 1.2 A convex realization of the code $C = \{000, 100, 010, 110, 101, 111\}$. If U_1, U_2, and U_3 were taken as place fields, the activity of their corresponding place fields would be the neural code C.

4. Given a neural code, is there an (efficient) algorithm to determine the embedding dimension?

While these questions are mathematically interesting in their own right, all have implications in neuroscience. While we motivated the introduction of convex neural codes through place cell codes, convex coding is a plausible paradigm for representing other types of information. For instance, there are cells in the visual cortex which respond to the orientation of objects in the environment. Like place cells, these cells have receptive fields which are approximately convex sets of angles. In order to study convexity elsewhere in the brain where the stimulus space is not well understood, such as the olfactory system, we need to characterize convexity in terms of the intrinsic structure of the neural code, rather than by reference to a realization in the stimulus space, in order to determine whether convex coding plays a role.

Further, if we wish to understand how convex codes actually arise in the brain, we wish to characterize the neural networks which give rise to convex neural codes. Simple algebraic and combinatorial signatures of convexity would simplify this problem significantly.

1.2 Summary of Previous Work

There has been a body of work focused on convex neural codes which be roughly divided into algebraic and topological or geometric lines of research. This thesis will primarily use algebra, thus a summary of algebraic work on neural codes is given, along with background material in algebra and algebraic geometry, in Section 1.3. Here, we present a summary of progress made using topological and geometric methods. The major object of study in topological or geometric approaches to convex neural codes is the simplicial complex of a code, defined as follows:

Definition 1.8 (Abstract simplicial complex)**.** *An abstract simplicial complex is a collection Δ of subsets of a vertex set, called simplices, which we take to be $[n] = \{1, 2, \ldots, n\}$, such that if $\sigma \in \Delta$ and $\tau \subset \sigma$, then $\tau \in \Delta$.*

Definition 1.9 (Simplicial complex of a neural code)**.** *Let $\mathcal{C}$ be a neural code. The simplicial complex of $\mathcal{C}$ is the minimal simplicial complex $\Delta(\mathcal{C})$ such that $\operatorname{supp}(\mathcal{C}) \subset \Delta(\mathcal{C})$.*

Definition 1.10 (Geometric realization)**.** *The geometric realization of an n-simplex (a simplex with $n-1$ vertices) is the convex hull of n affinely independent points in $\mathbb{R}^n$. For instance, a 0-simplex is a point, a 1-simplex is a line segment, a 2-simplex is a triangle, and a 3-simplex is a tetrahedron. If Δ is an abstract simplicial complex, then the geometric realization is a collection of the geometric realizations of simplices of Δ on the same vertex set.*

Perhaps the most interesting topological result on convex neural codes is that of Curto and Itskov (2008), which proves that through the nerve theorem, the stimulus space covered by a collection of convex place fields is homotopy equivalent to the geometric realization of the simplicial complex of the neural code, thus stimulus space topology can be recovered from the neural code alone. The authors of Curto et al. (2017) defined a local obstruction in terms of the topology of $\Delta(\mathcal{C})$, and proved that open and closed convex codes have no local obstructions, and that, on up to four neurons, all codes with no local obstructions are convex open. However, Lienkaemper et al. (2017) proved that the converse is not true in general via a counterexample on five neurons. Cruz et al. (2016) show that there does exist a closed convex realization of this code and give an example of six neuron code which is open convex but not closed convex. Furthermore, Cruz et. al show that codes which contain all intersections of maximal faces are both open and closed convex. Therefore, the only codes whose convexity cannot be determined by checking for local obstructions and for the presence of all intersections of maximal faces are those codes which are missing at least one intersection of maximal faces, but do not have local obstructions. Finally, Cruz et. al showed that the set of open convex codes with the same simplicial complex is closed under the addition of codewords–in other words, to find all convex codes with a given simplicial complex, it is sufficient to find the minimal convex codes on that complex.

Despite this progress, there is still no necessary and sufficient condition for a neural code to be convex defined in terms of the intrinsic structure of the code. Furthermore, even when a convex realization is known to exist, it is often in a very high dimension, and it is not known how to determine the minimal convex embedding dimension of an arbitrary convex code. The overall difficulty of the general case motivates us to work in special cases. In Chapter 2, we will define a class of convex codes known as inductively pierced neural codes for which it is very easy to construct low dimensional convex realizations and present some work towards identifying these inductively pierced neural codes.

1.3 Algebraic Geometry Background

In this section, we give the necessary algebraic geometry background and define several algebraic structures associated to neural codes. These definitions are adapted from Dummit and Foote (2004) and Reid (1988).

Structures known as rings generalize the properties of sets such as the integers, where operations of addition and multiplication are well defined and play nicely together:

Definition 1.11 (ring)**.** *A commutative ring with identity is a set R together two binary operations, $+ : R \times R \to R$ and $\times : R \times R \to R$ such that ,*

- $(a+b)+c = a+(b+c)$ *for all $a, b, c \in R$.* (*This is the associative property.*)
- *There exists $0 \in R$ such that $a + 0 = a$ for all $a \in R$.* (*The element 0 is called the additive identity.*)
- *For each $a \in R$, there exists $-a \in R$ such that $a + -a = 0$.* (*The element $-a$ is known as the additive inverse of a.*)
- *For all $a, b \in R$ $a + b = b + a$.* (*This the commutative property of addition.*)
- *For all $a, b, c \in R$, $a \times (b + c) = a \times b + a \times c$* (*This the distributive property.*)

- *There exits* $1 \in R$ *such that* $1 \times a = a$ *for all* $a \in R$. (*The element* 1 *is called the multiplicative identity.*)
- *For all* $a, b \in R$, $a \times b = b \times a$. (*This is the commutative property of multiplication.*)

The most familiar example of a ring is the set of integers under addition and multiplication–one can check that all the properties in Definition 1.11 hold. If a set F is a commutative ring and for every $f \in F$, there exists f^{-1} such that $ff^{-1} = f^{-1}f = 1$, well call F a field and f^{-1} the multiplicative inverse of f. Familiar examples of fields include the real, rational, and complex numbers. The rings we will work with here will primarily be polynomial rings. The elements of a polynomial ring are polynomials in several variables with coefficients in some field k.

Definition 1.12 (polynomial ring)**.** *A polynomial ring in one variable,* $k[x]$, *is the set of formal combinations* $\sum_{i=1}^{m} a_i x^i$ *with* $a_i \in k$. *To define polynomial rings over several variables, let* $x = (x_1, \ldots, x_n)$ *be a tuple of variables and* $\alpha = (\alpha_1, \ldots, \alpha_n)$ *be a tuple of integer exponents and define* $x^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$. *Then the polynomial ring* $k[x_1, \ldots, x_n]$ *is the set of formal combinations* $\sum_\alpha a_\alpha x^\alpha$ *with* $a_i \in k$. *Addition and multiplication are defined as they typically are for polynomial functions.*

For instance, the set $\mathbb{Q}[x_1, x_2, x_3]$ of all polynomials in three variables with rational coefficients forms a polynomial ring which includes elements such as $x_1x_2x_3^3 + \frac{7}{8}x_1x_2^2$ and $x_1^4x_2 + 3x_3$.

A map between rings which respects the additive and multiplicative structures is known as a ring homomorphism.

Definition 1.13 (ring homomorphism)**.** *Let* R *and* S *be commutative rings,* $\phi : R \to S$. *Then* ϕ *is ring homomorphism if and only if for all* $p, q \in R$, $\phi(p + q) = \phi(p) + \phi(q)$ *and* $\phi(pq) = \phi(p)\phi(q)$.

For instance, consider the rings $\mathbb{R}[x]$ and $\mathbb{R}$. For each $a \in \mathbb{R}$, we can define the evaluation at a map $e_a : \mathbb{R}[x] \to \mathbb{R}$ as follows:

$$e_a\left(\sum_{i=1}^{m} b_i x^i\right) = \sum_{i=1}^{m} b_i a^i.$$

The reader is encouraged to check that this is a homomorphism.

We define the kernel and image of a homomorphism $\phi : R \to S$ to be the sets

$$\ker(\phi) = \{\phi(r) = 0 | r \in R\}$$

and

$$\phi(R) = \{s = \phi(r) | s \in S\}.$$

Definition 1.14 (ideal)**.** *Let R be a commutative ring, and $I \subset R$. We say I is an ideal if I is a closed subgroup of R under addition and for all $a \in I$, $b \in R$, $ab \in I$.*

For instance, the kernel of a homomorphism $\pi : R \to S$ is always an ideal of R, since if $r, q \in \ker(\phi)$, then $\phi(p) = \phi(q) = 0$, and since ϕ is a homomorphism, $\phi(p+q) = 0$ and $p + q \in \ker(\phi)$. Likewise, if $p \in \ker(\phi)$ and $r \in R$, then $\phi(pr) = \phi(p)\phi(r) = 0\phi(r) = 0$. If we have a set of elements $S \subset R$, we can consider the smallest ideal containing them, $I = \langle S \rangle$. We say that S *generates* or is a *basis* for $\langle S \rangle$. For instance, if we take $2 \subset \mathbb{Z}$, then $\langle 2 \rangle = 2\mathbb{Z}$, the ideal of even integers. As another example, the ideal $\langle x^2, y \rangle \subset k[x, y, z]$ is the set of all polynomials in x, y, and z such that each term is divisible by either x^2 or y. For instance, this ideal contains the polynomials $x^2 + y$, $xyz + y^2$, and x^2z, but not $x^2z + x$.

Elementary algebra focuses on graphing the solution sets to polynomial equations. In algebraic geometry, we do much the same thing. We can evaluate any polynomial $f \in k[x_1, \ldots, x_n]$ at any point $v = (v_1, \ldots, v_n) \in k^n$ by substituting the value v_i for x_i. Then we can talk about the set of points v in k^n where $f(v) = 0$ for some $f \in k[x_1, \ldots, x_n]$. We formalize this with the notion of a variety:

Definition 1.15 (variety)**.** *Let I be an ideal of $k[x_1, \ldots, x_n]$. We define the variety*

$$V(I) \stackrel{def}{=} \{v \in k^n | f(v) = 0 \textit{ for all } f \in I.\}$$

For instance, the ideal $\langle y - x^2 \rangle \subset \mathbb{R}[x, y]$ has the variety $\{y = x^2 | x, y \in R^2\}$. Note that this is the graph of the function $y = x^2$ in $\mathbb{R}^2$.

Likewise, if we begin with a set of points $S \subset k^n$, we can define the ideal of functions which vanish on this set as

$$I(S) \stackrel{def}{=} \{f \in k[x_1, \ldots, x_n] | f(v) = 0 \text{ for all } v \in S\}.$$

1.3.1 Gröbner Bases

A *Gröbner basis* is a particular set of generators for an ideal of $k[x_1, \ldots, x_n]$ which can be computed by standard methods and which has some nice properties. In order to define Gröbner bases, we must define term orders, total orderings on the set of monomials in a polynomial ring. If we are working in one variable, this is easy: over $k[x]$, we can define a term order $\prec$ by demanding $x^a \prec x^b$ if $a < b$. How do we generalize this? Define $x^a = x_1^{a_1} \cdots x_n^{a_n}$. Over $k[x_1, \ldots, x_n]$, we say that $\prec$ is a valid monomial order if the following hold:

1. It is multiplicative: that is, $x^a \prec x^b$ implies $x^{a+c} \prec x^{b+c}$ for all $a, b, c \in \mathbb{N}^n$.

2. The constant monomial is the smallest: that is, $1 \prec x^a$ for all $a \in \mathbb{N}^n$.

For instance, a lexicographic order on the monomials in $k[x_1, \ldots, x_n]$ assigns an order to the variables of $x_1, \ldots, x_n$ and extends this to other monomials as follows. To compare monomials $\mathbf{x}^a$ and $\mathbf{x}^b$ we first compare the exponents on x_1 (a_1 and b_1) and say that $\mathbf{x}^a \prec \mathbf{x}^b$ if $a_1 > b_1$, $\mathbf{x}^b \prec \mathbf{x}^a$ if $b_1 > a_1$. If $a_1 = b_1$, we move to comparing the exponents on b_2, and so on. Thus the polynomial $x_1x_2x_3 + x_1^2x_2 + x_3^5$ is written as $x_1^2x_2 + x_1x_2x_3 + x_3^5$ when we order the terms lexicographically.

To obtain a greater diversity of term orders, we consider the class of weighted term orders. Given a weight vector ω and an arbitrary "tie breaker" term order $\prec$, say $\mathbf{x}^a \prec_\omega \mathbf{x}^b$ if either $a \cdot \omega > b \cdot \omega$ or $a \cdot \omega = b \cdot \omega$ and $\mathbf{x}^a \prec \mathbf{x}^b$. For instance, for $\omega = (2, 1, 4)$ and $\prec$ the lexicographic order, we can see that $x_2x_3 \prec_\omega x_1x_2$ and $x_1x_2^2 \prec_\omega x_3$.

Once we have a monomial order, we can pick out the leading term of any polynomial, that is, the term which is first in the term order. For instance, $x_1^2x_2$ is the leading term of $x_1^2x_2 + x_1x_2x_3 + x_3^5$. Define the initial ideal $in_\prec(I)$ of an ideal I to be the ideal generated by the leading terms of all polynomials in I. A subset $\mathcal{G} \subset I$ is a Gröbner basis of I if

$$in_\prec(I) = \langle in_\prec(g) : g \in \mathcal{G} \rangle.$$

A generating set is a *univeral* Gröbner basis if it is a Gröbner basis with respect to any term order.

A more thorough treatment of Gröbner bases can be found in Sturmfels (1996) and Cox et al. (1992). We introduce Gröbner bases here in order to introduce Conjecture 2.3 of Chapter 2 relating to Gröbner bases of ideals associated to neural codes.

1.3.2 The Gröbner Fan

The geometric objects polyhedral cones and fans can give us insight into the ways the Gröbner bases of an ideal under different term orders relate to one another. We take the following definitions from Sturmfels (1996). Roughly, polyhedral cones are the sets of non-negative linear combinations of a set of vectors and polyhedral fans are "nice" collections of cones.

Definition 1.16 (Polyhedral Cone)**.** *A set $P \subset \mathbb{R}^n$ is a polyhedral cone if it can be written in the form*

$$P = \{\lambda_1 u_1 + \ldots + \lambda_m u_m : \lambda_1, \ldots, \lambda_m \in \mathbb{R}^+\}.$$

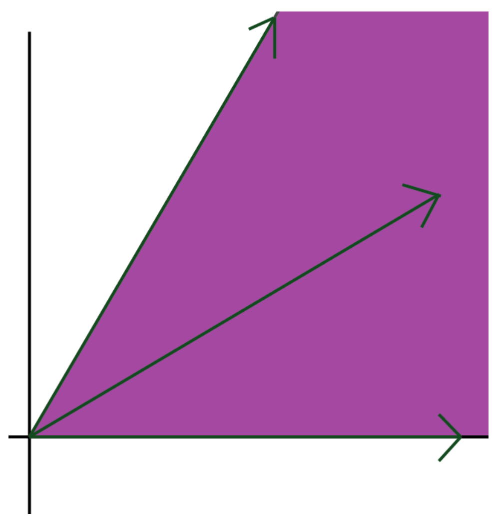

Figure 1.3 A polyhedral cone in $\mathbb{R}^2$, generated by the three green vectors. Note that the middle vector is redundant.

Intuitively, a face of a polyhedral cone is the set of points in the cone which are the most extreme in some direction. Formally, we have:

Definition 1.17 (Face)**.** *A face of a polyhedral cone P is the set $face_\omega(P) :=$ $\{\omega \cdot \mathbf{u} \in P \geq \omega \cdot \mathbf{v}$ for all $\mathbf{v} \in P\}$.*

Definition 1.18 (Polyhedral Fan)**.** *A collection of polyhedral cones Δ is a fan if*:

- *if $P \in \Delta$ and F is a face of P, then $F \in \Delta$*
- *if $P_1, P_2 \in \Delta$, $P_1 \cap P_2$ is a face of P_1 and P_2.*

Note the parallels with the definition of a simplicial complex.

Through the following standard theorems, cones and fans organize information about the set of Gröbner bases of a given ideal. Fixing an ideal

$I \subset k[x_1, \ldots, x_n]$, we define two weight vectors ω and ω' to be equivalent if $in_\omega(I) = in_{\omega'}(I)$. Then the following hold:

Proposition 1. *Each equivalence class of weight vectors is a relatively open convex polyhedral cone.*

The Gröbner fan $GF(I)$ is defined to be the set of closed cones $C[\omega]$ for each $\omega \in \mathbb{R}^n$. Luckily, this terminology is valid:

Proposition 2. *The Gröbner fan $GF(I)$ is a fan.*

For proofs and more details, see Sturmfels (1996).

1.4 The Neural Ideal

1.4.1 Definitions

We will show shortly that any neural code $\mathcal{C} \subset \mathbb{F}_2^n$ is a variety in $\mathbb{F}_2^n$. Thus, we can study this variety by studying its vanishing ideal over the polynomial ring $\mathbb{F}_2[x_1, \ldots, x_n]$. Given any neural code $\mathcal{C} \subset \mathbb{F}_2$, we can find the ideal $I(\mathcal{C})$ of all polynomials which vanish on $\mathcal{C}$.

Definition 1.19 ($I_\mathcal{C}$)**.**

$$I_\mathcal{C} \stackrel{def}{=} I(\mathcal{C}) = \{f \in \mathbb{F}_2[x_1, \ldots, x_n] | f(c) = 0 \text{ for all } c \in \mathcal{C}\}$$

For instance, for $\mathcal{C} = \{00, 01, 11\}$, we can see that $x_1(1 - x_2) \in I_\mathcal{C}$. What else is in $I_\mathcal{C}$? How does $V(I_\mathcal{C})$ relate to $\mathcal{C}$? For any $S \subset k^n$, $S \subseteq V(I(S))$, however, is is possible that $S \subsetneq V(I(S))$. In this case, however, $V(I_\mathcal{C}) = \mathcal{C}$. To show this, we introduce indicator polynomials:

Definition 1.20 (indicator polynomial)**.** *Let $v = (v_1, \ldots, v_n) \in \mathbb{F}_2^n$. Then the indicator polynomial of v, ρ_v, is defined to be a polynomial such that $\rho_v(v) = 1$ and $\rho_v(v') = 0$ for $v' \neq v$. We can construct ρ_v as*

$$\rho_v = \prod_{v_i=1} x_i \prod_{v_i=0} (1 - x_i)$$

The indicator polynomial ρ_c of a codeword $c = (c_i, \ldots, c_n)$ is the polynomial $\prod_{c_i=1} x_i \prod_{c_i=0}(1 - x_i)$. Note that $\rho_c(v) = 1$ if and only if $v = c$. Then for any neural code, the indicator polynomial of every codeword $c \notin \mathcal{C}$ is contained in $I_\mathcal{C}$, since p_c vanishes on every point but c, and $c \notin \mathcal{C}$, so p_c

vanishes at every point of C. This proves that $V(I_C) = C$, since the only set on which every p_c for $c \notin C$ vanishes is C. However, the indicator polynomials are not the only elements of I_C. Since $x^2 = x$ for any $x \in \mathbb{F}_2$, all polynomials of the form $x_i - x_i^2$, known as the boolean relations, are contained in I_C. Lemma 3.2 in Curto et al. (2013) shows that the indicator polynomials and the boolean relations are sufficient to generate I_C.

Instead of working directly with I_C, we reserve the term *neural ideal* for the ideal J_C, which does not contain the boolean relations. We construct the neural ideal J_C from this as follows:

$$J_C = \langle \rho_c | c \in \{0,1\}^n \setminus C \rangle.$$

Note that for each $c \notin C$, there is some polynomial $\rho \in J_c$ such that $\rho(c) = 1$. For each $c \in C$, however, $\rho(c) = 0$, since $\rho_v(c) = 0$ for all generators. This goes to say that C is exactly the variety of J_C. These definitions are taken from Curto et al. (2013).

Example 1. *Let* $C = \{000, 100, 101, 111, 110, 010\}$. *Then* $\mathbb{F}_2^3 \setminus C = \{001, 011\}$. *Then*

$$J_C = \langle (1 - x_1)(1 - x_2)x_3, (1 - x_1)x_2x_3 \rangle.$$

1.4.2 The Canonical Form

Curto et al. (2013) introduce a cannonical form $CF(J_C)$ which allows one to read off information about receptive field relationships from the neural ideal. They define pseudo-monomials to be polynomials of the form $f \in \mathbb{F}_2[x_1, \ldots, x_n]$,

$$f = \prod_{i \in \sigma} x_i \prod_{j \in \tau} (1 - x_j)$$

for $\sigma, \tau \subset [n]$, $\sigma \cap \tau = \emptyset$. Note that for any $v \in \mathbb{F}_2^n$, the indicator polynomial p_v is a pseudo-moonomial. A pseudo-monomial ideal is an ideal which can be generated by pseudo-monomials. Since $J_C = \langle \{p_v | v \notin C\} \rangle$, it is a pseudo-monomial ideal.

The canonical form of J_C is defined as a generating set consisting of minimal pseudo-monomials, where a pseudo-monomial $f \in J_C$ is minimal if and only if there does not exist another pseudomonomial $g \in J_C$ with $deg(g) < deg(f)$ such that $f = hg$ for some $h \in \mathbb{F}_2^n[x_1, \ldots, x_n]$. Curto et al. (2013) show that the generators in the cannonical form are of three kinds, each corresponding to a statement about the (minimal) relationships between receptive fields in any realization (convex or not) of a given code:

- Type 1: $\prod_{i\in\sigma} x_i \in J_C$ implies that $\bigcap_{i\in\sigma} U_i = \emptyset$, but for any lower order intersection $\alpha \subset \sigma$, $\bigcap_{i\in\alpha} U_i \neq \emptyset$.
- Type 2: $\prod_{i\in\sigma} x_i \prod_{j\in\tau}(1-x_j) \in J_C$ implies that $\bigcap_{i\in\sigma} U_i \subset \bigcup_{j\in\tau} U_j$.
- Type 3: $\prod_{j\in\tau}(1-x_j) \in J_C$ implies that the entire stimulus space X satisfies $X \subset \bigcup_{j\in\tau} U_j$, but that X is not contained in any lower order union $\bigcup_{j\in\alpha} U_j$ for $\alpha \subset \tau$.

By computing the cannonical form $CF(J_C)$, we thus arrive at a description of the receptive field structure of any realization of C.

Example 2. *Consider again the code* $C = \{000, 100, 101, 111, 110, 010\}$ *with* $J_C = \langle(1-x_1)(1-x_2)x_3, (1-x_1)x_2x_3\rangle$. *Note that* $x_3(1-x_1)$ *divides both* $(1-x_1)(1-x_2)x_3$ *and* $(1-x_1)x_2x_3$, *therefore* $\langle(1-x_1)(1-x_2)x_3, (1-x_1)x_2x_3\rangle \subset \langle x_3(1-x_1)\rangle$. *Thus, the minimal relationhip describing this code is* $U_3 \subset U_1$, *which comes from the type* 2 *relation* $x_3(1-x_1)$.

Curto et al. (2013) present an algorithm to find the canonical form of J_C using primary decompositions. Petersen et al. (2016) present and implement a faster algorithm. Garcia et al. (2016) explore the relationship between the canonical form and the Gröbner basis of the neural ideal, and finds that though $CF(J_C)$ is not always a Gröbner basis, when it is a Gröbner basis, it is always a reduced univeral Gröbner basis. Furthermore, when a pseduo-mononial is contained in the Gröbner basis of the neural ring, it is always an elment of the cannonical form.

1.5 Toric Ideals of Neural Codes

Another polynomial ideal associated to every neural code is the toric ideal, introduced in this context by Gross et al. (2016). Roughly, toric ideals of neural codes capture sets of codewords which balance out at each neuron. Toric ideals are well studied outside of the context of neural codes, thus we define toric ideals in the general case before exploring the particular case of the toric ideal of a neural code.

If A is an $n \times m$ matrix with integer entries, we can define a familiar linear map from $\mathbb{N}^m \to \mathbb{Z}^n$ by matrix multiplication. Specifically, if the column vectors of A are $\mathbf{a}_1, \ldots, \mathbf{a}_m$, we have the map

$$\phi_{A^*} : N^m \to Z^d, \mathbf{u} = (u_1, \ldots, u_n) \mapsto u_1\mathbf{a}_1 + \ldots + u_n\mathbf{a}_n.$$

We can define a ring homomorphism from $k[x_1, \ldots, x_n]$ to $k[y_1^{\pm 1}, \ldots, y_m^{\pm 1}]$ based on this linear map as follows (using the notational convention $\mathbf{x} = (x_1, \ldots, x_n)$, $\mathbf{a}_i = a_{i1}, \ldots, a_{i_n}$, $\mathbf{x}^{\mathbf{a_i}} = x_1^{a_{i1}} \cdots x_n^{a_{in}}$):

$$\phi_A : k[x_1, \ldots, x_n] \to k[y_1^{\pm 1}, \ldots, y_m^{\pm 1}], x_i \mapsto \mathbf{y}_{\mathbf{a}_i}.$$

The toric ideal is $I_A = \ker(\phi_A) \subset k[x_1, \ldots, x_n]$.

Example 3. *Let* $A = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 1 & 1 \end{bmatrix}$. *Then the map* ϕ_A *is*

$$\phi_A : k[x_1, x_2, x_3] \to [y_1, y_2]$$

is given by

$$\phi_A(x_1) = y_1 \qquad \phi_A(x_2) = y_1^2 y_2 \qquad \phi_A(x_3) = y_2.$$

Thus, we see that $x_1^2 x_3 - x_2 \in I_A$, *since* $\phi_A(x_1^2 x_3 - x^2) = y_1^2 y_2 - y_1^2 y_2 = 0$.

Toric ideals have a number of applications in statistics, optimization, and other fields, and the associated varieties, toric varieties, are important objects in algebraic geometry. Therefore, toric ideals are well studied, and have a number of convenient properties: basic results show that the toric ideal is a prime ideal generated by binomials. Further, several software packages such as 4ti2 (Hemmecke et al. (2003)) and Macaulay 2 (Grayson and Stillman (2002)) have good support for computing with toric ideals. Therefore, toric ideals of neural codes are promising objects of study, and any general discoveries about toric ideals made in the proccess are likely to be of broad interest.

The definition of a toric ideal of a neural code follows from the general definition of a toric ideal: to define the toric ideal of a neural code C, we can take the codewords to be the row vectors of a matrix and define I_C to be the toric ideal of this matrix. This is equivalent to the definition given in Gross et al. (2016), which is the following: Let C be a neural code on n neurons and $C^* = C \setminus (0, \ldots, 0)$. Define the *codeword ring* of C to be $W_C = k[\{y_c | c \in C^*\}]$ and the *neuron ring* to be $N_C = k[x_1, \ldots, x_n]$. Let a homomorphism $\phi_C : N_C \to W_C$ be given by

$$y_c \mapsto \prod_{i \in supp(c)} x_i.$$

Then the toric ideal T_C is the kernel of ϕ_C.

Example 4. *Let* $C = \{00, 10, 01, 11\}$. *Then*

$$W_C = k[y_{10}, y_{11}, y_{01}]$$

and

$$N_C = k[x_1, x_2].$$

We have $\phi_C(y_{10}) = x_1, \phi_C(y_{11}) = x_1x_2,, \phi_C(y_{01}) = x_2$. *Then* $\phi_C(y_{10}y_{01} - y_{11}) = x_1x_2 - x_1x_2 = 0$, *so* $y_{10}y_{01} - y_{11} \in \ker(\phi_C) = I_C$. *In fact,* $I_C = \langle y_{10}y_{01} - y_{11}\rangle$. *First,* $\langle y_{10}y_{01} - y_{11}\rangle \subset I_C$. *Now, suppose that* $\langle y_{10}y_{01} - y_{11}\rangle \subsetneq I_C$. *Since* I_C *is generated by binomials, this implies that there exists some* $m_1 - m_2 \in I_C \setminus \langle y_{10}y_{01} - y_{11}\rangle$. *Then* $\phi_C(m_1) = \phi_C(m_2)$. *Unless* y_{11} *divides either* m_1 *or* m_2, $m_1 = m_2$, *thus* $m_1 - m_2 = 0 \in \langle\langle y_{10}y_{01} - y_{11}\rangle\rangle$. *If* y_{11} *divides both* m_1 *and* m_2, *we can cancel it out. Therefore, suppose* y_{11} *divides* m_1. *Likewise, if* y_{10} *or* y_{01} *divides* m_1, *we can cancel out so that it does not divide* m_2. *In order for* $\phi_C(m_1) = \phi_C(m_2)$, *for each power of* y_{11} *in* m_1, *a power of* $y_{10}y_{01}$ *must divide* m_2. *Thus* $m_1 = y_{11}^m$ *and* $m_2 = y_{10}^m y_{01}^m$.

Now, we prove using strong induction that for all $m \in \mathbb{N}$ $y_{11}^m - y_{10}^m y_{01}^m \in \langle y_{10}y_{01} - y_{11}\rangle$. *The base case, for* $m = 1$, *is clear. Now, assume that for all* $k \in 1, \ldots, m$, *the statement holds. Now, take*

$$\begin{aligned} y_{11}(y_{11}^{m-1} - y_{10}^{m-1}y_{01}^{m-1}) + y_{01}y_{10}(y_{11}^{m-1} - y_{10}^{m-1}y_{01}^{m-1}) \\ -y_{11}y_{10}y_{01}(y_{11}^{m-2} - y_{10}^{m-2}y_{01}^{m-2}) = \\ y_{11}^m - y_{11}y_{10}^{m-1}y_{01}^{m-1} + y_{01}y_{10}y_{11}^{m-1} - y_{10}^m y_{01}^m - \\ y_{10}y_{01}y_{11}^{m-1} + y_{11}y_{10}^{m-1}y_{01}^{m-1} = y_{11}^m - y_{10}^m y_{01}^m \end{aligned}$$

Thus, since we have assumed $y_{11}^{m-1} - y_{10}^{m-1}y_{01}^{m-1}, y_{11}^{m-2} - y_{10}^{m-2}y_{01}^{m-2} \in \langle y_{11} - y_{10}y_{01}\rangle$ *and we have expressed* $y_{11}^m - y_{10}^m y_{01}^m$ *as a linear combination of these elements,* $y_{11}^m - y_{10}^m y_{01}^m \in \langle y_{11} - y_{10}y_{01}\rangle$. *Thus, we have shown that any potential generator for* I_C *is divisible by an element of the form* $y_{11}^m - y_{10}^m y_{01}^m$ *and that any element of this form is contained in* $\langle y_{10}y_{01} - y_{11}\rangle$, *we have shown that*

$$I_C = \langle y_{10}y_{01} - y_{11}\rangle.$$

Gröbner bases of toric ideals are well studied. A particular universal Gröbner basis, the Graver basis, consists of the set of all primitive binomials. A binomial $\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}}$ is primitive in I_C if $\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}} \in I_C$ and there is no other $\mathbf{x}^{\mathbf{a}'} - \mathbf{x}^{\mathbf{b}'} \in I_C$ such that $\mathbf{x}^{\mathbf{a}'}$ divides $\mathbf{x}^{\mathbf{a}}$ and $\mathbf{x}^{\mathbf{b}'}$ divides $\mathbf{x}^{\mathbf{b}}$. Note that the Graver basis is not neccesarily reduced, that is, it may contain binomials whose leading terms are not neccesary to generate the initial ideal under any term order.

Example 5. *Revisit the code* $C = \{00, 10, 11, 01\}$*.The proof in example* 4 *can be simplified dramatically by this fact. If we suppose* $m_1 - m_2 \in I_C$ *is a primitive binomial, then our argument that* $m_1 = y_{11}^m$ *and* $m_2 = y_{10}^m y_{01}^m$ *completes the proof, since we must have* $m = 1$ *for* $m_1 - m_2$ *to be primitive.*

Chapter 2

Inductively Pierced Neural Codes and their Toric Ideals

Toric ideals are well suited to studying a class of neural codes known as inductively pierced codes.

2.1 Inductively Pierced Codes

Here, we define and describe inductively pierced codes. First, we give the definitions from Gross et al. (2016). Next, we give an equivalent set of definitions and prove the equivalence. The authors of this paper define inductively pierced codes in terms of well formed Euler diagrams.

Definition 2.1 (Euler diagram, zone)**.** *An* Euler diagram *D for n sets is a collection of n simple closed curves in $\mathbb{R}^2$ labeled $\lambda_1, \ldots, \lambda_n$. The interior of the curve λ_i is the open set U_i. Nonempty intersections of the sets $U_1, \ldots, U_n$ and their complements $\bar{U}_1, \ldots, \bar{U}_n$ are known as* zones*. The word zone is also used to refer to a set $Z \subset \{\lambda_1, \ldots, \lambda_n\}$ such that*

$$\bigcap_{i \in Z} U_i \cap \bigcap_{j \notin Z} \overline{U}_j \neq \emptyset.$$

An Euler diagram is said to be well formed if each of the following are satisfied:

1. *Each curve label is used only once.*

2. *All curves intersect generally* (*that is, curves intersect only in finitely many points*)*.*

3. *A point in the plane is passed through at most two times by the curves in the diagram*

4. *Each zone is connected*

In figures 2.1 *and* 2.2*, we give examples and nonexamples of well formed Euler diagrams.*

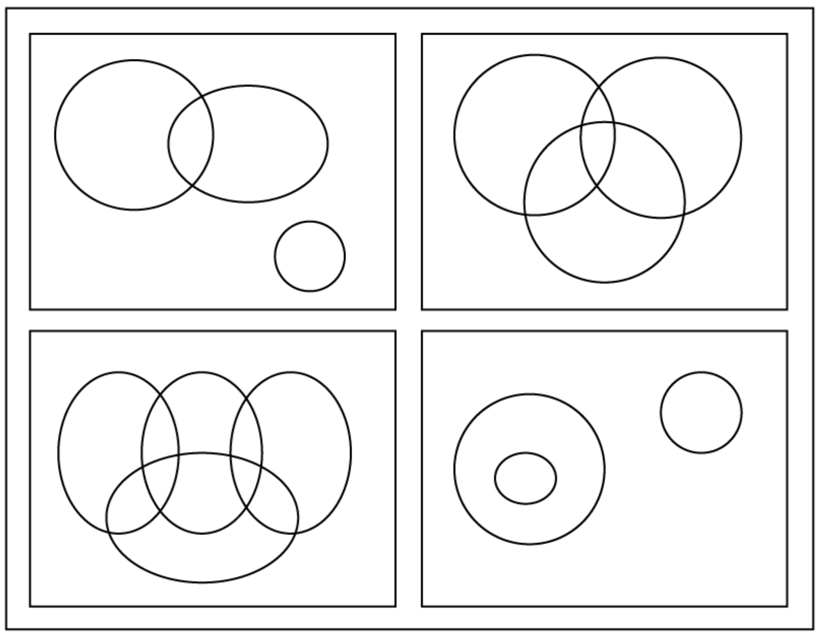

Figure 2.1 Here are some examples of well formed Euler diagrams.

We then define the abstract description of a Euler diagram as follows:

Definition 2.2 (abstract description)**.** *An* abstract description $\mathcal{D} = (\mathcal{L}, \mathcal{Z})$ *of an Euler diagram D is an ordered pair specifying the curve labels $\mathcal{L}$ and the zones $\mathcal{Z} \subset \mathcal{P}(\mathcal{L})$. If C is a neural code on n neurons, the abstract description which corresponds to the code, $\mathcal{D}_C$ is defined as $\mathcal{D}_C = \{[n], \mathcal{Z}_C\}$ where $\mathcal{Z}_C = \{\mathrm{supp}(c), c \in C\}$.*

See Figure 2.3 for an example of a Euler diagram and its abstract description. Next, we define some structures that will be useful in defining a k-piercing.

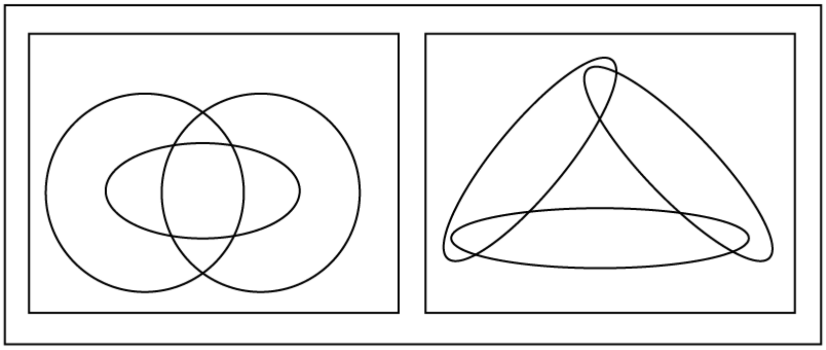

Figure 2.2 Here are some examples of Euler diagrams which are not well formed. The code on the left has a disconnected zone, though the place field in the center can be moved either up or down in order to change this disconnected zone into a triple intersection. The diagram on the right looks well formed at first glance, however, the zone corresponding to the empty set is disconnected.

Definition 2.3 (abstract place field, Λ-cluster)**.** *The* abstract place field *of a curve label in an abstract description $\mathcal{D} = \{(\lambda_1, \ldots, \lambda_n), \mathcal{Z}\}$ is defined as*

$$\chi_{\lambda_i} = \{Z \in \mathcal{Z} : \lambda_i \in Z\}.$$

See Figure 2.3 *for an example of an abstract place field. The Λ-cluster of a zone $Z \in \mathcal{Z}$ for some $\Lambda \subset \{\lambda_1, \ldots, \lambda_n\}$ is defined as*

$$\mathcal{Y}_{Z,\Lambda} = \{Z \cup \Lambda_i : \Lambda_i \subset \Lambda\}.$$

Note that $\mathcal{Y}_{Z,\Lambda}$ is not neccesarily a subset of $\mathcal{Z}$.

Finally, we define the $k-$piercing of an abstract description:

Definition 2.4 (k-piercing)**.** *Let $\mathcal{D} = (\mathcal{L}, \mathcal{Z})$ be an abstract description. Let $\Lambda = (\lambda_1, \ldots, \lambda_k) \subset \mathcal{L}$. Then λ_{k+1} is a k-piercing of Λ in $\mathcal{D}$ if there exists a zone $Z \in \mathcal{Z}$ such that*:

1. $\lambda_i \notin Z$ *for all* $i \leq k + 1$.
2. $\chi_{\lambda_{k+1}} = \mathcal{Y}_{Z \cup \lambda_{k+1}, \Lambda}$*, and*
3. $\mathcal{Y}_{Z,\Lambda} \subset Z$.

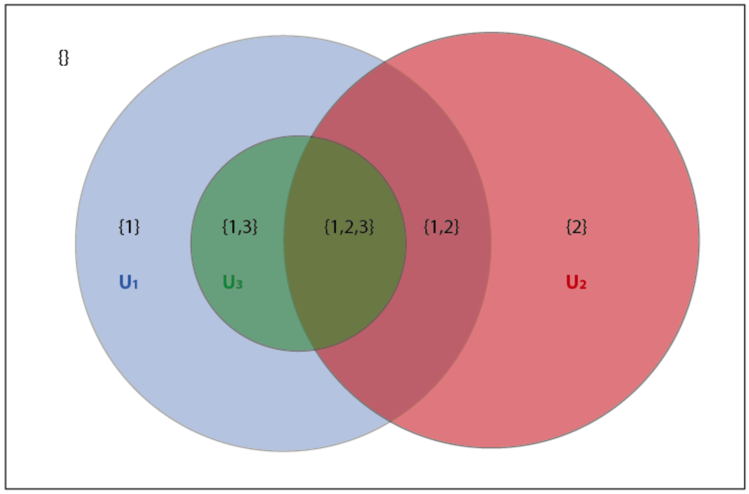

Figure 2.3 The zones in this Euler diagram are the regions $\{\{1\},\{1,3\},\{1,2,3\},\{1,2\},\{2\}\}$. We see that $\chi_1 = \{\{1\},\{1,3\},\{1,2,3\},\{1,2\}\}$ in the abstract description of this Euler diagram. In the Euler diagram, this corresponds to the regions with a blueish tint.

Next, we define what it means for an abstract description to be inductively pierced. In order to do this, we first describe the removal of a curve.

Definition 2.5 (Removal of a Curve)**.** *Let $\mathcal{D} = (\mathcal{L}, \mathcal{Z})$ be an abstract description with $\lambda \in \mathcal{L}$. We define*

$$\mathcal{D} - \lambda = \{\mathcal{L} \setminus \{\lambda\}, \mathcal{Z} - \lambda\}$$

where

$$\mathcal{Z} - \lambda = \{Z \setminus \{\lambda\} : Z \in \mathcal{Z}\}$$

Definition 2.6. [*k-inductively pierced*] *An abstract description $\mathcal{D} = (\mathcal{L}, \mathcal{Z})$ is k-inductively pierced if $\mathcal{D}$ has a l-piercing λ for $0 \leq l \leq k$ such that $\mathcal{D} - \lambda$ is k-inductively pierced. A neural code C is k-inductively pierced if its corresponding abstract description, $\mathcal{D}_C$ is k-inductively pierced.*

There are very nice geometric interpretations of 0- and 1- piercings. A curve is a 0-piercing if and only if it intersects no other curves in the diagram. Then the interior of a 0-piercing must be completely contained

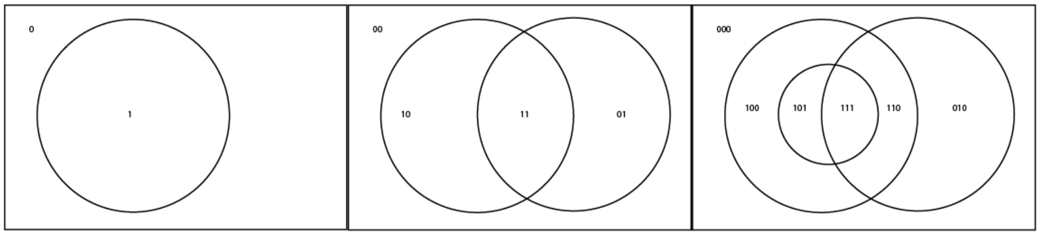

Figure 2.4 The code $\{000, 100, 010, 110, 101, 111\}$ is an inductively pierced code. We see that the code $\{0, 1\}$ is inductively pierced by definition, the code $\{00, 10, 01, 11\}$ is a 1-piercing of $\{0, 1\}$, and the code $\{000, 100, 010, 110, 101, 111\}$ is a 1-piercing of $\{00, 10, 01, 11\}$.

within another zone. A curve is a 1-piercing if and only if it intersects one other curve in the diagram and its interior is split into exactly two zones. Because of these characterizations, inductively 0- pierced codes are convex with minimal embedding dimension 1 and inductively 1-pierced codes are convex with minimal embedding dimension 1 or 2. See Figure 2.4 for a geometric realization of the inductively 1-pierced neural code $\{000, 100, 010, 110, 101, 111\}$.

In this work, rather than directly using these definitions, we work only with neural codes (with no reference to abstract descriptions or Euler diagrams) and rephrase a k-piercing as an operation, rather than a property of a curve label: that is, we can apply a k-piercing to a neural code C on n neurons to obtain C' on $n+1$ neurons. We prove that these sets of definitions provide an equivalent notion of a k-inductively pierced neural code.

To define the k-piercing operation, it will be useful to define the restriction of a neural code on neurons $x_1, \ldots, x_n$ to some subset of the neurons $x_{i_1}, \ldots, x_{i_k}$. The intuitive idea behind a restriction is that we will ignore the behavior all neurons other than $x_{i_1}, \ldots, x_{i_k}$, turning a neural code on n neurons into a neural code on k neurons. We express this in terms of matrices:

Definition 2.7 (restriction)**.** *Let C be a neural code on n neurons $x_1, \ldots, x_n$. The* restriction *of C to $\{x_{i_1}, \ldots, x_{i_k}\} \subset \{x_1, \ldots, x_n\}$, denoted $C|_{x_{i_1},\ldots,x_{i_n}}$ is the image of C under the linear transformation $r_{x_{i_1},\ldots,x_{i_k}}$, called the* restriction map *given by* *the matrix*

$$\begin{bmatrix} e_{i_1} \\ e_{i_2} \\ \vdots \\ e_{i_k} \end{bmatrix},$$

where e_{i_j} is the vector of length n which has a 1 *at index i_j and* 0*'s elsewhere.*

For instance, the restriction of the neural code $C = \{000, 101, 110\}$ to x_2 and x_3 is the image of C under the linear transformation r_{x_2,x_3} given by the matrix

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Then $C|_{x_2,x_3} = \{00, 01, 10\}$.

Note that the restriction of a neural code to all but one neuron corresponds to the removal of a curve as defined in Definition 2.5 in the abstract description corresponding to the neural code, since we remove the deleted neuron from the set of neurons and by removing index corresponding to the deleted neuron from each codeword, we remove the deleted neuron from the support of each codeword. With this, we are ready to define a k-piercing operation.

Definition 2.8 (k-piercing operation)**.** *Let C be a neural code on n neurons. Assume there exists a codeword $d \in C$ and a subset of neurons $\lambda = \{x_{i_1}, \ldots, x_{i_k}\}$ such that*

1. $\lambda \cap \operatorname{supp}(d) = 0$.
2. $D_\lambda = \mathbb{F}_2^k$ *for D defined as the set of codewords c such that $r_{[n]\setminus\lambda}(c) = d$.*

The k-piercing of C at λ with respect to background codeword $c \in C$ is obtained as follows:

3. *Extend each $c \in C$ with a* 0 *at the $(n+1)^{th}$ index.* (*This is equivalent to adding a neuron that never fires.*)
4. *For each $c \in C$ such that* $\operatorname{supp}(c) \cap \lambda \neq \emptyset$ *and $r_{[n]\setminus\lambda}(c) = d$, add a codeword c' which extends c with a* 1 *at the $(n+1)^{th}$ index.*

Definition 2.9. [*inductively k-pierced*] *A neural code C is inductively pierced if either*

- $C = \{0, 1\}$
- *There exists a permutation of indices, an inductively k-pierced code C' on n neurons, and $0 \leq m \leq k$ such that C is a m-piercing of C'.*

In other words, a code is inductively pierced if it can be built up from the one neuron code $C = \{0, 1\}$ through a series of piercings. Inductively k-pierced codes were introduced because their geometric realizations are particularly nice, especially when k is 0, 1, or 2.

Now, we prove that our definition of k-inductively pierced is equivalent to that in Gross et al. (2016).

Theorem 1. *A neural code satisfies definition* 2.6 *if and only if it satisfies definition* 2.9.

Proof. To prove this, we first prove the following lemma:

Lemma 1. *A curve n is a k-piercing in abstract description $\mathcal{D}_C$ of neural code C if and only if C can be obtained from $C|_{[n-1]}$ by a k-piercing operation.*

Proof. First, suppose curve $n = \lambda_{k+1}$ is a k-piercing of $\Lambda = \{\lambda_1, \ldots, \lambda_k\} \subset [n]$ with respect to background zone Z. Now, let $\lambda = \{x_1, \ldots, x_n\} \subset [n]$ and d be a codeword such that $\text{supp}(d) = Z$. Then condition (1) of the definition of a k-piercing operation is satisfied, since $\lambda_i \notin Z$ for each $i \leq k+1$ guarantees $\text{supp}(d) \cap \lambda = \emptyset$. Condition (2) of the definition of a k-piercing operator is guaranteed by condition (3) of the definition of a k-piercing in a curve. Next, note that the set of codewords in C with a 0 at the n^{th} index is the set of codewords of $C|_{[n-1]}$, each extended with a zero because conditions (2) and (3) of the definition of a k-piercing,

$$\chi_{\lambda_{k+1}} = \mathcal{Y}_{Z \cup \lambda_{k+1}, \Lambda},$$

and

$$\mathcal{Y}_{Z,\Lambda} \subset Z.$$

in an abstract description together that all codewords containing neuron n also come in a version which does not contain neuron n. Thus condition (3) of the definition of a k-piercing operation is satisfied. Finally, condition (2) of the definition of a k-piercing in a curve,

$$\chi_{\lambda_{k+1}} = \mathcal{Y}_{Z \cup \lambda_{k+1}, \Lambda},$$

guarantees that the set of codewords in C which contain a 1 at position n are exactly the codewords c of $C|_{[n-1]}$ such that $\text{supp}(c) \cap \lambda \neq 0$ and $r_{[n]\setminus\lambda(c)} = d$.

Thus, if curve $n = \lambda_{k+1}$ is a k-piercing of $\Lambda = \{\lambda_1, \ldots, \lambda_k\} \subset [n]$ with respect to background zone Z in the abstract description $\mathcal{D}_C$, then the neural code C can be obtained from the neural code $C|_{[n-1]}$ through a k-piercing operation.

Now, suppose C can be been obtained from $C|_{[n-1]}$ through a k-piercing operation at $\lambda = \{x_{i_1}, \ldots, x_{i_k}\}$ with respect to background codeword d. Then let $\Lambda = \{\lambda_1, \ldots, \lambda_k\}$ and $Z = \text{supp}(d)$. We prove that $n = \lambda_{k+1}$ is a k-piercing of Λ with respect to background codeword Z in Euler diagram $\mathcal{D}_C$. As before, condition (1) of the definition of a k-piercing in a curve is guaranteed by condition (1) of the k-piercing operation. Condition (2) of the definition of a k-piercing in a curve is guaranteed by conditions (2) and (4) of a k-piercing operation–condition (2) guarantees that all codewords whose zones composed of Z unioned with some subset of Λ are present in $\mathcal{D}_{C_{[n-1]}}$ and condition 4 adds a 1 to these codewords and only these codewords . Condition (3) of the definition of a k-piercing in a curve is guaranteed by conditions (2) and (3) of a k-piercing operation, condition (2) again guarantees that all codewords whose zones composed of Z unioned with some subset of Λ are present in $\mathcal{D}_{C_{[n-1]}}$ and condition (3) preserves all of these codewords in $\mathcal{D}_C$ by adding a zero at the n^{th} index. $\square$

Now, we prove the Theorem 1 using induction. As a base case, consider neural codes on one neuron. While no base case is explicitly mentioned in the definition of inductively pierced given in Gross et al. (2016), if we assume that they empty code is inductively pierced, then the codeword $\{0, 1\}$ is 1 1−piercing of the empty neural code, thus $\{0, 1\}$ is inductively pierced by Definition 2.6 (and is the only inductively pierced code on one neuron). As our inductive hypothesis, we assume all $n - 1$ neuron codes wich are inductively pierced by Definition 2.6 are inductively pierced by Definition 2.9. Suppose C is a neural code on $n > 1$ neurons that satisfes Definition 2.6. Then $C - n$ is an inductively k-pierced code on $n - 1$ neurons and n is a k-piercing in C, so, according to the inductive hypothesis, it is also inductively pierced by Definition 2.9, and C is a m−piercing of $C - n$ for some $m \leq k$. Then by the lemma above, C can be obtained from $C - n = C|_{[n-1]}$ by a k-piercing operation. Then C is inductively pierced by Definition 2.9 as well. Likewise, suppose C is a neural code on $n > 1$ neurons such that C is inductively pierced according to definition Definition 2.9. Then it can be obtained from $C|_{[n]} = C - n$ by a k-piercing operation, by the inductive hypothesis, we assume $C - n$ is inductively pierced by definition Definition 2.6 as well. By the lemma above, n is a k-piercing in C, thus C is inductively pierced by Definition 2.6 as well. Therefore, since a neural code is inductively pierced by Definition 2.6 if and only if it is inductively pierced by Definition 2.9, these definitions are equivalent.

□

2.2 An Algorithm to Identify Inductively k-Pierced Neural Codes

This characterization of inductively k-pierced codes suggests a recursive algorithm to identify inductively k-pierced codes and exhibit an ordering of the neurons in which piercings can be peformed. At each step, we look for neurons which may have been added at the most recent piercing, remove them and all codewords they appear in from the code, and recurse. We describe this algorithm in general and have implemented it for 0- and 1-pierced codes. We also describe connections to the algorithm for identifying 0- and 1- pierced codes using the cannonical form of the neural ring and an associate graph.

To identify zero pierced codes, observe that a neuron appears in exactly one codeword of a well formed neural code if and only if it has been added as a zero piercing and has not been pierced since it was added. Therefore, any neuron which appears in only one codeword could have been added as a zero piercing at the last step. Thus, we can remove all these neurons by deleting all codewords which contain them in their support and by deleting the indices of these neurons. This gives us a neural code on fewer neurons which we can apply the same proccess to until we have either removed all neurons or found that each neuron is found in more than one codeword. In the first case, we have shown that the code is inductively pierced and found a piercing order.

We can extend this to 1- piercings as follows: once we have removed all neurons which appear in only one codeword, we turn to looking at neurons which appear in two codewords. At this point, note that a neuron shows up in exactly two codewords (which may differ from each other at exactly one index) if and only if it has been added as a 1-piercing and has not been pierced since it was added. Therefore, we can remove each neuron which appears in exactly two codewords which differ at only one index, and then return to finding zero piercings.

If we wish to identify k-pierced codes, we continue this proccess, removing neurons if they appear in 2^k codewords which are arranged in the right way and declaring that the code is not inductively pierced otherwise. In order to find the piercing order, we keep track of the neurons we remove at each step: as we remove a neuron, we add it to the front of a list of the neurons. Thus, the list becomes sorted in an order which makes the code inductively pierced.

2.3 Toric Ideals of Inductively Pierced Codes

Can we use toric ideals to identify inductively k-pierced codes? Gross et al. (2016) prove that a well formed neural code C is zero-pierced if and only if $I_C = \langle 0 \rangle$ and that if a neural code is 1-inductively pierced, then the toric ideal I_C is generated by quadratics or $I_C = \langle 0 \rangle$. However, the converse is not true–there is a counterexample on as few as three neurons: the well formed neural code $C = \{000, 100, 010, 001, 110, 101, 011, 111\}$, realized in Figure 2.5, is not inductively 1-pierced (though it is inductively 2-pierced), but has a quadratic generating set. They make the following conjecture:

Figure 2.5 The neural code $C = \{000, 100, 010, 001, 110, 101, 011, 111\}$ is not inductively 1-pierced, but is generated by $\langle y_{110} - y_{100}y_{010}, y_{101} - y_{100}y_{001}, y_{011}y_{010}y_{001}, y_{111} - y_{110}y_{001}\rangle$, which is quadratic.

Conjecture 1. *For each n, there exists a term order such that each well formed neural code is* 0- *or* 1- *inductively pierced if and only if the reduced Gröbner basis contains only binomials of degree* 2 *or less.*

We prove the "only if" direction of this conjecture. An outline of our proof is as follows: first, we define inductively built codes, a broader class of codes including all inductively k−pierced codes. We show that the toric ideals of these codes have a nice inductive decomposition, and that there is a natural way to use this decomposition to describe the Graver basis of the code. From here, we show that if we can find a term order satisfying certain properties, the intial ideal generated by the set of quadratic binomials will match the initial ideal generated by the Graver basis, which is sufficient to establish that the quadratic binomials form a Gröbner basis. Finally, we show that a certain lexicographic order satisfies our properties.

Our definition of inductively built codes generalizes definition 2.9 of an inductively k-pierced code by relaxing the requirement that the set of old codewords to which the new neuron is added is no longer required to be the full set of 2^k codewords on k neurons, but can be any set of potentially adjacent codewords. Instead, all we require is that the new neuron be added such that its place field does not cover any existing zone.

To be more concrete, we give the following definition:

Definition 2.10 (inductively built neural code)**.** *A neural code is inductively build if it can be built up from the empty code using the following operation*:

Let C be a neural code on n neurons. Let $C' = C$. Now, adjoin a 0 to each codeword of C'–this is equivalent to adding a new neuron which never fires. Finally, choose some subset of the codewords in C, adjoin a 1, and add them to C'.

Equivalently, we are requiring every codeword of C' to be a codeword of C with a 0 or 1 adjoined and every codeword of C to be found in C' with a 0 adjoined.

Now, we describe the toric ideals of inductively built codes with the following theorem:

Theorem 2. *If C' can be built from C in one step, then there exists an ideal $J \subset N_{C'}$ such that*

$$I_C \cong J \subset I'_C.$$

Proof. Let C' be built from C using the operation in definition 2.10. Let $c' \in C'$ be the codeword which is a zero adjoined to $c \in C$. Define the maps

$$w : W_C \to W_{C'}, y_c \mapsto y_{c'}$$

and

$$n : N_C \to N_{C'}, x_i \mapsto x_i.$$

Our aim is to show $I_C \cong w(I_C) \subset I_C$. In order to so this, we wish to show that these maps commute with ϕ_C and $\phi_{C'}$, that is, that

$$n \circ \phi_C = \phi_{C'} \circ w.$$

Recall that $\phi_C, \phi_{C'}$ are defined by

$$\phi_C(y_c) = \prod_{c_i=1} x_i$$

and

$$\phi_{C'}(y'_c) = \prod_{c'_i=1} x_i.$$

Thus,

$$n \circ \phi_C(y_c) = n(\prod_{c_i=1} x_i) = \prod_{c_i=1} x_i$$

and

$$\phi_{C'} \circ w(y_{c'}) = \phi_{C'} \circ p_{c'} = \prod_{c'_i=1} x_i = \prod_{c_i=1} x_i$$

because $c'_i = 1$ if and only if $c_i = 1$. Therefore, the maps commute (on generators). Therefore, they commute overall. Then if $q \in I_C$, $w(q) \in I_C$. Therefore, $w(I_C) \subset I_{C'}$.

Next, note that the map w is injective, since $c' = d'$ if and only if $c = d$. Therefore, $I_C \cong w(I_C)$. Thus

$$I_C \cong w(I_C) \subset I_{C'}.$$

□

Corollary 1. *If C' is a k-piercing of C, there exists J such that*

$$I_C \cong J \subset I'_C.$$

Proof. This follows from the previous theorem because any k-piercing operation is an inductive building operation as well. □

Lemma 2. *Let C be an inductively 1-pierced neural code on n neurons such that the final neuron is added in a 1-piercing. Let the two codewords added at the last step be denoted by n and n', where n' is the codeword with the larger support. Let I_C be the toric ideal of C. If there exists a term order $\prec$ such that*

$$f y_n \prec g y_{n'}$$

for all f, g such that $f y_n - g y_{n'}$ is a primitive binomial in I_C and such that the Gröbner basis for $I_{C|_{1,\dots,n-1}}$ with respect to the restriction of $\prec$ to $I_{C|_{1,\dots,n-1}}$ is quadratic. Then the Gröbner basis for I_C with respect to $\prec$ is quadratic.

Proof. To prove that there exists a quadratic Gröbner basis, it is sufficient to prove that the set of all quadratics is a Gröbner basis. Let $\mathcal{B}$ be the set of quadratics in I_C. Then we wish to show that $in_<(I_C)$ is generated by $\{in_<(b) \mid b \in \mathcal{B}\}$. Since we are guaranteed that the set of primitive binomials, the Graver basis, is a universal Gröbner basis for a toric ideal, it is sufficient to show that the leading term of each primitive binomial is contained in $\langle\{in_<(b \mid b \in \mathcal{B}\}\rangle$.

Since we have assumed that the Gröbner basis for $I_{C|_{1,\dots,n-1}}$ is quadratic, if an element of the toric ideal does not include either y_n or $y_{n'}$, then its leading term is contained in $\langle\{in_<(b \mid b \in \mathcal{B}\}\rangle$. Therefore, we only need to show that leading terms of primitive binomials of the form $fy_n - gy_{n'}$ are contained in $\langle\{in_<(b \mid b \in \mathcal{B}\}\rangle$.

Since the last neuron is added in a 1- piercing, the difference in support between y_n and $y_{n'}$ is the index of the pierced neuron. Equivalently,

$$\frac{\phi(f)}{\phi(g)} = \frac{\phi(y_{n'})}{\phi(y_n)} = x_i.$$

In the first case, assume that the neuron n is the first piercing of the neuron i. Then the only codewords whose support contains i are y_i, y_i', and y_n'. (If i was addded as a 0-piercing rather than a 1-piercing, we omit $y_{i'}$.) We can assume that the binomial $fy_n - gy_{n'}$ is primitive, therefore, $y_{n'}$ does not divide f.

Therefore either y_i or $y_{i'}$ divides f. Let $i^* = i' - \mathbf{e}_i$ and $i'^* = i' - \mathbf{e}_i$. Then both

$$\begin{aligned}
y_i y_n - y_{i^*} y_{n'} &\in \mathcal{B} \\
y_i y_n &\in \{in_<(b) \mid b \in \mathcal{B}\} \\
y_{i'} y_n - y_{i'^*} y_{n'} &\in \mathcal{B} \\
y_{i'} y_n &\in \{in_<(b) \mid b \in \mathcal{B}\}.
\end{aligned}$$

Then since either $y_i y_n$ or $y_{i'} y_n$ divides fy_n, thus $fy_n \in \langle\{in_<(b) \mid b \in \mathcal{B}\}\rangle$.

In the second case, assume that the neuron i has been pierced before. Then the set of codewords with i in their support contains all codewords which arise through 0 and 1 piercings of i. We can ignore 0 piercings or piercings of 0 piercings which stay within the place field of neuron i, since we can order the piercings such that all piercings which use place field i as part of the background zone occur after all 1-piercings of place field i. Thus, we consider 1-piercings of place field i. If neuron j is added as a 1-piercing of i, then the codewords j and j' can be defined as the codewords whose supports match the supports of the two codewords which were added when neuron j was added, and such that

$$\frac{\phi(y_{j'})}{\phi(y_j)} = x_i.$$

Then

$$\begin{aligned} y_{j'}y_n - y_j y_{n'} \in I_C. \\ y_{j'}y_n \in \{in_<(b) \mid b \in \mathcal{B}\} \end{aligned}$$

Therefore, as above, if $fy_n - gy_{n'}$, either $y_i y_n$, $y_{i'} y_n$, or some $y_{j'} y_n$ such that j is a piercing of i divides f. Therefore, $fy_n \in \langle \{in_<(b) \mid b \in \mathcal{B}\}\rangle$. □

Theorem 3. *Let C be an inductively* 1- *pierced neural code. Then there exists a term order $\prec$ such that the Gröbner basis for I_C with respect to $\prec$ is quadratic.*

Proof. We can satisfy the conditions of the previous lemma with a lexicographic term order. Assume the neurons are labeled such that the m^{th} neuron is added in a 0- or 1- piercing at the m^{th} step. Note that $\max(\text{supp}(c))$ gives the highest index in the support of a codeword; our choice of guarantees that this is also the step of inductive piercing where that codeword was added. Now, order the variables $\{y_c | c \in C\}$ such that $y_c \prec y_d$ if and only if either $\max(\text{supp}(c)) > \max(\text{supp}(d))$ or $\max(\text{supp}(c)) = \max(\text{supp}(d))$ and $|\text{supp}(c)| \leq |\text{supp}(d)|$. Let $\prec$ be the lexicographic monomial order induced by this ordering of the variables.

We use induction to prove that the Gröbner basis $\mathcal{G}_\prec$ for I_C is quadratic if C is an inductively pierced code on n neurons. As a base case, we consider the only inductively 1-pierced neural code on one neuron, $C = \{0, 1\}$. Since I_C is empty in this case, the Gröbner basis with respect to any term order contains only quadratic binomials. Now, suppose the theorem holds for all inductively pierced neural codes on m neurons. We wish to show it holds for all neural codes on $m+1$ neurons. Let C be an inductively pierced neural code on $m+1$ neurons. Then by definition C is either a 0- or 1- piercing of a neural code C_m on m neurons. If C is a 0-piercing of C_m, then $I_C \cong I_{C_m}$. Specifically, $I_C = w(I_{C_m})$. Then $f \prec g$ in I_C if and only if $w(f) \prec w(g)$ in I_{C_m}, since the order of the variables in I_C induces the correct order on the variables in I_{C_m}.

Otherwise, assume that C is a 1-piercing of C_m. As in the proof of the previous lemma, assume that we perform all 1-piercings of a neuron before performing 0-piercings of that neuron or 1- piercings whose place fields are entirely contained within this neuron. By our inductive hypothesis and the fact that the order of the variables in I_C induces the correct order on the variables in I_{C_m}, the Gröbner basis for I_{C_m} quadratic. Now, note that if $f y_n - g y_{n'} \in I_C$ and is primitive, then $y_{n'}$ does not divide f and y_n does not divide g. Observe that y_n is the variable with smallest support in the codeword ring which contains the most recently added neuron. Thus $f y_n \prec g y_{n'}$, since our lexicographic order guarantees that any monomial divisible by y_n precedes any other monomial.

Therefore, this term order satisfies all conditions of the preceding lemma. Now, note that using this term order on $k[\mathbb{F}_2^n]$ induces the correct term order for W_C for all neural codes C on n neurons. Therefore, the "only if" direction of is true: for each n, there exists a term order $\prec$ such that each well formed neural code is inductively pierced only if the reduced Gröbner basis for this term order contains only binomials of degree 2 or less.

□

Note that since this proof does not reference the fact that there exists a quadratic generating set for inductively 1-pierced neural codes, it provides an indepentent proof of this fact, first prove in Gross et al. (2016) using edge coloring of hypergraphs.

However, this term order is not sufficient to prove the converse of this theorem, as the following counterexample demonstrates:

Example 6. *Let* $C = \{000, 001, 011, 101, 111, 010, 110, 100\}$. *Abbreviating with* $z_0 = y_{001}$, $z_1 = y_{011}, z_2 = y_{101}, z_3 = y_{111}, z_4 = y_{010}, z_5 = y_{110}$, *and* $z_6 = y_{100}$ *and using the lexicographic order, the reduced Gröbner basis is*

$$\begin{aligned}\langle z_1z_2 - z_0z_3, z_0z_4 - z_1, z_2z_4 - z_3, z_3z_4 - z_1z_5, z_0z_5 - z_3,\\ z_2z_5 - z_3z_6, z_0z_6 - z_2, z_1z_6 - z_3, z_4z_6 - z_5\rangle,\end{aligned}$$

which is quadratic. However, C *is a well formed neural code that is not inductively* 1*-pierced.*

In investigating converse of this theorem, we phrase it in terms of Gröbner fans. We define the k-fan of an ideal, $GF_k(I)$, to be the subset of the Gröbner fan $GF(I)$ consisting of equivalence classes of weight vectors which define a Gröbner basis consisting of polynomials of degree k or less. Let W_n be the set of all well formed neural codes on n neurons and $P_{(1,n)}$ be set of inductively 1-pierced neural codes on n neurons. To phrase the theorem in the language of Gröbner fans, we note the following:

Theorem 4. *For each* n, *there exists a term order such that a well formed neural code is inductively* 1*-pierced if and only if the reduced Gröbner basis contains binomials of degree* 2 *or less if and only if the following holds*:

For each n,

$$\bigcap_{C\in P_n} GF_2(I_C) \setminus \bigcup_{C\in W_n\setminus P_n} GF_2(I_C) \neq \emptyset.$$

Proof. Suppose for each n, there is some term order for which the conjecture holds. This term order is given by some weight vector. Since this term order gives a quadratic Gröbner basis for every toric ideal of an inductively 1-pierced neural code, $\bigcap_{C\in P_n} GF_2(I_C)$ is nonempty.

However, since this term order does not yield quadratic Gröbner bases for well formed neural codes that are not inductively 1-pierced, this distinguishing weight vector lies outside $\bigcup_{C \in W_n \setminus P_n} GF_2(I_C)$. Thus, if the conjecture holds,

$$\bigcap_{C \in P_n} GF_2(I_C) \setminus \bigcup_{C \in W_n \setminus P_n} GF_2(I_C) \neq \emptyset.$$

Now, suppose

$$\bigcap_{C \in P_n} GF_2(I_C) \setminus \bigcup_{C \in W_n \setminus P_n} GF_2(I_C) \neq \emptyset.$$

Then there exists at least one weight vector ω such that $\prec_\omega$ yields a quadratic Gröbner basis for each inductively 1-pierced nerual code on n neurons and does not yield a quadratic Gröbner basis for any other well formed neural codes. □

2.4 Examples and Computations

We compute the minimum and maximum degree of the Gröbner basis of the toric ideal of each neural code on 3 neurons. This computation was performed in Sage using the interface to the package Gfan.

Analysis of this table reveals that a well formed neural code on three neurons is inductively pierced if its minimal and maximal Gröbner degrees are 2. In other words, a well formed neural code on three neurons is inductively pierced if and only if it has a quadratic univeral Gröbner basis. Is this true in general? No. The neural code $C = \{0001, 0011, 0010, 0110, 0100, 1100, 1000\}$ is 1-inductively pierced, but has maximum Gröbner degree 3.

code	min degree	max degree
[[1, 0, 0] , [0, 1, 1] , [1, 1, 1]]	2	2
[[1, 0, 0] , [0, 1, 0] , [0, 0, 1] , [1, 1, 1]]	3	3
[[1, 0, 0] , [0, 1, 0] , [1, 1, 0] , [1, 0, 1]]	2	2
[[1, 0, 0] , [0, 1, 0] , [1, 1, 0] , [1, 1, 1]]	2	2
[[1, 0, 0] , [0, 1, 0] , [1, 0, 1] , [1, 1, 1]]	2	2
[[1, 0, 0] , [1, 1, 0] , [1, 0, 1] , [1, 1, 1]]	2	2
[[1, 0, 0] , [1, 1, 0] , [0, 1, 1] , [1, 1, 1]]	2	2
[[1, 1, 0] , [1, 0, 1] , [0, 1, 1] , [1, 1, 1]]	3	3
[[1, 0, 0] , [0, 1, 0] , [0, 0, 1] , [1, 1, 0] , [1, 0, 1]]	2	2
[[1, 0, 0] , [0, 1, 0] , [0, 0, 1] , [1, 1, 0] , [1, 1, 1]]	2	3
[[1, 0, 0] , [0, 1, 0] , [1, 1, 0] , [1, 0, 1] , [1, 1, 1]]	2	2
[[1, 0, 0] , [0, 1, 0] , [1, 0, 1] , [0, 1, 1] , [1, 1, 1]]	2	2
[[1, 0, 0] , [1, 1, 0] , [1, 0, 1] , [0, 1, 1] , [1, 1, 1]]	2	3
[[1, 0, 0] , [0, 1, 0] , [0, 0, 1] , [1, 1, 0] , [1, 0, 1] , [0, 1, 1]]	2	3
[[1, 0, 0] , [0, 1, 0] , [0, 0, 1] , [1, 1, 0] , [1, 0, 1] , [1, 1, 1]]	2	3
[[1, 0, 0] , [0, 1, 0] , [1, 1, 0] , [1, 0, 1] , [0, 1, 1] , [1, 1, 1]]	2	3
[[1, 0, 0] , [0, 1, 0] , [0, 0, 1] , [1, 1, 0] , [1, 0, 1] , [0, 1, 1] , [1, 1, 1]]	2	3

Table 2.1 Minimum and maximum degrees of Gröbner bases for all neural codes on 3 neurons.

Chapter 3

Codes in General Position and Approximation With Polytopes

The authors of Cruz et al. (2016) introduce the idea of realizations of neural codes which are in *general position*. To give their definition, we need to define the *Hausdorff Distance*.

Definition 3.1 (Hausdorff Distance)**.** *Let X and Y be subsets of a metric space U. The Hausdorff distance between X and Y is defined by*

$$d_H(X, Y) = \max\{\sup_{x \in X} \inf_{y \in Y} d(x, y), \sup_{y \in Y} \inf_{x \in X} d(x, y)\}.$$

Definition 3.2 (General Position)**.** *A realization $\mathcal{U} = \{U_1, \ldots, U_n\}$ is in general position if there exists $\epsilon > 0$ such that for all covers $\mathcal{V} = \{V_1, \ldots, V_n\}$ such that U_i and V_i are within Hausdorff distance ϵ, the neural codes $C(\mathcal{V})$ and $C(\mathcal{U})$ are equal.*

In other words, if a realization is in general position, we can deform the place fleids a little bit without changing the neural code. While Cruz et. al. show that general position is too strong a condition for their purposes (detecting when a neural code has both a closed convex and an open convex realization), it is reasonable to assume biologically meaningful neural codes are in general position.

We prove that if a neural code posseses a convex realization in general position, it posseses a realization with convex polytopes. We take the following definition of a convex polytope:

Definition 3.3 (Convex Polytope)**.** *The convex hull of a set of points is the smallest convex set containing these points. A convex polytope is the convex hull of a fine number of points in* $\mathbb{R}^n$*.*

Note that a convex polytope in two dimensions is an ordinary convex polygon.

Lemma 3. *Let* U *be a compact convex set and* $U' = \bigcup_{x \in U} B_\epsilon(x)$*. Then there exists a convex polytope* P *such that* $U \subset P \subset U'$*.*

Proof. Let $U \subset \mathbb{R}^d$. For each $x \in U$, let $v_1, \ldots, v_{d+1} \in U' \setminus U$ such that their convex hull is an d simplex S_x such that x is contained in the interior of S_x. Note that the set $\{\text{int}(S_x)\}_{x \in U}$ forms an open cover of U. Since U is compact, there exists a finite subcover $\mathcal{S} = \{\text{int}(S_1, \ldots, S_m\}$. Now, let V be the set of vertices of the simplices whose interiors get used in our open cover, let P be the convex hull of V. We claim that $U \subset P \subset U'$. To prove $P \subset U'$, note that each $v \in V$ is contained in U'. Then since P is the smallest convex set containing V and U' is a convex set containing V, $P \subset U'$. Next, we show that $U \subset P$. Note that since $\mathcal{S}$ is a cover of U, each $x \in U$ is contained in $\text{int}(S_i)$ for some i, there exist $v_1, \ldots, v_{d+1} \subset V$ such that x is contained in the interior of the convex hull of $v_1, \ldots, v_{d+1}$. Then x is contained in $\text{conv}(V) = P$. Thus $U \subset P$, completing our proof. □

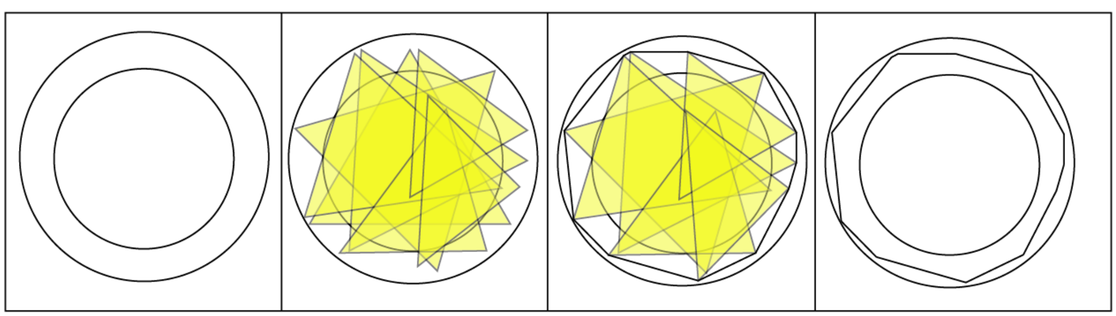

Figure 3.1 An illustration of the procedure used to construct P in lemma 3.

Theorem 5. *Let* C *be a neural code such that there exists a convex realization* $\mathcal{U} = \{U_1, \ldots, U_n\}$ *in general position. Then there exists a convex realization* $\mathcal{P} = \{P_1, \ldots, P_n\}$ *such that each* P_i *is a convex polytope.*

Proof. Let $\mathcal{U} = \{U_1, \ldots, U_n\}$ be in general position for some $\epsilon > 0$. We show that we can find P_i is within Hausdorff distance ϵ of U_i. Let $(U_i)_\epsilon = \bigcup B_\epsilon(x)_{x \in U_i}$. Then $(U_i)_\epsilon$ is in Hausdorff distance ϵ of U_i, since each point of U_i is a point of $(U_i)_\epsilon$, and each point of $(U_i)_\epsilon$ is within distance ϵ of some point in U_i. Now, using our lemma, we find P_i such that $U_i \subseteq P_i \subseteq (U_i)_\epsilon$. Then $d_H(U_i, P_i) \leq \epsilon$. This follows because $U_i \subset P_i \subset (U_i)_\epsilon$, so every point of U_i is distance 0 from a point of P_i and every point of P_i is distance at most ϵ from a point of U_i.

Since we required the realization to be in general position, $\mathcal{C}(\mathcal{P}) = \mathcal{C}(\mathcal{U}) = \mathcal{C}$. Thus, there is a realization of $\mathcal{C}$ using convex polytopes.

□

Note that the converse of this proof isn't true: there exist neural codes which cannot be realized in general position, but can be realized with convex polytopes. For instance, the code

$$\begin{aligned}\mathcal{C} = \{&111000, 110001, 100011, 000111, 001110, 011100,\\ &110000, 100001, 000011, 000110, 001100, 011000, \emptyset\},\end{aligned}$$

introduced in Cruz et al. (2016), is convex open but not convex closed. Therefore, it cannot be realized in general position. However, we give a realization with convex polygons based on the orignal open realization in Figure 3.2.

We have not yet found an example of a neural code for which there exists a convex realization, but not a convex realization with convex polytopes. This motivates the following question:

Question 1. *Is a neural code convex if and only if there exists a realization with convex polytopes*?

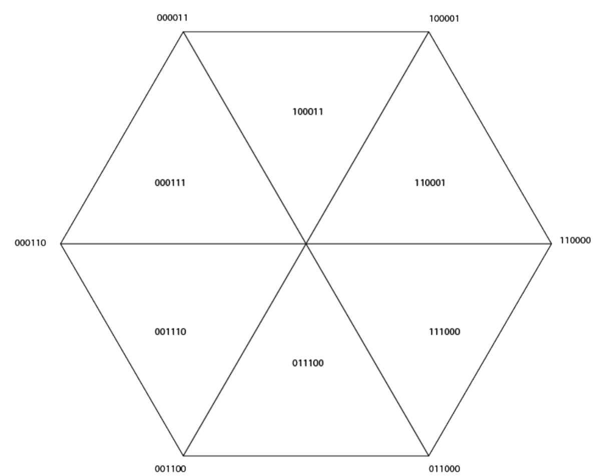

Figure 3.2 A neural code which is realizable with convex polygons, but not in general position. Note that the region labels which fall outside the hexagon refer to regions along the boundaries of the nearest triangles.

Chapter 4

Future Work

A number of open questions remain. First and foremost, we intend to settle Conjecture 2.3, perhaps through further exploration of the 2-fan. After this, several interesting generalizations of this conjecture remain. Are there similar ways to characterize inductively k-pierced codes, for $k > 1$? A guess at how to extend Conjecture 2.3, proposed in Gross et al. (2016), would be to conjecture the following:

Conjecture 2. *For each k and n, there exists a term order such a code on n neurons is m-inductively pierced, for $m \leq k$, if and only if the reduced Gröbner basis contains binomials of degree $k + 1$ or less.*

In our proof that inductively pierced neural codes have quadratic Gröbner bases, we showed that neural codes which are inductively built–that is, in which new place fields are added in ways which do not cover up old zones–have nice relationships between toric ideals of intermediate codes. That is, recall theorem 2 that $I_C \cong J \subset I_{C'}$ if C' is obtained from C using an inductive building operation. What more can we say about toric ideals of inductively built codes? Can we identify codes built in this way or identify which codes built in this way are convex via the toric ideal?

A problem with seeking to distinguish inductively pierced codes from other well formed codes is that it is difficult to determine which neural codes are well formed in the first place. Furthermore, it would be interesting to identify inductively pierced neural codes because, while a number of biologically feasible neural codes are not inductively pierced, it is likely that biologically realistic neural codes should be well formed. Each of the three ways a convex neural code can fail to be well formed is problematic biologically. If place fields are chosen with any degree of randomness, even subject to environmental constraints, the probability that a triple intersection of place fields will occur is zero, thus we should not expect to see triple intersections in real life neural codes. Similarly, the probability that the boundaries of two place fields intersect in an infinite number of points would be zero if place fields are laid down randomly, however, we might expect two place fields to share a portion of their boundary if, for instance, this shared boundary represented a barrier in the environment. Finally, a disconnected zone would be disadvantageous to an animal relying on place fields for navigation, since it would not be able to distinguish between two or more spatially separated regions.

Here, note the parallels to convex codes and codes in general position. While well formed nerual codes are not explicitly required to be convex, since the boundaries of place fields are required to be simple closed curves in $\mathbb{R}^2$, all place fields are required to be contractible. However, intersections between place fields are not required to be contractible. Thus, well formed neural codes are not required to be good covers. Note that realizations of neural codes as good covers–that is, where every place field and every nonempty intersection of place fields is contractible–are not required to be well formed, since zones only have to be connected, not simply connected. Requiring place fields to be convex does not fix this problem. If a neural code is convex and in general position, then it can be realized without a triple intersection and without two place fields sharing a boundary. However, again, it can have disconnected zones. Therefore convex codes, convex codes in general position, and well formed codes are distinct, but related, classes of neural codes.

Codes which can be realized with convex polytopes are also worthy of further study. Since polytopes have discrete descriptions, it seems likely they are easier to understand than general convex sets. Other more restricted cases, such as codes realizable with convex polygons in $\mathbb{R}^2$, have the potential to be tractable while being broadly applicable and shedding insight into the wider class of convex neural codes.

Bibliography

Burgess, Neil. 2014. The 2014 nobel prize in physiology or medicine: a spatial model for cognitive neuroscience. *Neuron* 84(6):1120–1125.

Burgess, Neil, and John O'Keefe. 2003. Neural representations in human spatial memory. *Trends in cognitive sciences* 7(12):517–519.

Cox, David, John Little, and Donal O'shea. 1992. *Ideals, varieties, and algorithms*, vol. 3. Springer.

Cruz, Joshua, Chad Giusti, Vladimir Itskov, and Bill Kronholm. 2016. On open and closed convex codes. *arXiv preprint arXiv:160903502* .

Curto, Carina, Elizabeth Gross, Jack Jeffries, Katherine Morrison, Mohamed Omar, Zvi Rosen, Anne Shiu, and Nora Youngs. 2017. What makes a neural code convex? *SIAM J Appl Algebra Geometry* 1:222–238.

Curto, Carina, and Vladimir Itskov. 2008. Cell groups reveal structure of stimulus space. *PLoS Comput Biol* 4(10):e1000,205.

Curto, Carina, Vladimir Itskov, Alan Veliz-Cuba, and Nora Youngs. 2013. The neural ring: an algebraic tool for analyzing the intrinsic structure of neural codes. *Bulletin of mathematical biology* 75(9):1571–1611.

Dummit, David Steven, and Richard M Foote. 2004. *Abstract algebra*, vol. 3. Wiley Hoboken.

Garcia, Rebecca, Luis David García Puente, Ryan Kruse, Jessica Liu, Dane Miyata, Ethan Petersen, Kaitlyn Phillipson, and Anne Shiu. 2016. Gröbner bases of neural ideals. *arXiv preprint arXiv:161205660* .

Grayson, Daniel R, and Michael E Stillman. 2002. Macaulay 2, a software system for research in algebraic geometry.

Gross, Elizabeth, Nida Kazi Obatake, and Nora Youngs. 2016. Neural ideals and stimulus space visualization. *arXiv preprint arXiv:160700697* .

Hemmecke, Raymond, Ralf Hemmecke, and Peter Malkin. 2003. 4ti2 version 1.2–computation of hilbert bases, graver bases, toric gröbner bases, and more. *Available at* www4ti2de .

Lienkaemper, Caitlin, Anne Shiu, and Zev Woodstock. 2017. Obstructions to convexity in neural codes. *Advances in Applied Mathematics* 85:31–59.

Petersen, Ethan, Nora Youngs, Ryan Kruse, Dane Miyata, Rebecca Garcia, and Luis David Garcia Puente. 2016. Neural ideals in sagemath. *arXiv preprint arXiv:160909602* .

Reid, Miles. 1988. *Undergraduate algebraic geometry*. Cambridge University Press Cambridge.

Sturmfels, Bernd. 1996. *Gröbner bases and convex polytopes*, vol. 8. American Mathematical Soc.

———. 2005. What is a gröbner basis. *Notices Amer Math Soc* 52(10):1199–1200.