

Algebraic foundations of machine learning

Yulia Alexandr (UCLA)

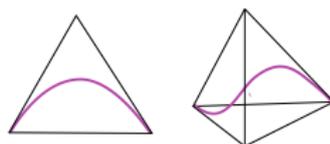
University of Florida
January 28, 2026

Overview

I use *algebraic geometry* and *combinatorics* to study the structure and properties of models that arise in applications.

- Statistics

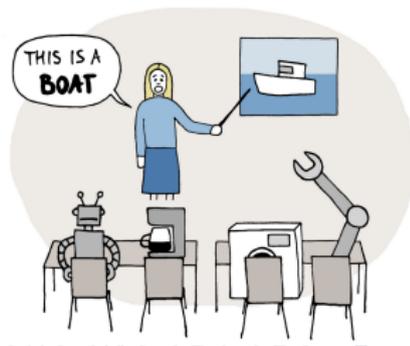
- ▶ Data privacy
- ▶ Causal inference
- ▶ Information geometry
- ▶ Phylogenetics



MACHINE LEARNING

- Machine learning

- ▶ Guarantees for model outputs
- ▶ Neural network verification
- ▶ Safety-critical and trustworthy AI



Algebraic toolkit

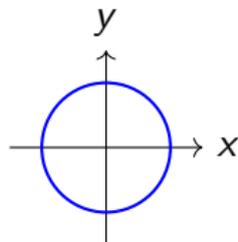
Models as algebraic varieties

An *algebraic variety* is the set of all points that satisfy a system of polynomial equations.

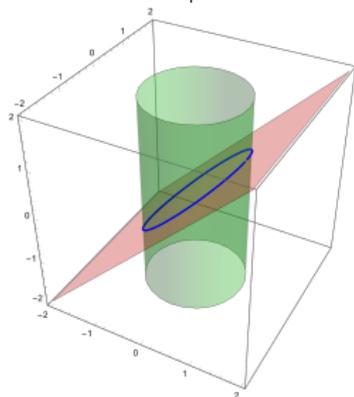
Ideal

$$\langle x^2 + y^2 - 1 \rangle$$

Variety



$$\langle x^2 + y^2 - 1, x - z \rangle$$



The independence model

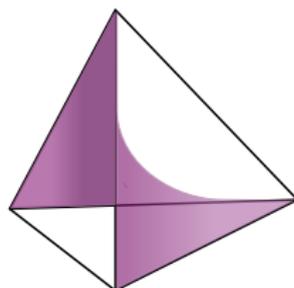
Many statistical models are made up of distributions whose coordinates satisfy polynomial equations.

Example

Let X_1 and X_2 be binary random variables. Let $p_{ij} = \mathbb{P}(X_1 = i, X_2 = j)$. Then X_1 and X_2 are independent if and only if

$$\det \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = p_{11}p_{22} - p_{12}p_{21} = 0.$$

Recall that X_1 and X_2 are *independent* if

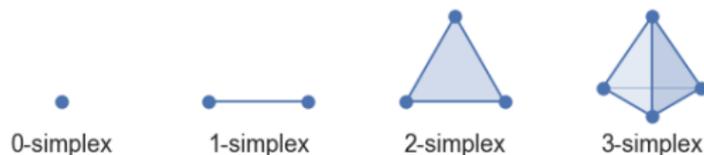
$$\mathbb{P}(X_1 = i, X_2 = j) = \mathbb{P}(X_1 = i) \cdot \mathbb{P}(X_2 = j).$$


Algebraic statistics

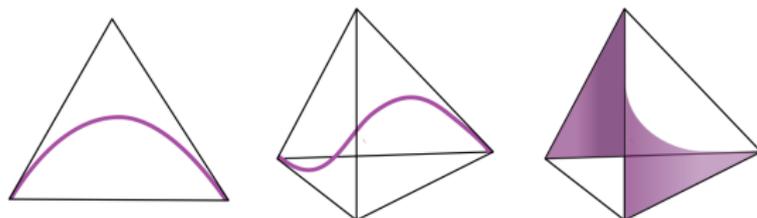
Statistical models

A *probability simplex* is defined as

$$\Delta_{n-1} = \{(p_1, \dots, p_n) : p_1 + \dots + p_n = 1, p_i \geq 0 \text{ for } i \in [n]\}.$$



An *algebraic statistical model* is an intersection of some variety \mathcal{V} with the probability simplex.

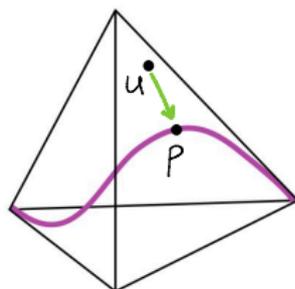


Maximum likelihood estimation

Let $\mathcal{M} = \mathcal{V} \cap \Delta_{n-1}$ be a statistical model.

Given empirical data $u \in \Delta_{n-1}$, the *maximum likelihood estimation (MLE)* problem is to find $p \in \mathcal{M}$ that maximizes:

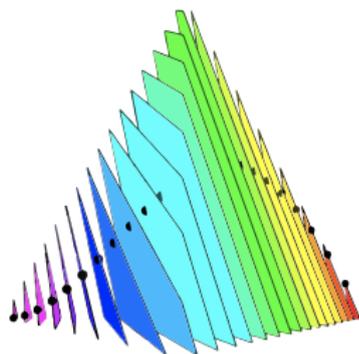
$$\ell_u(p) = \sum_{i=1}^n u_i \log p_i.$$



For a $q \in \mathcal{M}$, the set of all data $u \in \Delta_{n-1}$ that maximize to q is the *logarithmic Voronoi cell* at q .

Proposition (A.–Heaton)

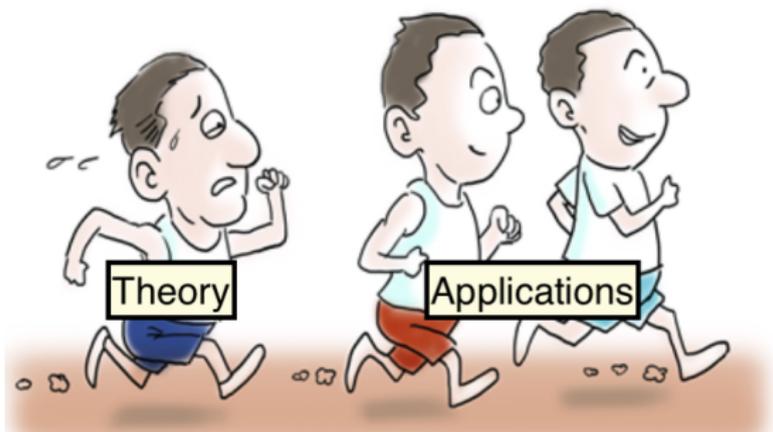
Logarithmic Voronoi cells are convex sets.



Why do we care?

- **Data privacy**, especially for **statistical disclosure limitation**
 - ▶ Logarithmic Voronoi cells at singular and boundary points of the model
 - **Estimate sensitivity** to data perturbation
 - ▶ Boundaries of logarithmic Voronoi cells (open problem)
 - **Maximizing divergence** to a toric model
 - ? Which data point $u \in \Delta_{n-1}$ is farthest from its MLE?
 - ▶ Developed an algorithm to find the global maximum and maximizers using logarithmic Voronoi cells and chamber complexes
- ⇒ Neural networks develop in such a way to maximize the mutual information between the input and output of each layer

Algebraic machine learning



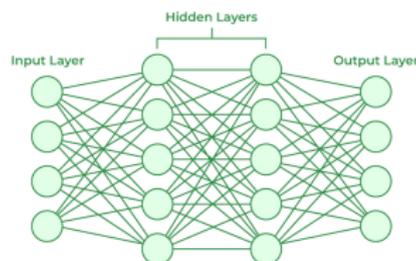
Neural networks

Any **feedforward neural network** with an activation function σ gives rise to

$$f_{\theta} : x \mapsto g_L \circ \sigma \circ g_{L-1} \dots \sigma \circ g_1(x)$$

where each layer has linear map $g_{\ell} : y \mapsto W^{(\ell)}y$ with parameter $\theta_{\ell} = W^{(\ell)}$.

- $n_0 =$ dimension of the input space
- $n_{\ell} =$ width of the ℓ th layer



For a dataset $X = [x_1, x_2, \dots, x_n]$ and unknown parameters θ we want to describe the **constraints** between the coordinates of the model outputs

$$F_X(\theta) = [f_{\theta}(x_1), f_{\theta}(x_2), \dots, f_{\theta}(x_n)].$$

ReLU invariants

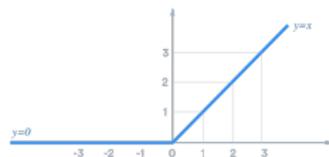
ReLU networks

A *ReLU network* is given by the activation function

$$\sigma : y = (y_1, \dots, y_{n_\ell}) \mapsto (\max\{0, y_1\}, \dots, \max\{0, y_{n_\ell}\})$$

at each layer of the neural network.

- $f_\theta(x)$ is piece-wise linear
 \implies subdivision of the **input space** into linear regions
- $F_X(\theta)$ is piece-wise multi-linear for fixed data X
 \implies subdivision of the **parameter space** into **activation regions**



Takeaway:

restricting to an activation region A turns $F_X(\theta)$ into a polynomial map!

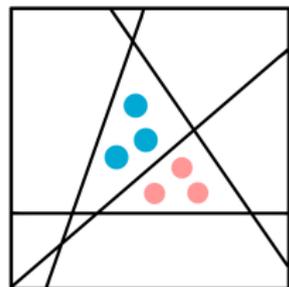
The main question

Problem

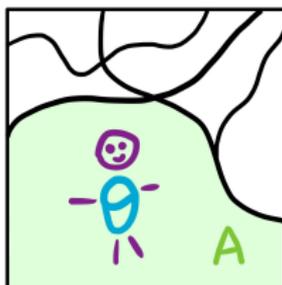
Describe the equations and inequalities that define the image of the map

$$\varphi_X^A : \theta \mapsto F_X^A(\theta)$$

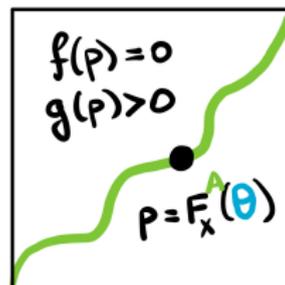
as θ varies over an activation region A in the parameter space.



input space

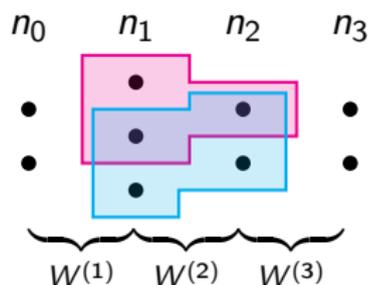


parameter space



prediction space

The parametrization



$$X = [x_1, x_2, x_3, x_4]$$

$$a_1 = [(1, 1, 0), (1, 0)]$$

$$a_2 = [(0, 1, 1), (1, 1)]$$

$$A = [a_1, a_1, a_2, a_2]$$

$$M_1(\theta) = W^{(3)} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} W^{(2)} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} W^{(1)}$$

$$M_2(\theta) = W^{(3)} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} W^{(2)} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} W^{(1)},$$

$$\varphi_X^A : \theta = (W^{(1)}, W^{(2)}, W^{(3)}) \mapsto [M_1(\theta) [x_1 \ x_2] \mid M_2(\theta) [x_3 \ x_4]]$$

The parametrization

In general, for k blocks:

$$\varphi_X^A : \theta \mapsto [M_1(\theta)X_1 \mid M_2(\theta)X_2 \mid \cdots \mid M_k(\theta)X_k].$$

ReLU output variety: $V_X^A = \overline{\text{im}(\varphi_X^A)}$ with ideal I_X^A .

$$\varphi^A : \theta \mapsto [M_1(\theta) \mid M_2(\theta) \mid \cdots \mid M_k(\theta)].$$

ReLU pattern variety: $U^A = \overline{\text{im}(\varphi^A)}$ with ideal J^A .

Proposition (A.-Montúfar)

Any polynomial $f \in J^A$ gives rise to a unique polynomial $g = \psi^{-1}f \in I_X^A$, where ψ is a linear change of coordinates dependent on X .

So, we can study the ideal J^A of the pattern variety instead!

Implicitization

Given a model, parametrized by

$$\varphi : \theta = (\theta_1, \dots, \theta_n) \mapsto (f_1(\theta), f_2(\theta), \dots, f_m(\theta)),$$

we are interested in describing the polynomials defining $\overline{\text{image}(\varphi)}$. This process is called *implicitization*.

Implicitization

Given a model, parametrized by

$$\varphi : \theta = (\theta_1, \dots, \theta_n) \mapsto (f_1(\theta), f_2(\theta), \dots, f_m(\theta)),$$

we are interested in describing the polynomials defining $\overline{\text{image}(\varphi)}$. This process is called *implicitization*.

Example (The independence model.)

Parametrization:

$$(\theta_1, \theta_2) \mapsto (\underbrace{\theta_1 \theta_2}_{p_1}, \underbrace{\theta_1(1 - \theta_2)}_{p_2}, \underbrace{(1 - \theta_1)\theta_2}_{p_3}, \underbrace{(1 - \theta_1)(1 - \theta_2)}_{p_4}).$$

Implicit ideal: $I = \langle p_1 p_4 - p_2 p_3, p_1 + p_2 + p_3 + p_4 - 1 \rangle$.



The generators of the ideal I are called *model invariants*.

Why do we care?

- Characterization of reachable outputs
 - ▶ does a point lie in the image of a neural network map?
- Safety verification
 - ▶ can the network ever produce an unsafe or undesired output?
 - ▶ let S be the set of unsafe outputs
 - ▶ if $V_X^A \cap S = \emptyset$, the network is **provably safe**
- Symbolic extrapolation
 - ▶ use invariants to reason about outputs even far from the training data
 - ▶ guarantee behavior without relying on over-approximation

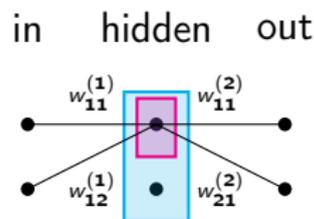


Takeaway: invariants = provable guarantees!

Example: 2 blocks

Consider a general dataset $X = [x_1, x_2, x_3, x_4]$.

- $X_1 = [x_1, x_2]$ follow the pattern $(1, 0)$.
- $X_2 = [x_3, x_4]$ follow the pattern $(1, 1)$.



ReLU output variety: $\theta \mapsto [M_1(\theta)X_1 \mid M_2(\theta)X_2]$ with $\theta = (W^{(1)}, W^{(2)})$

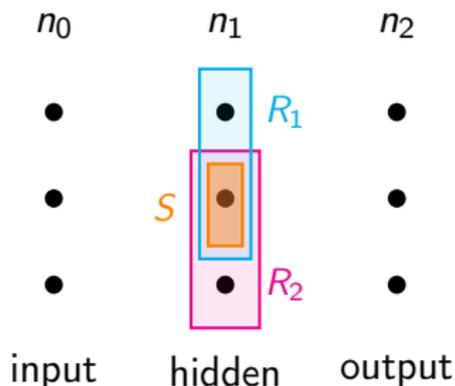
$$M_1(\theta) = \begin{pmatrix} w_{11}^{(1)} & w_{11}^{(2)} \\ w_{11}^{(1)} & w_{21}^{(2)} \end{pmatrix}, M_2(\theta) = \begin{pmatrix} w_{11}^{(1)} w_{11}^{(2)} + w_{21}^{(1)} w_{12}^{(2)} & w_{12}^{(1)} w_{11}^{(2)} + w_{22}^{(1)} w_{12}^{(2)} \\ w_{11}^{(1)} w_{21}^{(2)} + w_{21}^{(1)} w_{22}^{(2)} & w_{12}^{(1)} w_{21}^{(2)} + w_{22}^{(1)} w_{22}^{(2)} \end{pmatrix}.$$

ReLU pattern variety: $\theta \mapsto [M_1(\theta) \mid M_2(\theta)]$

$$J^A = \langle \det(M_1), \det(M_2 - M_1) \rangle.$$

The ideal I_X^A is obtained from J^A in terms of fixed but arbitrary data X_1, X_2 .

Two blocks, shallow networks



Let $|R_1| = r_1$, $|R_2| = r_2$, $|S| = s$.
Let $t = r_1 + r_2 - 2s$.

Theorem (A.-Montúfar)

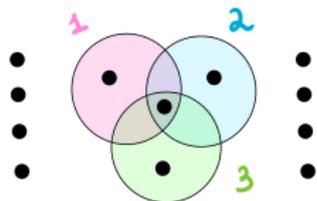
The ideal J^A contains:

- 1 $(r_1 + 1)$ -minors of M_1 ;
- 2 $(r_2 + 1)$ -minors of M_2 ;
- 3 $(n_1 + 1)$ -minors of $[M_1 \mid M_2]$ and $[M_1^T \mid M_2^T]$;
- 4 $(t + 1)$ -minors of $M_1 - M_2$.

Conjecture: no other polynomials are needed to generate the ideal.

Generalizations

- Two blocks (deep networks):
 - ▶ similar to the shallow case, but now we count shared *paths*.
- Multiple blocks:
 - ▶ rank conditions appear on linear combinations $\sum_i \lambda_i M_i$,
 - ▶ and on block matrices built from these combinations.



2,256 invariants of degree 3, 4, and 5!

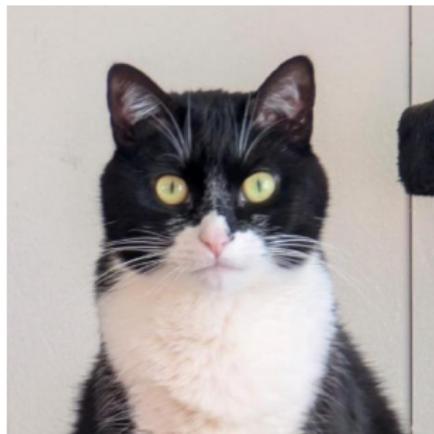
- Dimension:
 - ▶ expected dimension holds when n_1 is small

Open problems:

- Sufficiency of generators for the shallow case
- Finite representation for deep networks with multiple blocks
- Inequalities defining the image

Robustness verification

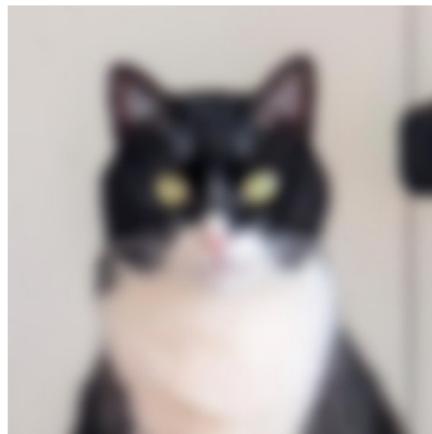
Robustness to data perturbation



Input ξ

Prediction: "Cat"

Perturbation \rightarrow
 $\|\delta\| < \epsilon$



Input $\xi + \delta$

Prediction: "Toaster"

Question: Can we *prove* that no such δ exists?

Algebraic verification

The model

Polynomial map $f_\theta : \mathbb{R}^n \rightarrow \mathbb{R}^k$ with decision boundary

$$\mathcal{V} = \{x \in \mathbb{R}^n : f_{\theta,c}(x) = f_{\theta,c'}(x) \text{ for some } c \neq c'\}.$$

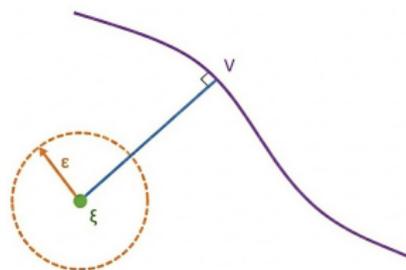
The goal

Verify that the prediction for input ξ is constant within a radius ε .

The geometric translation

We replace the check with a distance calculation:

$$\text{Robustness at } \xi \iff \min_{x \in \mathcal{V}} \|x - \xi\| > \varepsilon$$



The algebraic complexity

To find the closest point on \mathcal{V} , we must compute all complex critical points. The number of these points is bounded by the [Euclidean Distance degree](#).

Complexity of verification

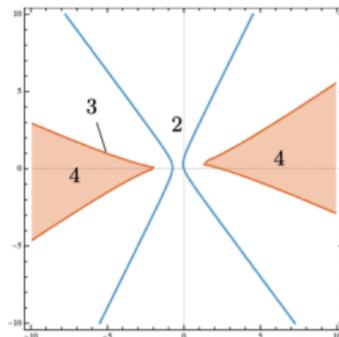
1. How complex is verification generically?

Theorem (A.-Duan-Montúfar):

For a network (n, h, k) with generic parameters and degree d activation, the ED degree is:

$$\text{EDdeg}(\mathcal{V}) = d \sum_{i=0}^{m-1} (d-1)^i, \quad m = \min(n, h).$$

Exact formula for certain **deep** architectures.



2. Which data points are unstable?

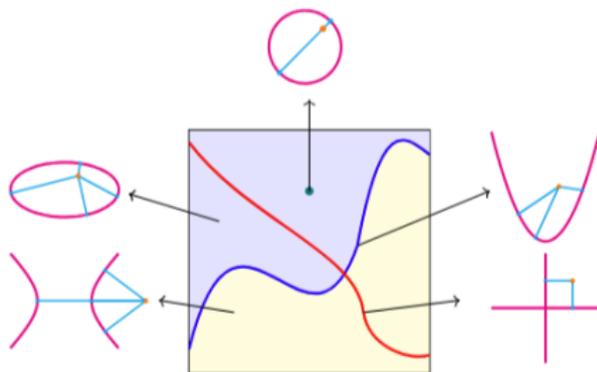
ED discriminant = set of **data points** where critical points collide.

A.-Duan-Montúfar: symbolic algorithm for hypersurfaces.

3. Which parameters simplify complexity?

Parameter discriminant = set of **parameters** where ED degree drops.

Theorem (A.-Duan-Montúfar): full characterization for quadratic case.



Research program

Algebraic statistics and machine learning

- Geometry in statistical inference
 - ▶ Describe (logarithmic) Voronoi cells at singular points
- Algebraic invariants
 - ▶ Derive invariants for self-attention mechanisms
 - ★ Extend to other architectures
 - ▶ Incorporate invariants into neural network verification
 - ★ Theoretical guarantees
 - ★ Computational package via Homotopy Continuation
- Impact and mentorship
 - ▶ Expand algebraic methods into [your favorite field]
 - ▶ Build a research group bridging theory and real-world impact

Thank you! Questions?

**VARIETY
IS THE
SPICE OF LIFE**

