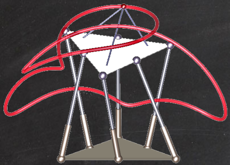
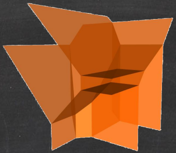
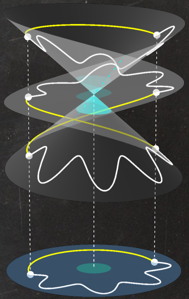
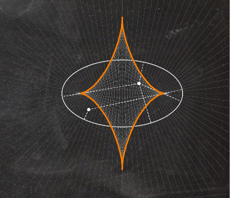






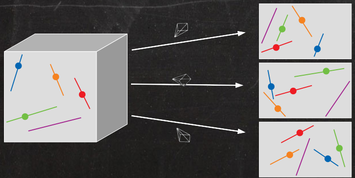
Algebra & Geometry in Data Science & AI



Kathlén Kohn

KTH VETENSKAP OCH KONST

WASP WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM



data science & AI require a vast math toolbox

optimization

machine learning

statistics

algebra & geometry

scientific computing

analysis

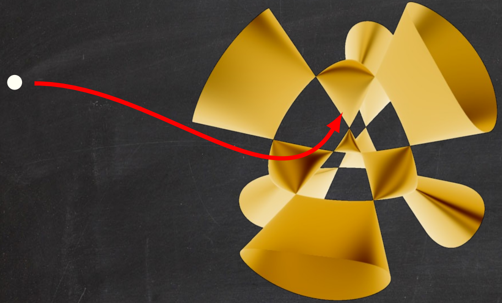
...

The world is non-linear!

Many models in the sciences and engineering are characterized by polynomial equations. Such a set is an algebraic variety $X \subset \mathbb{R}^n$.

Varieties look like manifolds almost everywhere, but typically have singularities.

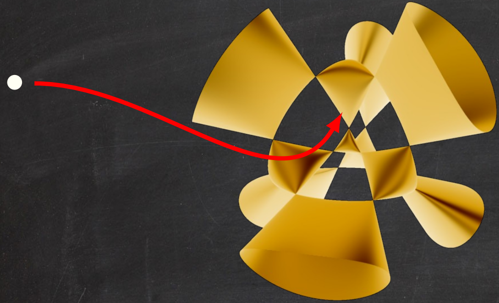
Varieties in data science & AI



algebraic optimization

given •, find best point on (possibly unknown) manifold, variety, etc.

Varieties in data science & AI



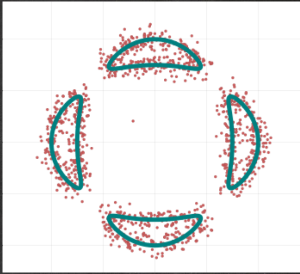
algebraic optimization

given •, find best point on (possibly unknown) manifold, variety, etc.

manifold hypothesis

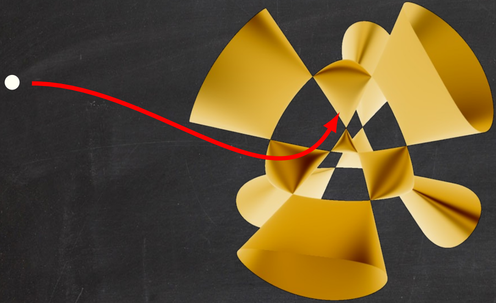
variety hypothesis

data comes from low-dimensional manifold, variety, etc.



want to infer information about underlying manifold, variety, etc.

Varieties in data science & AI



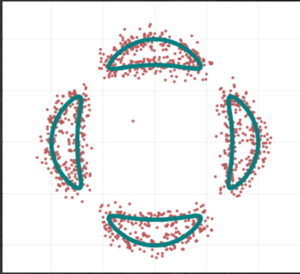
algebraic optimization

given •, find best point on (possibly unknown) manifold, variety, etc.

manifold hypothesis

variety hypothesis

data comes from low-dimensional manifold, variety, etc.



want to infer information about underlying manifold, variety, etc.

algebraic inverse problems

given observations, want to recover ground truth

Netflix problem

Alice	1			4		
Bob		2	5			
Carol			4	5		
Dave	5				4	
⋮						

What are the unknown ratings?

Netflix problem

						⋯
Alice	1			4		
Bob		2	5			
Carol			4	5		
Dave	5				4	
⋮						

Guess: This matrix should be of low rank!

Netflix problem

						…
Alice	1			4		
Bob		2	5			
Carol			4	5		
Dave	5				4	
⋮						

Guess: This matrix should be of low rank!

Underlying variety is
$\{A \in \mathbb{R}^{\#\text{users} \times \#\text{movies}} \mid \operatorname{rank}(A) \leq r\}$.

What is r ??

Netflix problem

						...
Alice	1			4		
Bob		2	5			
Carol			4	5		
Dave	5				4	
⋮						

Guess: This matrix should be of low rank!

Underlying variety is
$\{A \in \mathbb{R}^{\#\text{users} \times \#\text{movies}} \mid \mathrm{rank}(A) \leq r\}$.

What is r ??

Complete the matrix such that it has rank r !

inverse problem

Netflix problem

						…
Alice	1			4		
Bob		2	5			
Carol			4	5		
Dave	5				4	
⋮						

Guess: This matrix should be of low rank!

Underlying variety is
$\{A \in \mathbb{R}^{\#\text{users} \times \#\text{movies}} \mid \operatorname{rank}(A) \leq r\}$.

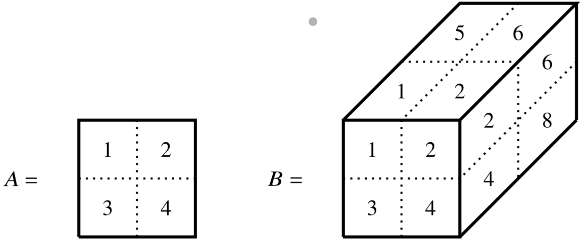
What is r ??

Complete the matrix such that it has rank r ! inverse problem

Complete the matrix such that it is close to a rank-r matrix ! optimization

Big Data & Tensors

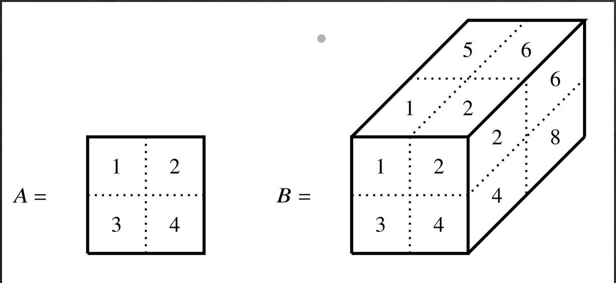
Often, data has many dimensions to it!



A 2×2 matrix A and a $2 \times 2 \times 2$ tensor B.

Big Data & Tensors

Often, data has many dimensions to it!



A 2×2 matrix A and a $2 \times 2 \times 2$ tensor B.

Big data gives rise to huge, high-dimensional tensors.

$\rightsquigarrow$ need to understand **tensor rank**, their **eigenvectors**, etc.

Maximum Likelihood Estimation

Experiment: Toss a biased coin twice, and record the total number of heads

Task: From many such experiments, recover the bias of the coin

Maximum Likelihood Estimation

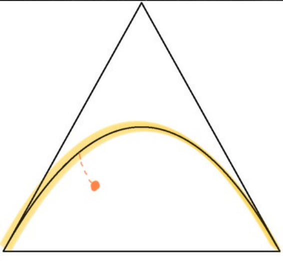
Experiment: Toss a biased coin twice, and record the total number of heads

Task: From many such experiments, recover the bias of the coin

The possible distributions of the experiment outcome are parametrized by

$$[0,1] \longrightarrow \Delta_2 := \{(P_0, P_1, P_2) \in \mathbb{R}^3_{\geq 0} \mid P_0 + P_1 + P_2 = 1\},$$
$$p \longmapsto (p^2, \quad 2p(1-p), \quad (1-p)^2)$$

head-head *head-tail & tail-head* *tail-tail*



Maximum Likelihood Estimation

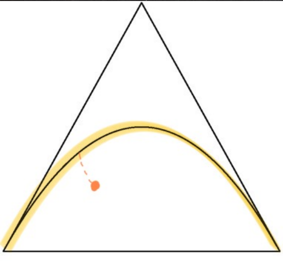
Experiment: Toss a biased coin twice, and record the total number of heads

Task: From many such experiments, recover the bias of the coin

The possible distributions of the experiment outcome are parametrized by

$$[0,1] \longrightarrow \Delta_2 := \{(P_0, P_1, P_2) \in \mathbb{R}^3_{\geq 0} \mid P_0 + P_1 + P_2 = 1\},$$
$$p \longmapsto (p^2, \quad 2p(1-p), \quad (1-p)^2)$$

head-head *head-tail & tail-head* *tail-tail*



After n experiments, the vector of counts $u = (u_0, u_1, u_2)$ provides an empirical distribution $\frac{1}{n}u$.

Maximum Likelihood Estimation

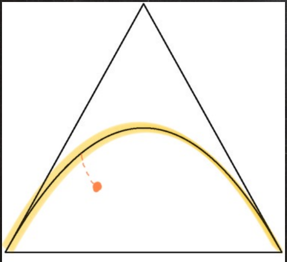
Experiment: Toss a biased coin twice, and record the total number of heads

Task: From many such experiments, recover the bias of the coin

The possible distributions of the experiment outcome are parametrized by

$$[0,1] \longrightarrow \Delta_2 := \{(P_0, P_1, P_2) \in \mathbb{R}^3_{\geq 0} \mid P_0 + P_1 + P_2 = 1\},$$
$$p \longmapsto (p^2, \quad 2p(1-p), \quad (1-p)^2)$$

head-head *head-tail & tail-head* *tail-tail*



After n experiments, the vector of counts $u = (u_0, u_1, u_2)$ provides an empirical distribution $\frac{1}{n}u$.

The likelihood that the bias p gave rise to u is $(p^2)^{u_0} \cdot (2p(1-p))^{u_1} \cdot ((1-p)^2)^{u_2}$.

Maximum Likelihood Estimation

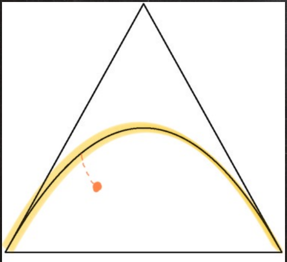
Experiment: Toss a biased coin twice, and record the total number of heads

Task: From many such experiments, recover the bias of the coin

The possible distributions of the experiment outcome are parametrized by

$$[0,1] \longrightarrow \Delta_2 := \{(P_0, P_1, P_2) \in \mathbb{R}^3_{\geq 0} \mid P_0 + P_1 + P_2 = 1\},$$
$$p \longmapsto (p^2, \quad 2p(1-p), \quad (1-p)^2)$$

head-head   *head-tail & tail-head*   *tail-tail*



After n experiments, the vector of counts $u = (u_0, u_1, u_2)$ provides an empirical distribution $\frac{1}{n}u$.

The likelihood that the bias p gave rise to u is $(p^2)^{u_0} \cdot (2p(1-p))^{u_1} \cdot ((1-p)^2)^{u_2}$.

The p maximizing this most likely gave rise to u. It is called the maximum likelihood estimate (MLE).

MLE of matrix normal distributions

Multivariate normal distribution for matrix-valued random variable X of format $m \times n$ has probability density function

$$\frac{\exp(-\frac{1}{2}\mathrm{tr}[V^{-1}(X-M)^\top U^{-1}(X-M)])}{(2\pi)^{\frac{mn}{2}} \det(V)^{\frac{m}{2}} \det(U)^{\frac{n}{2}}},$$

where $M \in \mathbb{R}^{m\times n}$, $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$.

MLE of matrix normal distributions

Multivariate normal distribution for matrix-valued random variable X of format $m \times n$ has probability density function

$$\frac{\exp(-\frac{1}{2}\mathrm{tr}[V^{-1}(X-M)^\top U^{-1}(X-M)])}{(2\pi)^{\frac{mn}{2}}\det(V)^{\frac{m}{2}}\det(U)^{\frac{n}{2}}},$$

where $M \in \mathbb{R}^{m\times n}$, $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$.

Equivalently, the vectorization $\mathrm{vec}(X)$ is distributed as the standard multivariate normal distribution with mean vector $\mathrm{vec}(M)$ and covariance matrix

$$V \otimes U := \begin{bmatrix} v_{11}U & \cdots & v_{1n}U \\ \vdots & & \vdots \\ v_{n1}U & \cdots & v_{nn}U \end{bmatrix} \in \mathbb{R}^{mn\times mn}.$$

MLE of matrix normal distributions

Multivariate normal distribution for matrix-valued random variable X of format $m \times n$ has probability density function

$$\frac{\exp(-\frac{1}{2}\mathrm{tr}[V^{-1}(X-M)^\top U^{-1}(X-M)])}{(2\pi)^{\frac{mn}{2}} \det(V)^{\frac{m}{2}} \det(U)^{\frac{n}{2}}},$$

where $M \in \mathbb{R}^{m\times n}$, $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$.

Equivalently, the vectorization $\mathrm{vec}(X)$ is distributed as the standard multivariate normal distribution with mean vector $\mathrm{vec}(M)$ and covariance matrix

$$V \otimes U := \begin{bmatrix} v_{11}U & \cdots & v_{1n}U \\ \vdots & & \vdots \\ v_{n1}U & \cdots & v_{nn}U \end{bmatrix} \in \mathbb{R}^{mn\times mn}.$$

All such covariance matrices are parametrized via the group $\mathrm{GL}_m \times \mathrm{GL}_n$:

$$g_1^\top g_1 \otimes g_2^\top g_2 = (g_1 \otimes g_2)^\top (g_1 \otimes g_2), \qquad \text{for } g_1 \in \mathrm{GL}_m, g_2 \in \mathrm{GL}_n$$

Gaussian group models

The **Gaussian group model** of a group $G \subseteq \mathrm{GL}_m$ is the set of a normal distributions on $\mathbb{R}^m$ with covariance matrices in

$$\mathcal{M}_G := \left\{ g^\top g \mid g \in G \right\}.$$

Gaussian group models

The **Gaussian group model** of a group $G \subseteq \mathrm{GL}_m$ is the set of a normal distributions on $\mathbb{R}^m$ with covariance matrices in

$$\mathcal{M}_G := \left\{ g^\top g \mid g \in G \right\}.$$

Given data samples $(Y_1, \ldots, Y_n)$ with $Y_i \in \mathbb{R}^m$, viewed as the columns of a matrix $Y \in \mathbb{R}^{m \times n}$, the logarithm of the likelihood (up constant scalars) is

$$\ell_Y(g) = n \log \det(g^\top g) - \|g \cdot Y\|_2^2.$$

Gaussian group models

The **Gaussian group model** of a group $G \subseteq \mathrm{GL}_m$ is the set of a normal distributions on $\mathbb{R}^m$ with covariance matrices in

$$\mathcal{M}_G := \left\{ g^\top g \mid g \in G \right\}.$$

Given data samples $(Y_1, \ldots, Y_n)$ with $Y_i \in \mathbb{R}^m$, viewed as the columns of a matrix $Y \in \mathbb{R}^{m \times n}$, the logarithm of the likelihood (up constant scalars) is

$$\ell_Y(g) = n \log \det(g^\top g) - \|g \cdot Y\|_2^2.$$

We want to find an MLE, i.e., a maximizer $g \in G$ of ℓ_Y !

MLE of Gaussian group models

Proposition

Under mild assumptions (satisfied by e.g. matrix normal distributions),

$$\sup_{g \in G} \ell_Y(g) = - \inf_{\tau \in \mathbb{R}_{>0}} \left(\tau \left(\inf_{h \in G \cap \mathrm{SL_m}} \|h \cdot Y\|_2^2 \right) - nm \log \tau \right).$$

MLE of Gaussian group models

Proposition

Under mild assumptions (satisfied by e.g. matrix normal distributions),

$$\sup_{g \in G} \ell_Y(g) = - \inf_{\tau \in \mathbb{R}_{>0}} \left(\tau \left(\inf_{h \in G \cap \mathrm{SL_m}} \|h \cdot Y\|_2^2 \right) - nm \log \tau \right).$$

The group $H := G \cap \mathrm{SL}_m$ acts on $\mathbb{R}^{m \times n}$ via left multiplication: $(h, Y) \mapsto h \cdot Y$.

MLE of Gaussian group models

Proposition

Under mild assumptions (satisfied by e.g. matrix normal distributions),

$$\sup_{g \in G} \ell_Y(g) = - \inf_{\tau \in \mathbb{R}_{>0}} \left(\tau \left(\inf_{h \in G \cap \mathrm{SL_m}} \|h \cdot Y\|_2^2 \right) - nm \log \tau \right).$$

The group $H := G \cap \mathrm{SL}_m$ acts on $\mathbb{R}^{m \times n}$ via left multiplication: $(h, Y) \mapsto h \cdot Y$.
The orbit of the data matrix Y is $H \cdot Y = \{h \cdot Y \mid h \in H\}$.

MLE of Gaussian group models

Proposition

Under mild assumptions (satisfied by e.g. matrix normal distributions),

$$\sup_{g \in G} \ell_Y(g) = - \inf_{\tau \in \mathbb{R}_{>0}} \left(\tau \left(\inf_{h \in G \cap \mathrm{SL_m}} \|h \cdot Y\|_2^2 \right) - nm \log \tau \right).$$

The group $H := G \cap \mathrm{SL}_m$ acts on $\mathbb{R}^{m \times n}$ via left multiplication: $(h, Y) \mapsto h \cdot Y$.

The orbit of the data matrix Y is $H \cdot Y = \{h \cdot Y \mid h \in H\}$.

An MLE can be computed in 2 steps:

1) Find a point of minimal norm in the orbit $H \cdot Y$.
2) Compute the unique value τ minimizing $\tau \|h \cdot Y\|_2^2 - nm \log \tau$.

The MLE is $\tau h^\top h$.

MLE of Gaussian group models

Proposition
Under mild assumptions (satisfied by e.g. matrix normal distributions),

$$\sup_{g \in G} \ell_Y(g) = - \inf_{\tau \in \mathbb{R}_{>0}} \left(\tau \left(\inf_{h \in G \cap \mathrm{SL_m}} \|h \cdot Y\|_2^2 \right) - nm \log \tau \right).$$

The group $H := G \cap \mathrm{SL}_m$ acts on $\mathbb{R}^{m \times n}$ via left multiplication: $(h, Y) \mapsto h \cdot Y$.
The orbit of the data matrix Y is $H \cdot Y = \{h \cdot Y \mid h \in H\}$.
An MLE can be computed in 2 steps:

1) Find a point of minimal norm in the orbit $H \cdot Y$.
2) Compute the unique value τ minimizing $\tau \|h \cdot Y\|_2^2 - nm \log \tau$.

The MLE is $\tau h^\top h$.

Algorithms from invariant theory that compute the capacity

$$\mathrm{cap}_H(Y) := \inf_{h \in H} \|h \cdot Y\|_2^2$$

can be used to compute MLEs ! [algorithmic papers by Bürgisser, Franks, Garg, Oliveira, Walter, Wigderson, ...]

Maximum Likelihood Thresholds

Given a family of distributions, how many data samples are needed for an MLE to exists almost surely?

Maximum Likelihood Thresholds

Given a family of distributions, how many data samples are needed for an MLE to exists almost surely?
How many for the MLE to be unique?
How many for the likelihood to be bounded?

Maximum Likelihood Thresholds

Given a family of distributions, how many data samples are needed for an MLE to exists almost surely?
How many for the MLE to be unique?
How many for the likelihood to be bounded?

These have been open questions for the family of all matrix normal distributions on $\mathbb{R}^{m \times n}$ (Dutilleul 1999; Lu, Zimmerman 2004; Srivastav, von Rosen, von Rosen 2008; Werner, Jansson, Stoica 2008; Rós, Bijma, de Munck, de Gunst 2016; Soloveychik, Trushin 2016; Drton, Kuriki, Hoff 2021)

Maximum Likelihood Thresholds

Given a family of distributions, how many data samples are needed for an MLE to exists almost surely? mlt_e

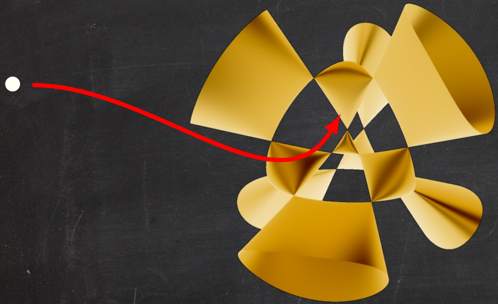
How many for the MLE to be unique? mlt_u

How many for the likelihood to be bounded? mlt_b

These have been open questions for the family of all matrix normal distributions on $\mathbb{R}^{m\times n}$ (Dutilleul 1999; Lu, Zimmerman 2004; Srivastav, von Rosen, von Rosen 2008; Werner, Jansson, Stoica 2008; Rós, Bijma, de Munck, de Gunst 2016; Soloveychik, Trushin 2016; Drton, Kuriki, Hoff 2021)

Theorem [invariant theorists Harm Derksen & Visu Makam, 2021]
Let $d := \gcd(m,n)$ and $r := (m^2+n^2-d^2)/(mn)$. The ML thresholds of the matrix normal model satisfy $\mathrm{mlt}_b = \mathrm{mlt}_e$, and

- If $m = n = 1$, then $\mathrm{mlt}_e = \mathrm{mlt}_u = 1$.
- If $m = n > 1$, then $\mathrm{mlt}_e = 1$ and $\mathrm{mlt}_u = 3$.
- If $m \neq n$ and $r \in \mathbb{Z}$, then $\mathrm{mlt}_e = r$.
  If $d = 1$, then $\mathrm{mlt}_u = r$, otherwise $\mathrm{mlt}_u = r+1$.
- If $m \neq n$ and $r \notin \mathbb{Z}$, then $\mathrm{mlt}_e = \mathrm{mlt}_u = \lceil (m^2+n^2)/(mn) \rceil$.

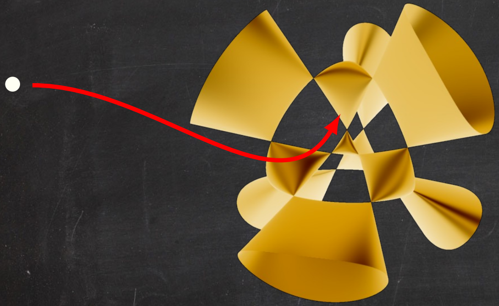


algebraic optimization

given •, find best point on (possibly unknown) manifold, variety, etc.

Examples:

- low-rank matrix approximation
- maximum likelihood estimation



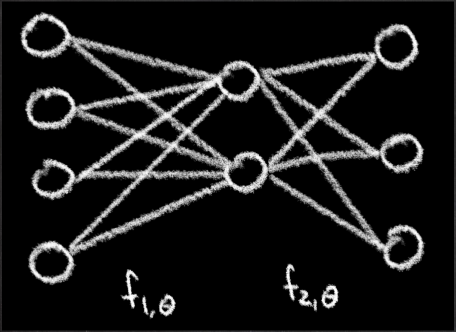
algebraic optimization

given •, find best point on (possibly unknown) manifold, variety, etc.

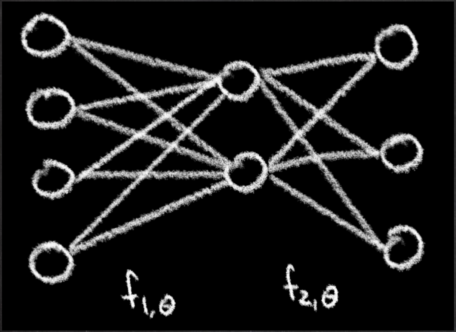
Examples:

- low-rank matrix approximation
- maximum likelihood estimation
- machine learning with neural networks

feedforward neural networks



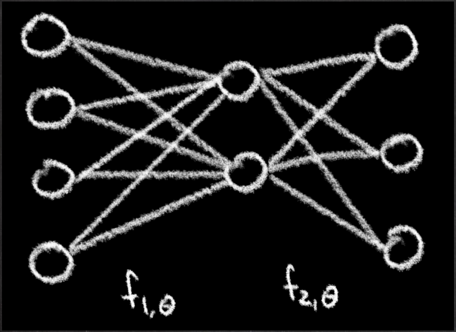
feedforward neural networks



are parametrized families of functions

$$\mu : \mathbb{R}^N \longrightarrow \mathcal{M},$$
$$\theta \longmapsto f_{L,\theta} \circ \ldots \circ f_{1,\theta}$$

feedforward neural networks

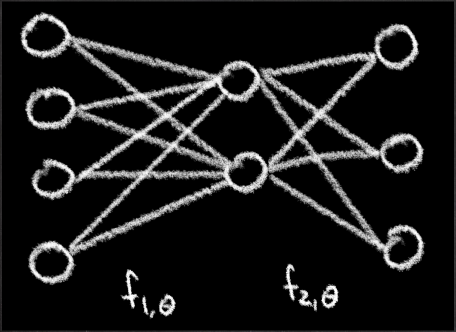


are parametrized families of functions

$$\mu : \mathbb{R}^N \longrightarrow \mathcal{M},$$
$$\theta \longmapsto f_{L,\theta} \circ \ldots \circ f_{1,\theta}$$

$L = \#$ layers, $f_{i,\theta} = (\sigma_i, \ldots, \sigma_i) \circ \alpha_{i,\theta}$,

feedforward neural networks

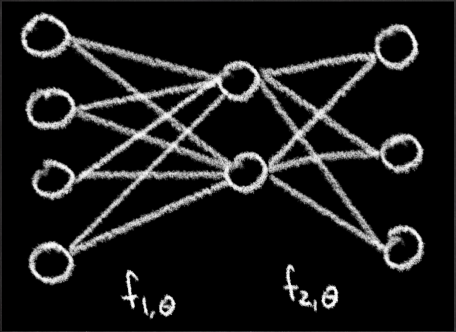


are parametrized families of functions

$$\mu : \mathbb{R}^N \longrightarrow \mathcal{M},$$
$$\theta \longmapsto f_{L,\theta} \circ \ldots \circ f_{1,\theta}$$

$L = \#$ layers, $f_{i,\theta} = (\sigma_i, \ldots, \sigma_i) \circ \alpha_{i,\theta}$,
$\sigma_i : \mathbb{R} \to \mathbb{R}$ activation, $\alpha_{i,\theta}$ affine linear

feedforward neural networks



$\mathcal{M} = \mathrm{im}(\mu) =$ neuromanifold

it is a manifold with boundary and singularities

are parametrized families of functions

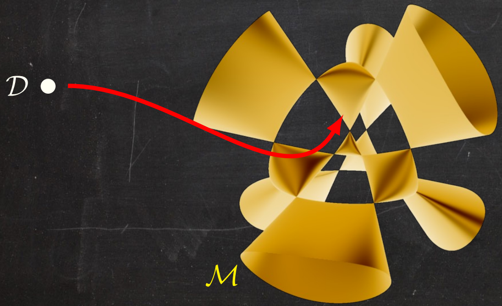
$$\mu : \mathbb{R}^N \longrightarrow \mathcal{M}, \quad \theta \longmapsto f_{L,\theta} \circ \ldots \circ f_{1,\theta}$$

$L = \#$ layers, $f_{i,\theta} = (\sigma_i, \ldots, \sigma_i) \circ \alpha_{i,\theta}$,
$\sigma_i : \mathbb{R} \to \mathbb{R}$ activation, $\alpha_{i,\theta}$ affine linear

training a network

Given training data $\mathcal{D}$, the goal is to minimize the **loss**

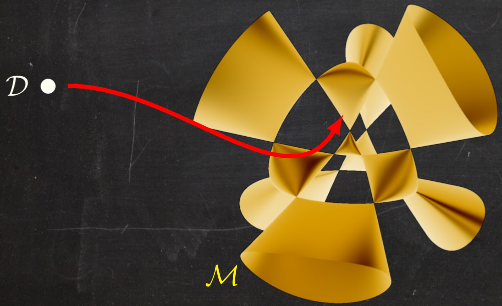
$$\mathbb{R}^N \xrightarrow{\mu} \mathcal{M} \xrightarrow{\ell_{\mathcal{D}}} \mathbb{R}.$$



training a network

Given training data $\mathcal{D}$, the goal is to minimize the **loss**

$$\mathbb{R}^N \xrightarrow{\mu} \mathcal{M} \xrightarrow{\ell_{\mathcal{D}}} \mathbb{R}.$$



Geometric questions:

- How does the network architecture affect the geometry of the function space?
- How does the geometry of the function space impact the training of the network?

understanding networks via algebraic optimization

understanding networks via algebraic optimization

Algebraic settings:

network architecture		
activation	**network structure**	**loss**

understanding networks via algebraic optimization

Algebraic settings:

network architecture		
activation	**network structure**	**loss**
identity		
ReLU		
polynomial		

understanding networks via algebraic optimization

Algebraic settings:

network architecture		
activation	**network structure**	**loss**
identity	fully-connected	
ReLU	convolutional	
polynomial	attention	

understanding networks via algebraic optimization

Algebraic settings:

network architecture			
activation	**network structure**	**loss**	
identity	fully-connected	squared-error loss	= Euclidean dist
ReLU	convolutional	Wasserstein distance	= polyhedral dist.
polynomial	attention	cross-entropy	≅ KL divergence

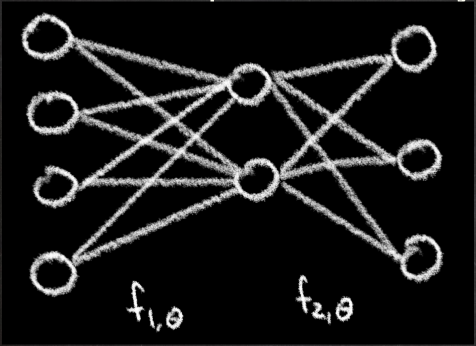
understanding networks via algebraic optimization

Algebraic settings:

network architecture			
activation	**network structure**	**loss**	
identity	fully-connected	squared-error loss	= Euclidean dist
ReLU	convolutional	Wasserstein distance	= polyhedral dist.
polynomial	attention	cross-entropy	≅ KL divergence

neuromanifold = semi-algebraic set defined by polynomial equalities
and inequalities

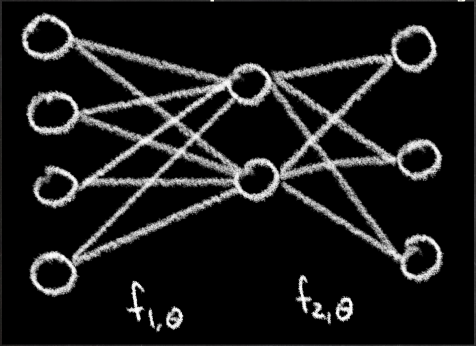
example: linear fully-connected networks



In this example:

$$\mu : \mathbb{R}^{2\times 4} \times \mathbb{R}^{3\times 2} \longrightarrow \mathbb{R}^{3\times 4},$$
$$(W_1, W_2) \longmapsto W_2 W_1.$$

example: linear fully-connected networks

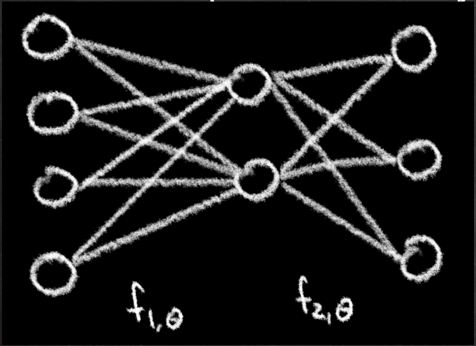


In this example:

$$\mu : \mathbb{R}^{2\times 4} \times \mathbb{R}^{3\times 2} \longrightarrow \mathbb{R}^{3\times 4},$$
$$(W_1, W_2) \longmapsto W_2 W_1.$$

$$\mathcal{M} = \{W \in \mathbb{R}^{3\times 4} \mid \mathrm{rank}(W) \leq 2\}$$

example: linear fully-connected networks



In this example:

$$\mu : \mathbb{R}^{2\times 4} \times \mathbb{R}^{3\times 2} \longrightarrow \mathbb{R}^{3\times 4},$$
$$(W_1, W_2) \longmapsto W_2 W_1.$$

$$\mathcal{M} = \{W \in \mathbb{R}^{3\times 4} \mid \mathrm{rank}(W) \leq 2\}$$

In general:

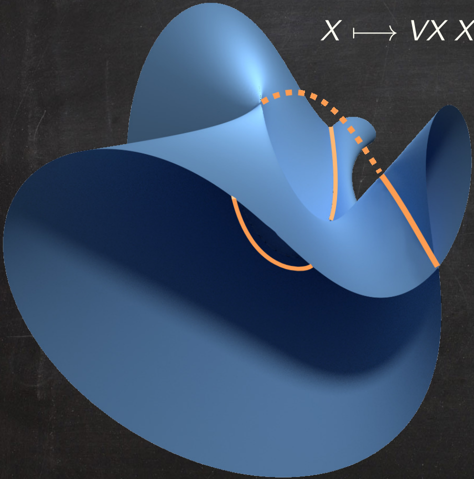
$$\mu : \mathbb{R}^{k_1\times k_0} \times \mathbb{R}^{k_2\times k_1} \times \ldots \times \mathbb{R}^{k_L\times k_{L-1}} \longrightarrow \mathbb{R}^{k_L\times k_0},$$
$$(W_1, W_2, \ldots, W_L) \longmapsto W_L \cdots W_2 W_1.$$

$\mathcal{M} = \{W \in \mathbb{R}^{k_L\times k_0} \mid \mathrm{rank}(W) \leq \min(k_0, \ldots, k_L)\}$ is an algebraic variety and we know its singularities etc.

example: attention networks

A single-layer lightning self-attention network with weights $Q, K \in \mathbb{R}^{a\times d}$ and $V \in \mathbb{R}^{d'\times d}$ is

$$\mathbb{R}^{d\times t} \longrightarrow \mathbb{R}^{d'\times t},$$
$$X \longmapsto VX\,X^\top K^\top QX.$$



A slice of the 5-dimensional neuromanifold $\mathcal{M}$ for $a = d = t = 2, d' = 1$.

It is singular along the orange curve, and has boundary points where the curve leaves/enters $\mathcal{M}$.

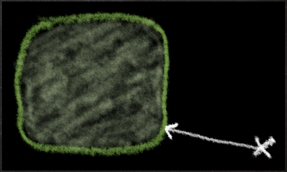
understanding networks via algebraic optimization

Algebraic settings:

network architecture			
activation	**network structure**	**loss**	
identity	fully-connected	squared-error loss	= Euclidean dist
ReLU	convolutional	Wasserstein distance	= polyhedral dist.
polynomial	attention	cross-entropy	$\cong$ KL divergence

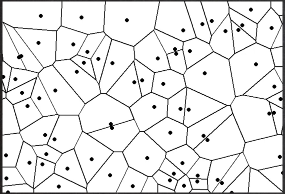
neuromanifold = semi-algebraic set

its boundaries and singularities can be especially exposed during training



Voronoi cells

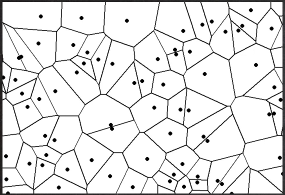
Given a set $\mathcal{M} \subseteq \mathbb{R}^n$, the Voronoi cell of $x \in \mathcal{M}$ consists of all $u \in \mathbb{R}^n$ such that x is "closest" among all points in $\mathcal{M}$.



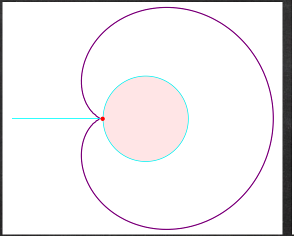
$\mathcal{M}$ might be finite

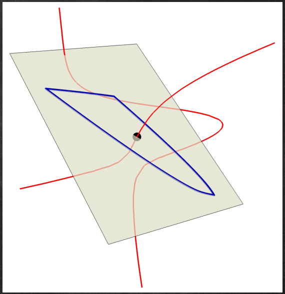
Voronoi cells

Given a set $\mathcal{M} \subseteq \mathbb{R}^n$, the Voronoi cell of $x \in \mathcal{M}$ consists of all $u \in \mathbb{R}^n$ such that x is "closest" among all points in $\mathcal{M}$.



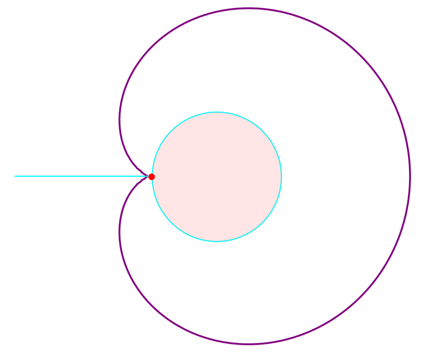
$\mathcal{M}$ might be finite





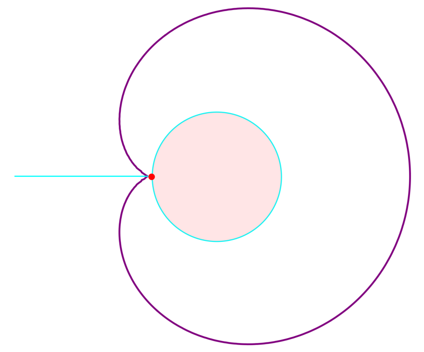
or a manifold, variety, semi-algebraic set, etc.

Voronoi cells with respect to Euclidean distance



$\mathcal{M} \subseteq \mathbb{R}^2$ is the purple curve

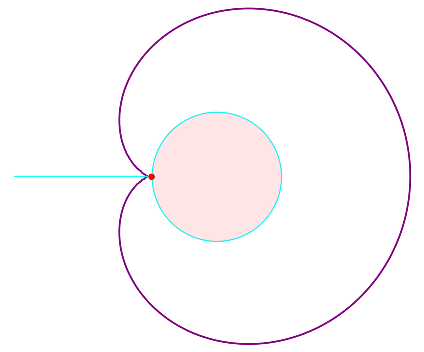
Voronoi cells with respect to Euclidean distance



$\mathcal{M} \subseteq \mathbb{R}^2$ is the purple curve

at all smooth points $x \in \mathcal{M}$, the Voronoi cell is a line segment

Voronoi cells with respect to Euclidean distance

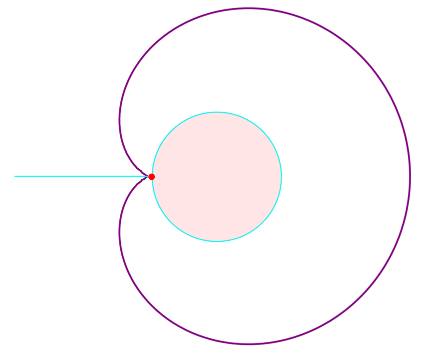


$\mathcal{M} \subseteq \mathbb{R}^2$ is the purple curve

at all smooth points $x \in \mathcal{M}$, the Voronoi cell is a line segment

the Voronoi cell at the singularity is 2-dimensional, i.e., that point is the closest with positive probability

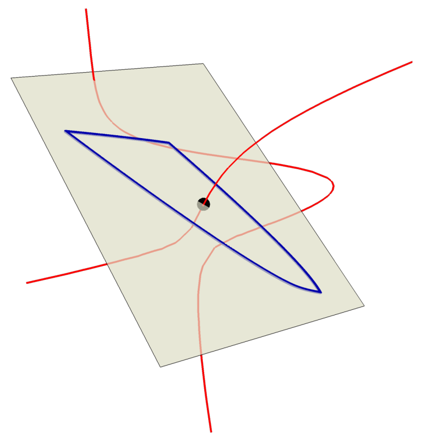
Voronoi cells with respect to Euclidean distance



$\mathcal{M} \subseteq \mathbb{R}^2$ is the purple curve

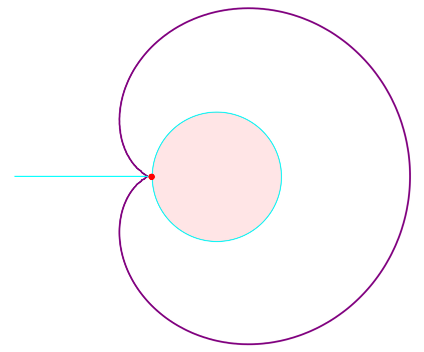
at all smooth points $x \in \mathcal{M}$, the Voronoi cell is a line segment

the Voronoi cell at the singularity is 2-dimensional, i.e., that point is the closest with positive probability



$\mathcal{M} \subseteq \mathbb{R}^3$ is the red curve

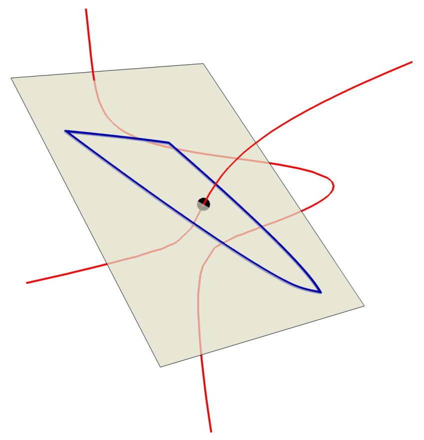
Voronoi cells with respect to Euclidean distance



$\mathcal{M} \subseteq \mathbb{R}^2$ is the purple curve

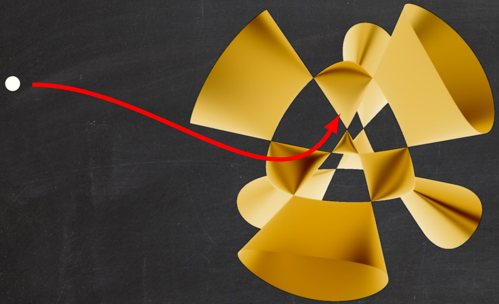
at all smooth points $x \in \mathcal{M}$, the Voronoi cell is a line segment

the Voronoi cell at the singularity is 2-dimensional, i.e., that point is the closest with positive probability



$\mathcal{M} \subseteq \mathbb{R}^3$ is the red curve

at smooth points, the Voronoi cell is a convex, semi-algebraic, 2-dimensional subset of the normal plane

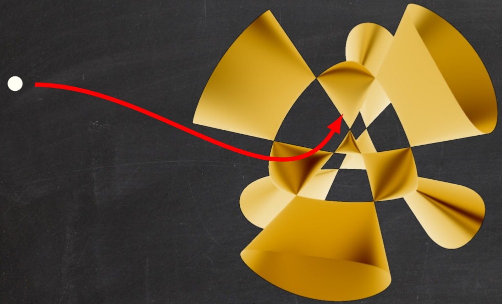


Examples:

- low-rank matrix approximation
- maximum likelihood estimation
- machine learning with neural networks

algebraic optimization
given •, find best point on (possibly unknown) manifold, variety, etc.

Often, the manifold / semialgebraic set is unknown or hard to understand!



algebraic optimization

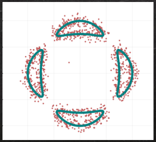
given •, find best point on (possibly unknown) manifold, variety, etc.

Examples:

- low-rank matrix approximation
- maximum likelihood estimation
- machine learning with neural networks

Often, the manifold / semialgebraic set is unknown or hard to understand!

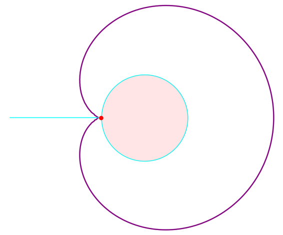
Can we learn something from samples?



medial axis & reach

$\mathcal{M} \subseteq \mathbb{R}^n$

The union of the boundaries of all Voronoi cells is the **medial axis** of $\mathcal{M}$.

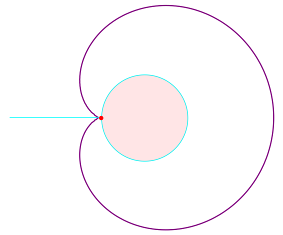


medial axis & reach

$\mathcal{M} \subseteq \mathbb{R}^n$

The union of the boundaries of all Voronoi cells is the **medial axis** of $\mathcal{M}$.

It consists of all points in $\mathbb{R}^n$ that have two "closest" points on $\mathcal{M}$.

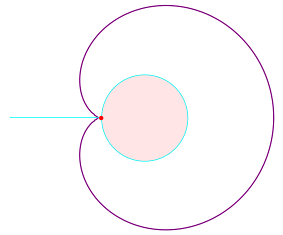


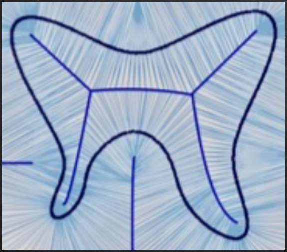
medial axis & reach

$\mathcal{M} \subseteq \mathbb{R}^n$

The union of the boundaries of all Voronoi cells is the **medial axis** of $\mathcal{M}$.

It consists of all points in $\mathbb{R}^n$ that have two "closest" points on $\mathcal{M}$.





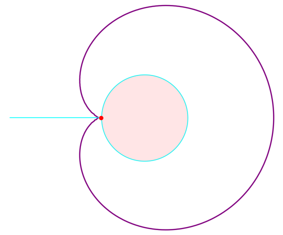
If $\mathcal{M}$ is a smooth variety, its medial axis with respect to Euclidean distance has positive distance from $\mathcal{M}$.

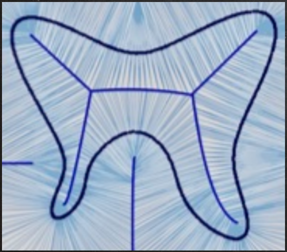
medial axis & reach

$\mathcal{M} \subseteq \mathbb{R}^n$

The union of the boundaries of all Voronoi cells is the **medial axis** of $\mathcal{M}$.

It consists of all points in $\mathbb{R}^n$ that have two "closest" points on $\mathcal{M}$.





If $\mathcal{M}$ is a smooth variety, its medial axis with respect to Euclidean distance has positive distance from $\mathcal{M}$.

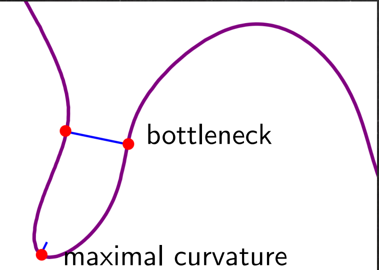
This distance is the **reach** of $\mathcal{M}$.

$\mathcal{M} \subseteq \mathbb{R}^n$ smooth variety

$$\Rightarrow \mathrm{reach}(\mathcal{M}) = \text{min}\left\{\text{smallest bottleneck width}, \frac{1}{\text{maximal curvature}}\right\}$$

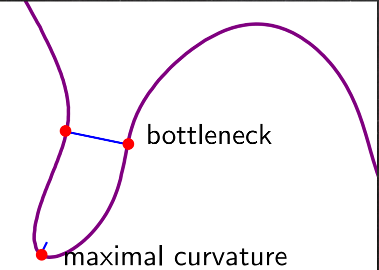
$\mathcal{M} \subseteq \mathbb{R}^n$ smooth variety

$$\Rightarrow \text{reach}(\mathcal{M}) = \min \left\{ \text{smallest bottleneck width}, \frac{1}{\text{maximal curvature}} \right\}$$



$\mathcal{M} \subseteq \mathbb{R}^n$ smooth variety

$$\Rightarrow \mathrm{reach}(\mathcal{M}) = \min \left\{ \text{smallest bottleneck width}, \frac{1}{\text{maximal curvature}} \right\}$$

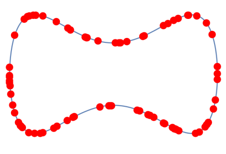


$\{x, y\} \subset \mathcal{M}$ is a bottleneck if $x - y$ is normal to both tangent spaces $T_x\mathcal{M}$ and $T_y\mathcal{M}$

its width is $\frac{1}{2}\|x - y\|_2$

reach & sampling

$\mathcal{M} \subseteq \mathbb{R}^n$ smooth variety, $S \subseteq \mathcal{M}$ finite sample, $0 < \varepsilon < \sqrt{\frac{3}{20}}\, \mathrm{reach}(\mathcal{M})$

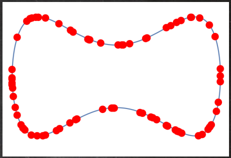


For all $x \in \mathcal{M}$, there is $s \in S$ with $\|x - s\|_2 < \varepsilon$

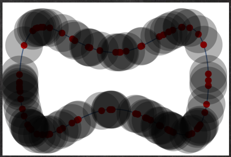
reach & sampling

$\mathcal{M} \subseteq \mathbb{R}^n$ smooth variety, $S \subseteq \mathcal{M}$ finite sample, $0 < \varepsilon < \sqrt{\frac{3}{20}}\,\mathrm{reach}(\mathcal{M})$

For all $x \in \mathcal{M}$, there is $s \in S$ with $\|x - s\|_2 < \varepsilon$



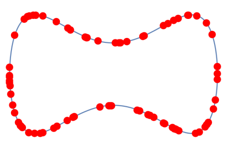
$U =$ union of all ε-balls around all points in S



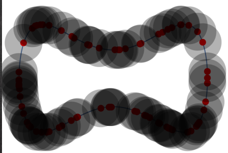
reach & sampling

$\mathcal{M} \subseteq \mathbb{R}^n$ smooth variety, $S \subseteq \mathcal{M}$ finite sample, $0 < \varepsilon < \sqrt{\frac{3}{20}}\,\mathrm{reach}(\mathcal{M})$

For all $x \in \mathcal{M}$, there is $s \in S$ with $\|x - s\|_2 < \varepsilon$



$U =$ union of all ε-balls around all points in S



Theorem [Niyogi, Smale, Weinberger]
$\mathcal{M}$ is a deformation retract of U.

They have the same homology!

Homology of U is computable from the associated Čech complex

How to actually solve
algebraic inverse problems
?

2d pictures

given observations, want
to recover ground truth

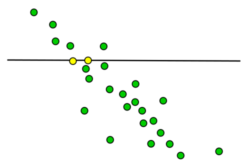
3d modell

Observations are often noisy, and can even be corrupted with outliers.
RANSAC (RANdom SAmple Consensus) provides robust estimation !

Observations are often noisy, and can even be corrupted with outliers.
RANSAC (RANdom SAmple Consensus) provides robust estimation !

1) Randomly select a subset of the data
2) Fit a model to the selected subset
3) Determine the number of outliers
4) Repeat steps 1-3 to find a consensus (& outliers)

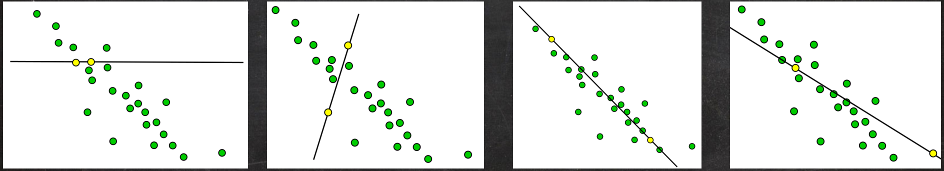
Example: fitting a line to points



Observations are often noisy, and can even be corrupted with outliers.
RANSAC (RANdom SAmple Consensus) provides robust estimation !

1) Randomly select a subset of the data
2) Fit a model to the selected subset
3) Determine the number of outliers
4) Repeat steps 1-3 to find a consensus (& outliers)

Example: fitting a line to points



few outliers!

Observations are often noisy, and can even be corrupted with outliers. RANSAC (RANdom SAmple Consensus) provides robust estimation !

1) Randomly select a subset of the data
2) Fit a model to the selected subset
3) Determine the number of outliers
4) Repeat steps 1-3 to find a consensus (& outliers)

2d pictures

$\longrightarrow$

3d modell

for general algebraic inverse problems, step 2) means to solve a system of polynomial equations!

Observations are often noisy, and can even be corrupted with outliers.
RANSAC (RANdom SAmple Consensus) provides robust estimation !

1) Randomly select a subset of the data
2) Fit a model to the selected subset
3) Determine the number of outliers
4) Repeat steps 1-3 to find a consensus (& outliers)

2d pictures

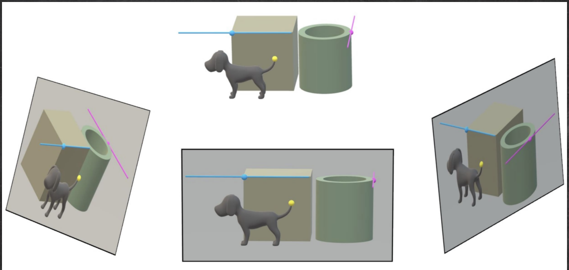
$\longrightarrow$

3d modell

for general algebraic inverse problems, step 2) means to solve a system of polynomial equations!

need to do this very fast! (due to step 4)) can solve polynomial systems via Gröbner bases can solve polynomial systems via Gröbner bases or homotopy continuation

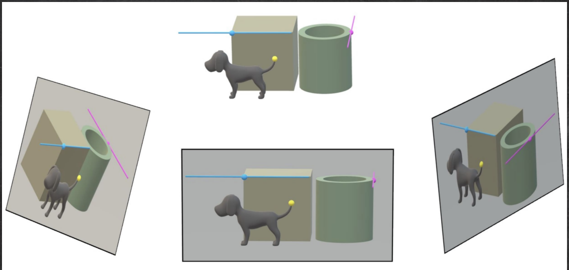
example: 3d reconstruction from unknown cameras



example: 3d reconstruction from unknown cameras

Given: point, point on line & point on line on each 2d-image

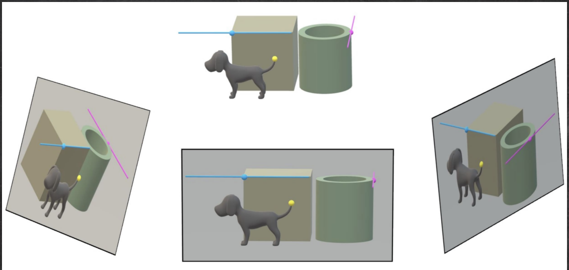
Goal: compute point, point on line & point on line in 3-space, and positions $c_1, c_2, c_3 \in \mathbb{R}^3$ & orientations $R_1, R_2, R_3 \in \mathrm{SO}(3)$ of cameras



example: 3d reconstruction from unknown cameras

Given: point, point on line & point on line on each 2d-image

Goal: compute point, point on line & point on line in 3-space, and positions $c_1, c_2, c_3 \in \mathbb{R}^3$ & orientations $R_1, R_2, R_3 \in \mathrm{SO}(3)$ of cameras

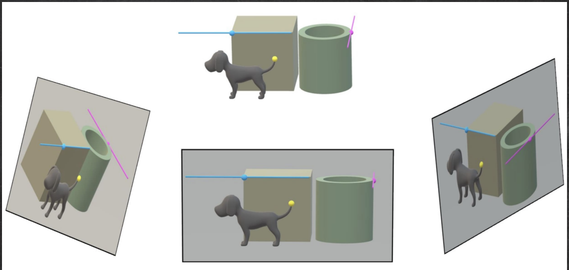


Generally has 312 complex solutions (modulo the appropriate group action).

example: 3d reconstruction from unknown cameras

Given: point, point on line & point on line on each 2d-image

Goal: compute point, point on line & point on line in 3-space, and positions $c_1, c_2, c_3 \in \mathbb{R}^3$ & orientations $R_1, R_2, R_3 \in \mathrm{SO}(3)$ of cameras



Generally has 312 complex solutions (modulo the appropriate group action).

Gröbner basis methods won't terminate ...

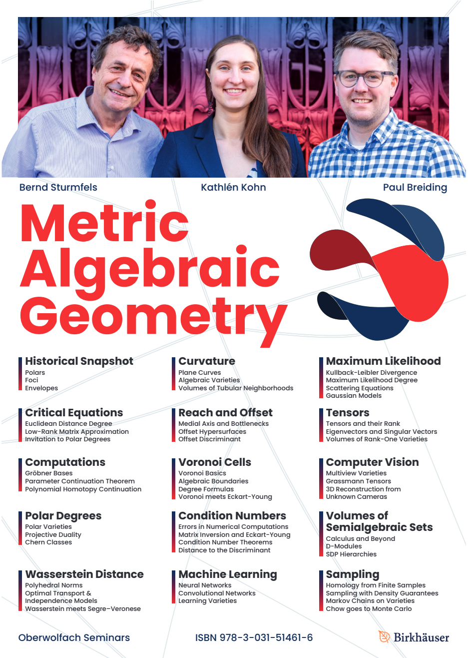
Homotopy continuation can solve in 660ms on average on Intel core i7-7920HQ processor with 4 threads Fabbri et. al.: TRPLP – Trifocal Relative Pose from Lines at Points, CVPR 2020

Data science requires us to rethink the schism between mathematical disciplines!

differential geometry ⇒

algebraic geometry ⇒

data science ⇒



open access :)