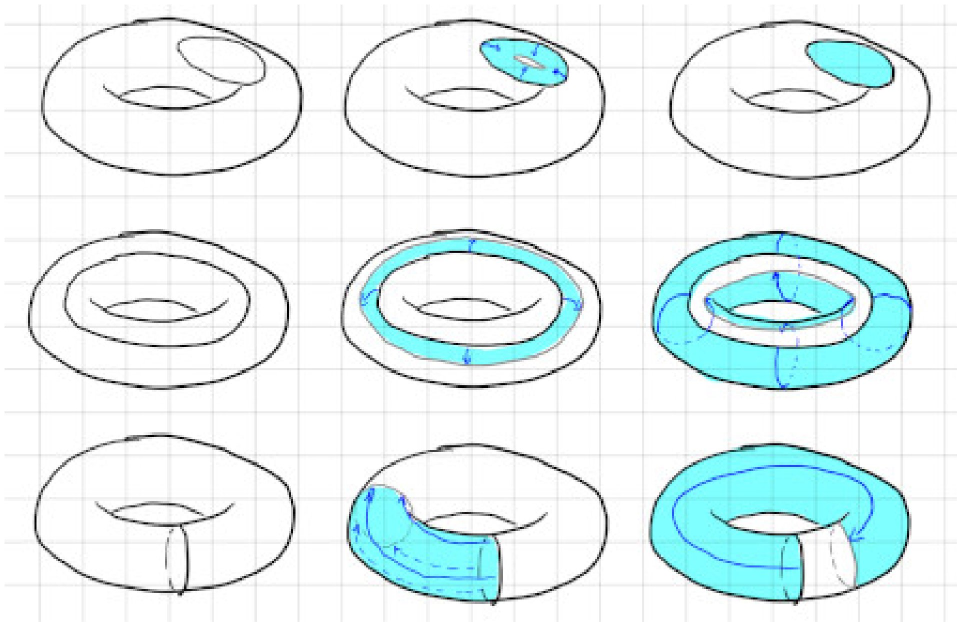
Algebraic Concepts in Machine Learning and Signal Processing



Pavel Loskot
pavelloskot@intl.zju.edu.cn

ZJU-UIUC INSTITUTE
Zhejiang University-University of Illinois at Urbana-Champaign Institute
浙江大学伊利诺伊大学厄巴纳香槟校区联合学院

SIGNAL 2023: ***The Eight International Conference on Advances in Signal, Image and Video Processing***

March 13–17, 2023, Barcelona, Spain

IARIA

About Me

Pavel Loskot joined the ZJU-UIUC Institute as Associate Professor in January 2021. He received his PhD degree in Wireless Communications from the University of Alberta in Canada, and the MSc and BSc degrees in Radioelectronics and Biomedical Electronics, respectively, from the Czech Technical University of Prague. He is the Senior Member of the IEEE, Fellow of the HEA in the UK, and the Recognized Research Supervisor of the UKCGE.

In the past 25 years, he was involved in numerous industrial and academic collaborative projects in the Czech Republic, Finland, Canada, the UK, Turkey, and China. These projects concerned mainly wireless and optical telecommunication networks, but also genetic regulatory circuits, air transport services, and renewable energy systems. This experience allowed him to truly understand the interdisciplinary workings, and crossing the disciplines boundaries.

His current research focuses on statistical signal processing and importing methods from Telecommunication Engineering and Computer Science to model and analyze systems more efficiently and with greater information power.

Objectives

1. Moving beyond calculus

- explore algebraic structures which can be used in signal processing and machine learning

2. A starting point for new researchers in this area

- a tutorial which identifies the key concepts and terminology to focus on

Outline

1. Basic Algebraic Concepts
2. Algebraic Topology
3. Topological Data Analysis
4. Conclusions

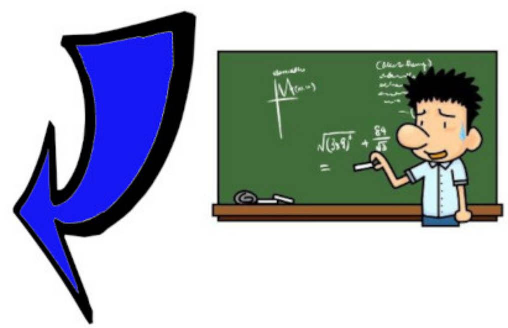
Mathematics-Engineering Gap

Engineering

- pragmatic, design oriented
- things driven
- complexity becoming an issue
- increase use of math models
- increase use of abstractions
- large pool of engineers

Mathematics

- pure vs. applied, but always rigorous
- concepts driven
- study of abstractions
- specialized skills/knowledge
- favorite areas: bio-med, finance
- small pools of mathematicians



Opportunity

- adopt common/advanced math concepts for “easy” use in engineering
- go beyond calculus and numerical computations
- allow working with advanced math objects, structures and models

Common Tools in Computational Engineering

Data processing

- data modeling
- statistical inference
- causal inference

Numerical solvers

- system observations and control
- optimizations
- finite element method

Machine learning

- regression, classification
- clustering, labeling
- prediction, prescription

Analysis

- calculus, real analysis
- little bit of algebra
- social network analysis
- sensitivity analysis
- Bayesian analysis

Mathematical models

- vectors, matrices
- functions
- tensors and graphs
- random variables and processes

Digitization and virtualization

- tools for handling abstractions (algebras, logic)
- work with advanced math objects (manifolds, functors)

Part 1:
Basic Algebraic Concepts

ALGEBRAS

Algebra defines laws of computations for numbers (number systems).

Abstract Algebra manipulates algebraic (e.g. numeric and geometric) objects.

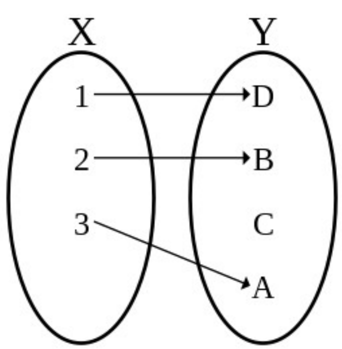
Arithmetic provides rules to calculate numerical expressions.

Sets notation

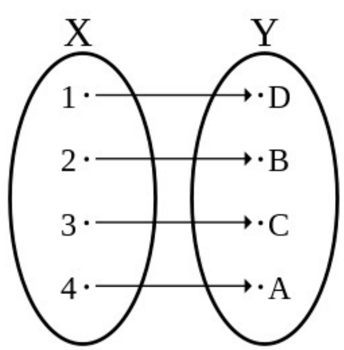
$A = \{a, b, \ldots\}$	set	$\emptyset$	empty set
$A \cup B$	union	$A \cap B$	intersection
$A \subseteq B$	subset	$A \subset B$	proper subset
$A \supseteq B$	superset	$A \supset B$	proper superset
$A \setminus B$	set difference	$\mathcal{P}(A)$	powerset (set of all subsets)

Maps

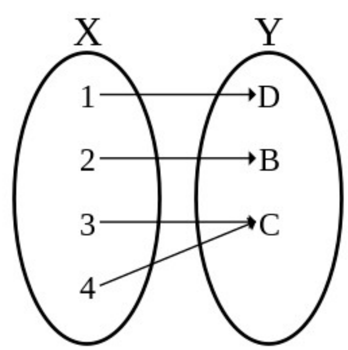
injection
(injective & non-surjective)



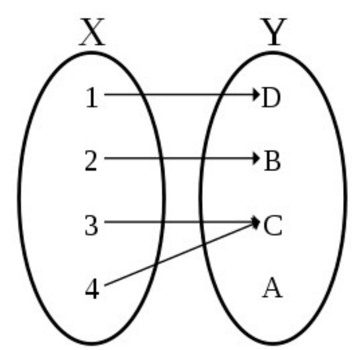
bijection
(injective & surjective)



surjection
(non-injective & surjective)



–
(non-injective & non-surjective)



Algebras (cont.)

Semi-group $(S,+)$

- closed and associative w.r.t. operator '+'

Monoid $(S,+)$

- a semi-group with neutral element, i.e., $a+z=a\ \forall a\in S$

Group $(S,+)$

- a monoid with inverse element, i.e., $a+\bar{a}=z\ \forall a\in S$

Abelian group $(S,+)$

- commutative w.r.t. operator '+', i.e., $a+b=b+a$

Ring $(S,+,*)$

- $(S,+)$ is commutative group and $(S,*)$ is semi-group
- distributive, i.e., $a*(b+c)=a*b+a*c\ \forall a,b,c\in S$

Field $(S,+,*)$

- ring with $(S\setminus\{0\},*)$ being a group

Algebras (cont.)

Examples

- $(\mathbb{Z}, -)$ is not semi-group (not associative)
- $(\mathbb{N}, +)$ is semi-group (not group, since no 0)
- $(\mathbb{N}_0, +)$ is monoid (no inverse element)
- $(\mathbb{Z}, *)$ is monoid (no inverse element)
- $(\mathbb{Z}, +)$ is group
- $(\mathbb{Z}_n, +)$ is group
- $(\mathbb{Z}_n, *)$ is monoid (no inverse element for 0)
- $(\mathbb{Z}_n \setminus \{0\}, *)$ is group if n is prime
- $(\mathbb{Z}, +, *)$ is ring (not field)
- $(\mathbb{Z}_n, +, *)$ is finite ring and finite field if n is prime

Other topics

- polynomial arithmetic
- universal algebra
- algebraic structures (sets, vectors, graphs)
- set and graph theory
- category theory
- knot theory
- algebraic geometry
- topology
- algebraic topology (persistent homology, homotopy, complexes)

Graphs

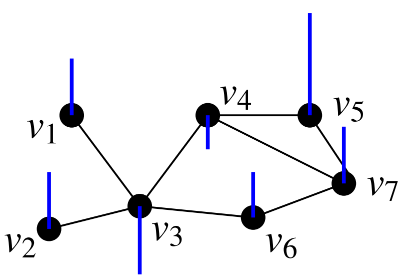
Graph	Network	System
vertex	node	component
edge	link	interaction

Mathematics

- static objects: graph theory, graph topology, social network analysis
- enumerating and constructing graphs/sub-graphs/paths/motifs/clusters
- properties: centrality, distributions, associativity, modularity, distance

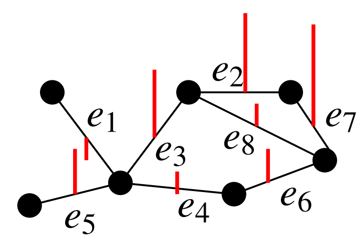
Engineering

- graph computing, graph signal processing, knowledge graphs
- dynamic objects: data structures and models of networked systems
- properties: robustness, max flow
- routing, searching, navigation, epidemic spreading, info cascades



$f\colon V \mapsto \mathcal{R}^{|V|}$

Node	Flows	Flow balance
sink	absorbed	inflows > outflows
source	generated flows	inflows < outflows
router	mix and split	inflows = outflows



$f\colon E \mapsto \mathcal{R}^{|E|}$

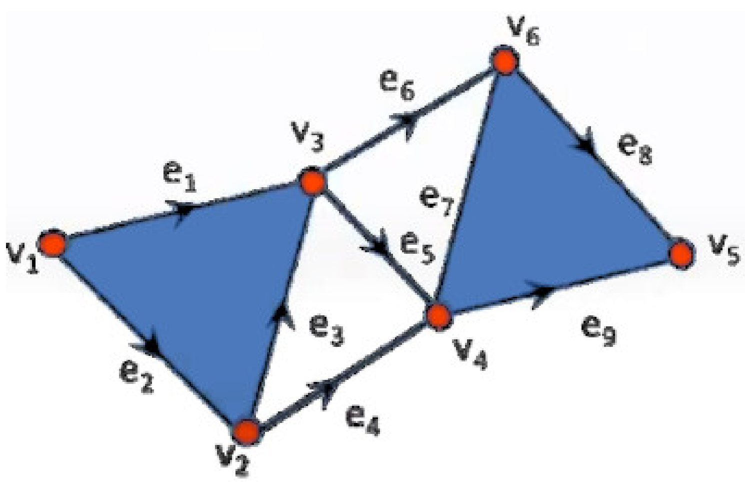
Graph Simplexes

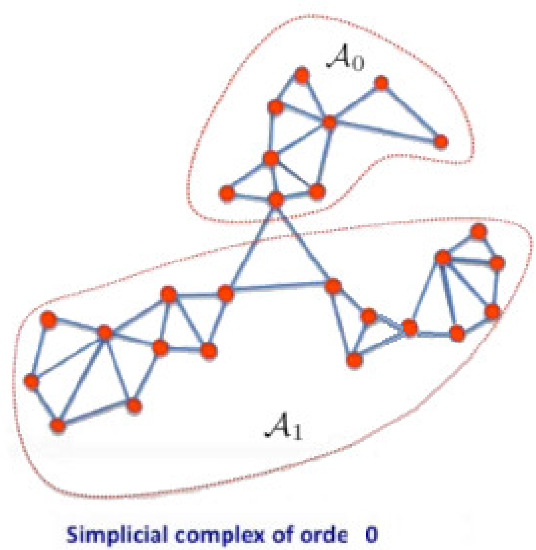
k-simplex

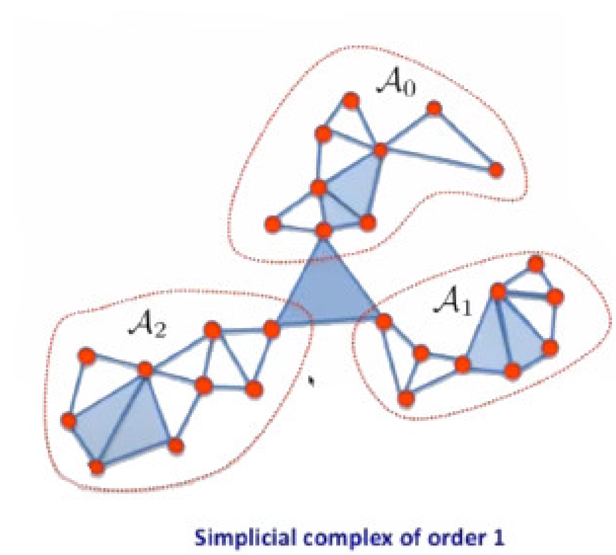
- a closed-path object with k edges and $k+1$ nodes
  → 0-simplex is a node
  → 1-simplex is an edge (pairwise relations or flows)
  → 2-simplex is a open/closed triangle (triple-wise relations)

Graph cuts

- graph partitioned into simplical complexes of order k







Data processing

- graphs are visual representations of k-order relationships among data

Graph as a Matrix

Complete graph representation

- adjacency matrix and incidence matrix
- Laplacian and degree matrix
- linear model: $\boldsymbol{X}_{t+1} = \boldsymbol{A}\boldsymbol{X}_t + \boldsymbol{u}_t$

Graph Approximation of a N-Dimensional Function

Sobol's expansion (deterministic function)

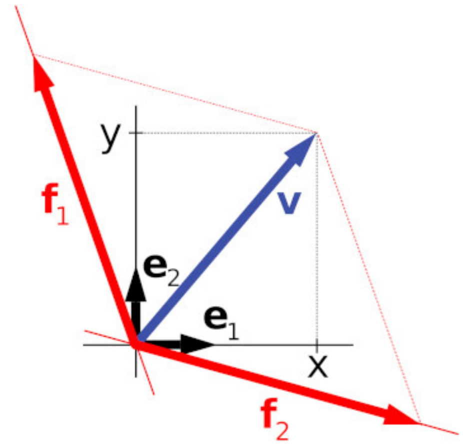
$$Y = f(X_1, X_2, \ldots, X_N) = f_0 + \sum_i \underbrace{f_i(X_i)}_{\text{nodes}} + \sum_{i<j} \underbrace{f_{ij}(X_i, X_j)}_{\text{edges}}$$

$$+ \sum_{i<j} \underbrace{f_{i<j<l}(X_i, X_j, X_l)}_{\text{triangles}} + \cdots + \sum_{\text{except } i} f_{12\cdots N-1}(X_1, \ldots, X_{N-1})$$

Variance expansion (stochastic function)

$$V(Y) = \sum_i \underbrace{V_i}_{\text{nodes}} + \sum_{i<j} \underbrace{V_{ij}}_{\text{edges}} + \sum_{i<j<l} \underbrace{V_{ijl}}_{\text{triangles}} + \cdots + V_{12\ldots N}$$

Tensors



Multi-dimensional arrays?

- yes, but only one (narrow) interpretation

Geometric vectors?

- magnitude & direction remain the same in different bases (frames of reference)
- rank 1 tensor, contravariant vector

Key properties

- tensor can be represented as ordered list of numbers (vector) in given basis
- object represented by a tensor does not change in different bases
  $\rightarrow$ not every representation is a tensor
- tensor rank (order, degree) is dimension of the object it represents

Contravariant vector $(1,0)$-tensor

- basis are columns of $\boldsymbol{B}$, so $\boldsymbol{v} = \boldsymbol{B} \cdot \tilde{\boldsymbol{v}}$
- basis rotation & scaling via $\boldsymbol{T}$

$$\boldsymbol{v} = \underbrace{\boldsymbol{BT}}_{\text{basis}} \cdot \underbrace{\boldsymbol{T}^{-1}\tilde{\boldsymbol{v}}}_{\text{components}}$$

Covariant vector (covector) $(0,1)$-tensor

- co-varies with basis transformation
- it is a linear function $\langle \boldsymbol{v}, \boldsymbol{x} \rangle$
- value $\langle \boldsymbol{v}, \boldsymbol{x} \rangle$ is independent of basis

Tensors (cont.)

Linear transformation $(1,1)$-tensor

- change of basis: $\tilde{\boldsymbol{y}} = \boldsymbol{T}\boldsymbol{y}$ and $\tilde{\boldsymbol{x}} = \boldsymbol{T}\boldsymbol{x}$
- if $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, then $\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{A}}\tilde{\boldsymbol{x}}$ where $\tilde{\boldsymbol{A}} = \boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1}$
  $\rightarrow$ $\boldsymbol{T}^{-1}$ is contravariant
  $\rightarrow$ $\boldsymbol{T}$ is covariant
  $\rightarrow$ $\boldsymbol{T}\boldsymbol{A}\boldsymbol{T}^{-1}$ is $(1,1)$-tensor, i.e., rank 2 tensor (2×2 matrix)

Bi-linear form $B:\ \boldsymbol{u},\boldsymbol{v} \mapsto \mathbb{R}$

$$\begin{aligned} B(\boldsymbol{u}+\boldsymbol{w},\boldsymbol{v}) &= B(\boldsymbol{u},\boldsymbol{v}) + B(\boldsymbol{w},\boldsymbol{v}) \\ B(\lambda\boldsymbol{u},\boldsymbol{v}) &= \lambda B(\boldsymbol{u},\boldsymbol{v}) \\ B(\boldsymbol{u},\boldsymbol{v}+\boldsymbol{w}) &= B(\boldsymbol{u},\boldsymbol{v}) + B(\boldsymbol{u},\boldsymbol{w}) \\ B(\boldsymbol{u},\lambda\boldsymbol{v}) &= \lambda B(\boldsymbol{u},\boldsymbol{v}) \end{aligned} \quad \Rightarrow \quad B(\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{u}^T\boldsymbol{A}\boldsymbol{v} = \sum_{i,j=1}^{n} a_{i,j}u_i v_j = A_{ij}u^i v^j$$

- $\boldsymbol{A}$ is rank $(0,2)$-tensor (with two covectors)
- $\boldsymbol{u}$ and $\boldsymbol{v}$ are $(1,0)$-tensors (contravariants)
- with transform of basis $\boldsymbol{T}$, $\tilde{A}_{ij} = A_{ij}T^i_k T^j_l$

Tensors (cont.)

Metric tensor $(0,2)$-tensor

- $g_p(\boldsymbol{x}_p, \boldsymbol{y}_p) \in \mathbb{R}$, $\boldsymbol{x}_p$ and $\boldsymbol{y}_p$ are tangent vectors
  $\rightarrow$ g_p is bi-linear function
  $\rightarrow$ $g_p(\boldsymbol{x}_p, \boldsymbol{y}_p) = g_p(\boldsymbol{y}_p, \boldsymbol{x}_p)$ (symmetry)
  $\rightarrow$ $g_p \neq 0$ for $\boldsymbol{x}_p \neq 0$ and some $\boldsymbol{y}_p$ (non-degeneracy)
- can be use to define basis-independent vector metrics
  $\rightarrow$ g is a dot product of vectors

$$g(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u} \cdot \boldsymbol{v} = g_{ij} u^i v^j = \boldsymbol{u}^T \boldsymbol{I} \boldsymbol{v} \quad \Rightarrow \quad g \equiv \boldsymbol{I} \text{ (identity matrix)}$$

- basis-independent magnitude, distance, and angle
  $\rightarrow$ they do not change with liner transformation of basis
  $\rightarrow$ can involve integration (area)

$$||\boldsymbol{u}|| = \sqrt{g_{ij} u^i u^j}, \quad ||\boldsymbol{u} - \boldsymbol{v}|| = \sqrt{g_{ij} (u - v)^i (u - v)^j}, \quad \cos(\theta) = \frac{g_{ij} u^i v^j}{||\boldsymbol{u}|| \, ||\boldsymbol{v}||}$$

Summary

- tensor transforms input tensor (or none) into output tensor with basis-invariant properties
- (n, m)-tensor has n contravariant and m covariant components
- rank $(n + m)$ is the total number of components (axis)

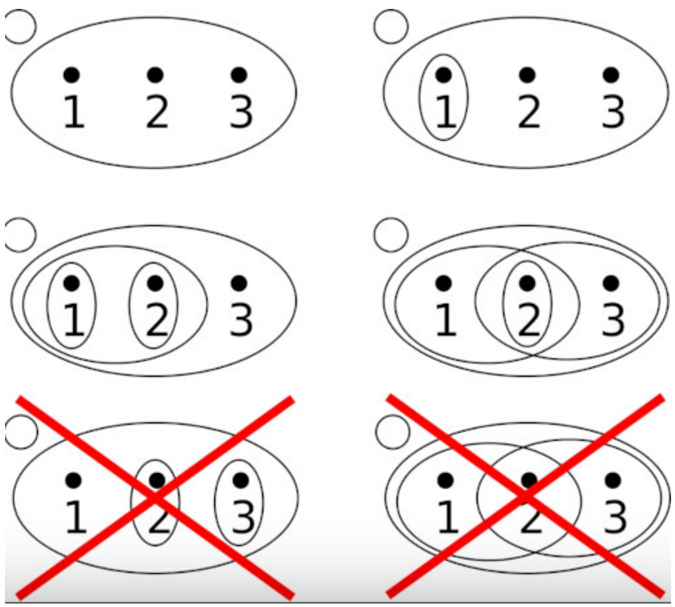
Part 2:

Algebraic Topology

Topology

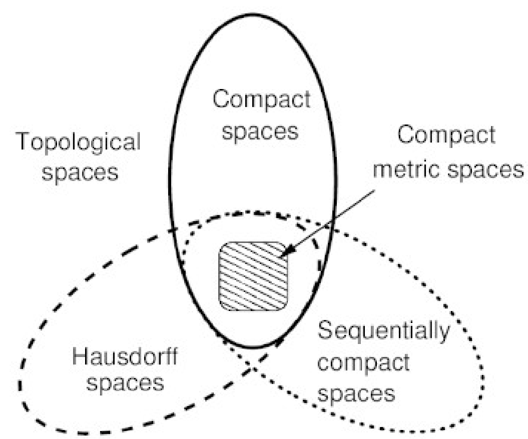
Topological space

- open set of points having certain topology
  → generalization of 1D open interval
  → points can be any mathematical structure
  → exact definition is axiom based
- allows for defining
  → closeness of points
  → neighborhoods (subsets)
  → limits, continuity, connectedness
  → distance may be undefined
- Euclidean space, Hilbert space
- metric space, manifolds



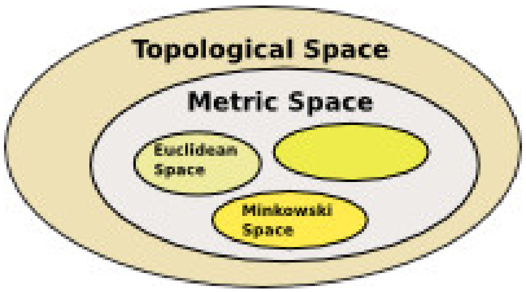
Constructs involving topology

- multiple topologies over subsets of points
- maps between topological spaces
  → allow defining associations
- category of topology/topological space
  → category theory, K-theory
  → homotopy and homology theory



Topological Spaces

Metric space



- add a notion of distance between points in a set
  → physical, angular, between states, invariants
- any mathematical objects with a distance
  → manifolds, graphs, normed space, length space

Map $f: M_1 \mapsto M_2$ between (M_1, d_1) and (M_2, d_2)

- isometry: distance preserving
- quasi-isometry: preserves large-scale topology
- Lipschitz map: stretch/contract distances
- homeomorphism: continuous bijection whose inverse is also continuous
  → define topological equivalences

Graphs

- discrete topology
- combinatorial problems
- embedding in metric spaces
  → machine learning

Key considerations

- topology of metric space
- distance between points/objects
- functions between and to metric spaces
- construction of topological/vector spaces
- generalizations

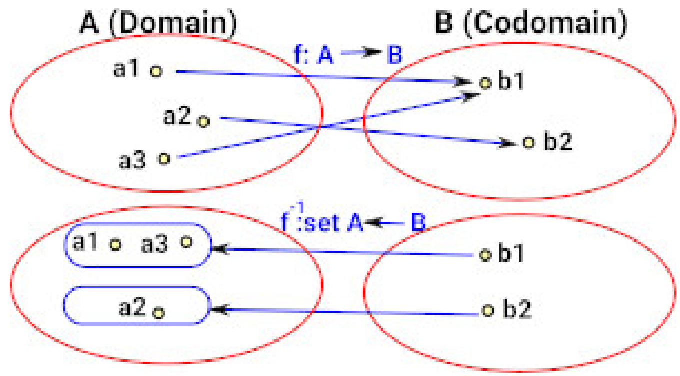
FUNCTION TYPES

Inverse functions

- inverse of addition: $\mathcal{N} \mapsto \mathcal{Z}$
- inverse of multiplication: $\mathcal{N} \mapsto \mathcal{Q} \subset \mathbb{R}$
- bijective map: isomorphism and homeomorphism

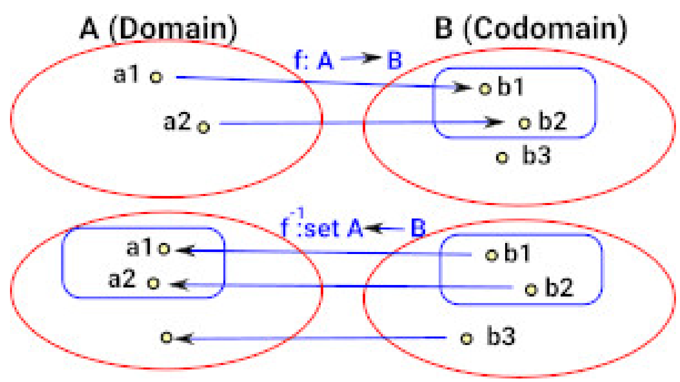
Surjective functions

- not invertible, cannot go back to the same set, but can go back to set of sets
- give rise to fiber structure in Topology and Category Theory



Injective functions

- not invertible, cannot go back to the same set, but can define subset
- give rise to sub-objects in Topology and pullbacks in Category Theory



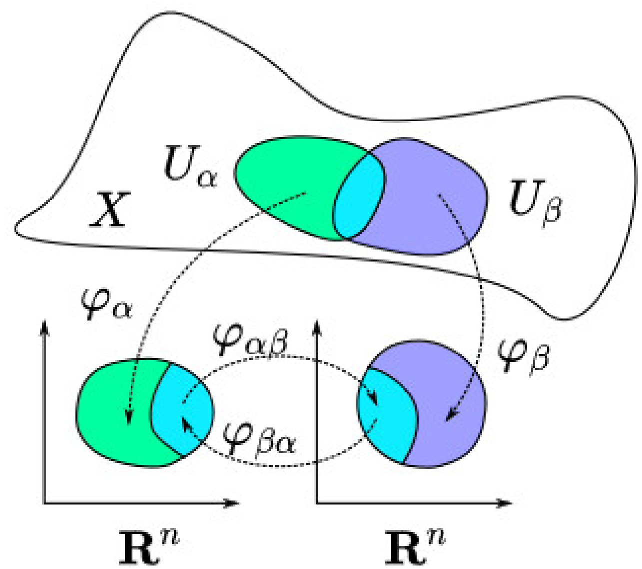
Manifolds

Motivation

- describe complicated structures as topological properties of simpler spaces

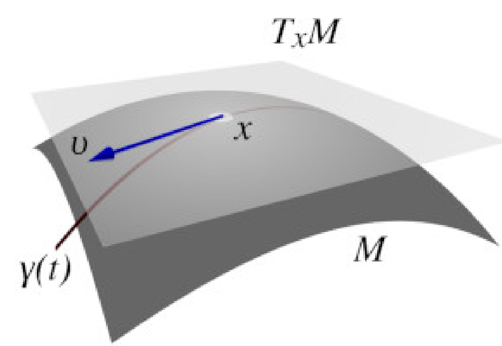
Definition

- a topological space that is locally Euclidean space at every point
- smooth manifolds are differentiable
  → allows calculus
- bijective maps/charts: φ_α and φ_β
  all φ form atlas
  transition maps $\varphi_{\alpha\beta}$ and $\varphi_{\beta\alpha}$
  manifold ↔ curve ↔ Euclidean coordinates



Tangent manifold

- tangent vectors in a given basis at all points
  → basis is important for properties between points
  → change of basis can be expressed as tensor
- Riemann metric tensor defined at every point $p \in M$

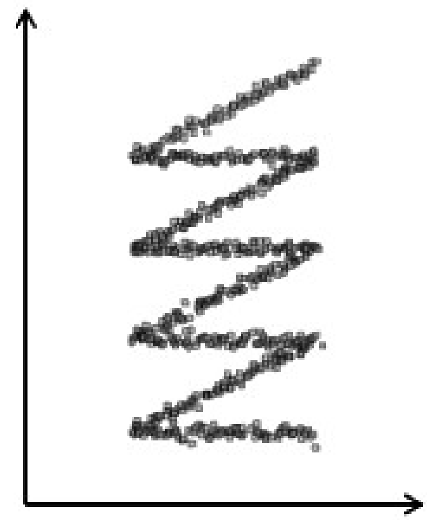


$$p \mapsto g_p(\boldsymbol{x}(p), \boldsymbol{y}(p)) = \boldsymbol{x}(p) \cdot \boldsymbol{y}(p) \in \mathbb{R}$$

Manifolds (cont.)

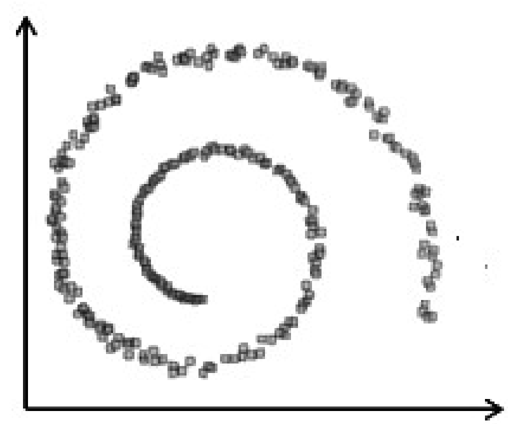
Intuition

- manifold is a low-dimensional smooth object embedded in high-dimensional space
- manifold cannot contain self-intersections, components with different dimensions

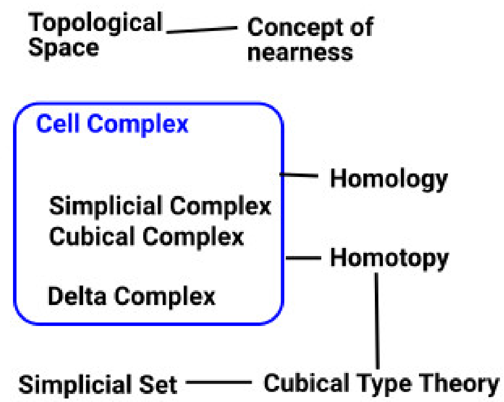


Manifold hypothesis

- high dimensional data are points in a low dimensional manifold with added high dimensional noise
- extrinsic dimensionality of dataset often larger than intrinsic dimensionality of phenomenon
  → dimensionality reduction techniques
  → embedded sub-space within original space with some degrees-of-freedom
- implications to ML
  → classification separates entangled manifolds
  → visualization of deep learning
  → assess the required size of neural network
  → design NN layers to manipulate manifolds



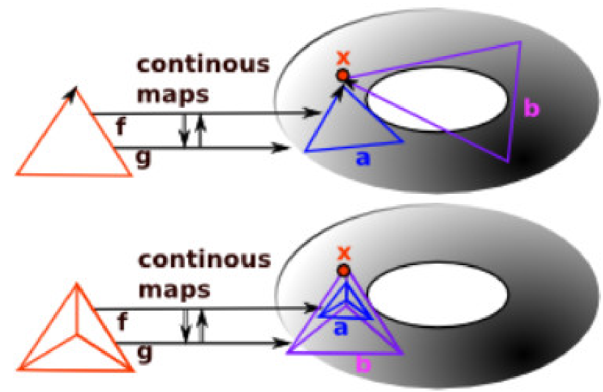
Algebraic Topology



Groups

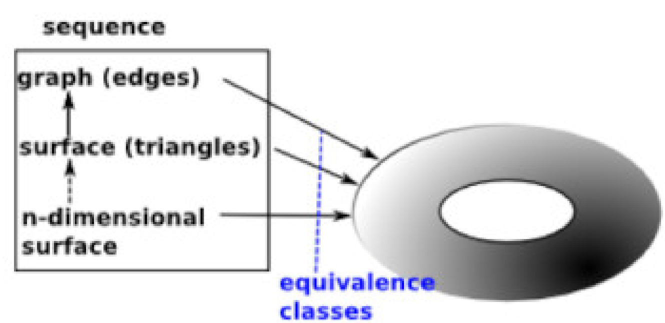
- sets with addition, multiplication, composition
- can be symmetric, communicative (Abelian), ...
- they seem to be nearly ubiquitous
  → algbr. geometry, number theory, cryptography

Homotopy



- equivalence between a circle and loops on a topological object of interest
  → circle can be defined to lie on an n-sphere
- loops generate a group of algebraic objects
  → via composition

Homology

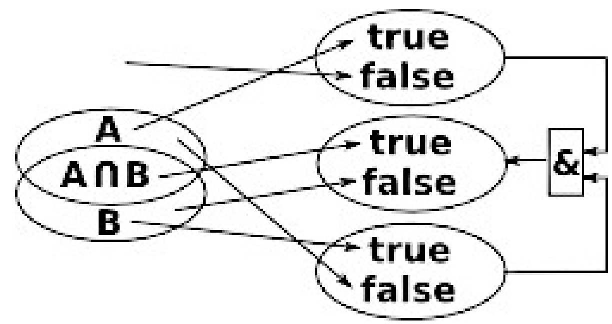


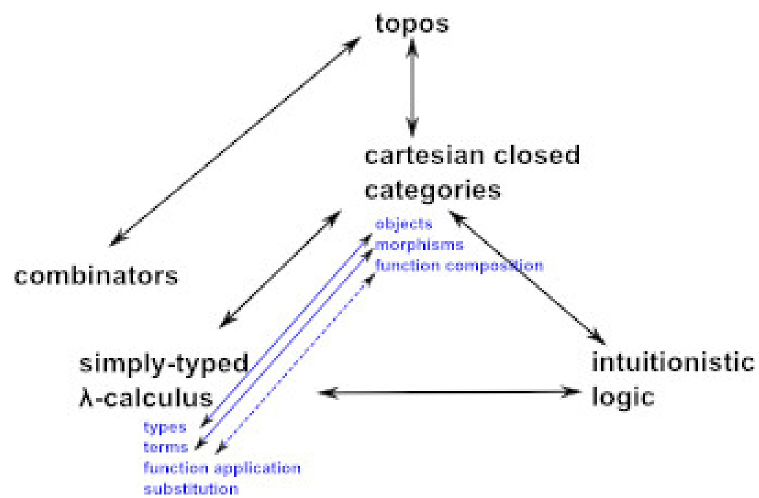
- from circle to a closed n-dimen. manifold
  → generalizes homotopy
- relates algebraic structures to topological structures
  → the key idea of algebraic topology

Topology and Logic

... can be inter-linked as

- define subsets in Venn diagram by logical funct's
- logical functions are subject to Boolean algebra
- deep connection between
  → λ-calculus: types, variables, and functions
  → constructive logic: propositions and proofs
  → cartesian closed categories: objects

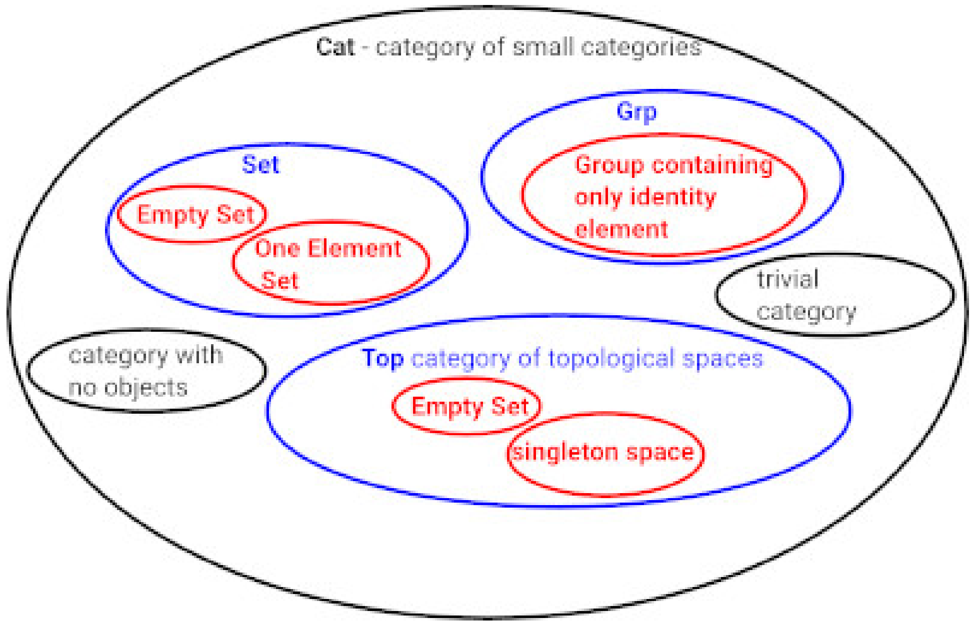




Logic types

- constructive: only consider values that can be proved true or false
- propositional: most common, assume operators and/or/negation
- first-order: extends propositional logic with predicates and quantifiers
- higher-order: adds metapredicate and quantifiers and assume sets of sets

Category Theory



Main objective

- generalizes many foundational mathematical concepts
  → abstracts to a high-order theory, defines laws of a general/free algebra
- mathematical structures: sets, groups, topological spaces, vectors etc.
- similar to set theory, but elements are complete objects within categories
  → links/arrows between objects define structure of the category

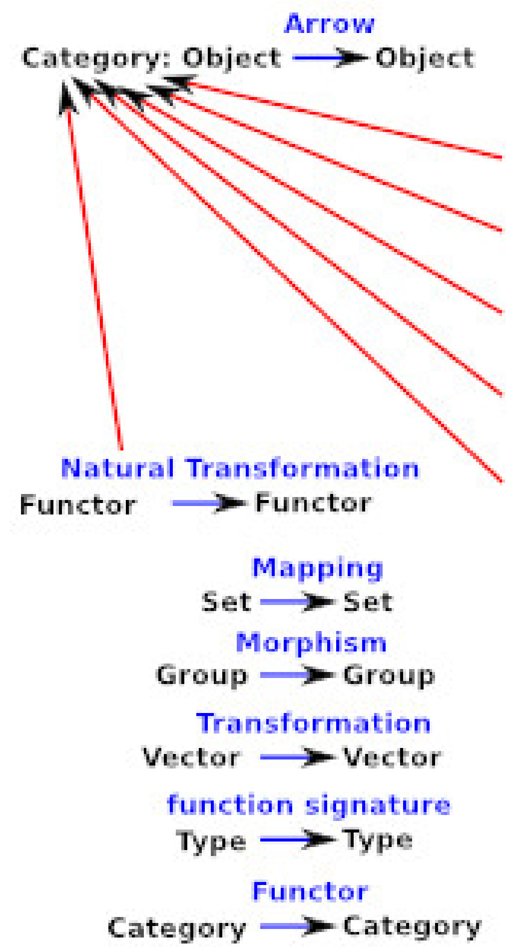
Two basic approaches

- algebra of functions: objects are functions, links are function composition
- graph theory: objects are nodes, links are graph edges

Category Theory (cont.)

Key ideas

- objects can be complete algebras themselves
  → but objects cannot be completely arbitrary
  → defines/describes objects up to isomorphism
- describe objects externally
- this can yield information about
  → axioms
  → rules of association
  → properties transferable between objects
  → properties universal to objects
- links/arrows/paths generalizes morphism
- initial object: has no morphisms into it
- terminal object: has unique morphism with all other objects in the same category

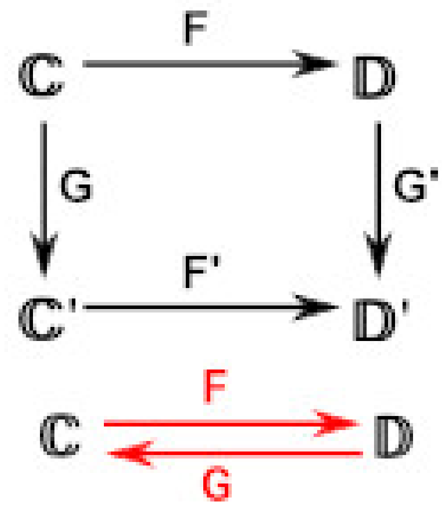


Set Theory vs. Category Theory

- sets: structure as internal associations among subsets
- categories: structure as external associations among categories

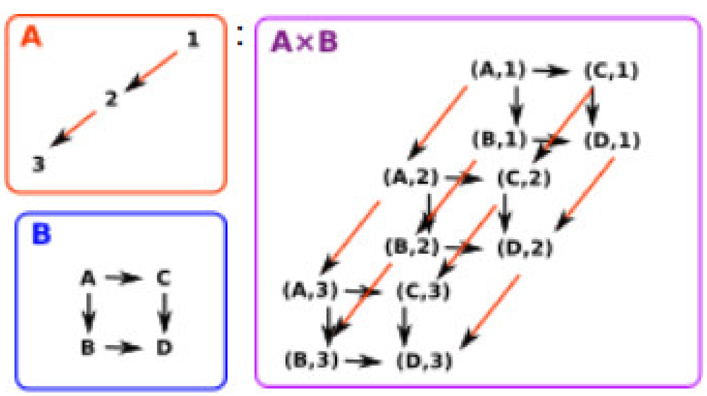
Category Theory (cont.)

Arrow diagrams

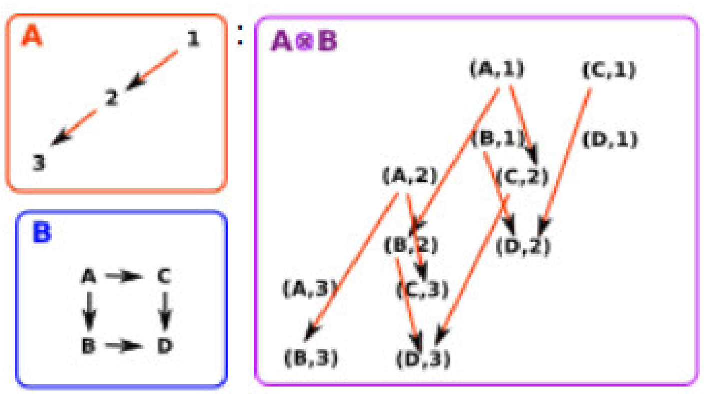


- objects and their associations represented as directed graphs
- arrows represent functions, morphisms and other $\rightarrow$ generalized as <u>functors</u>
- $F'(G(C))$ is isomorphic with $G'(F(C))$
- identity functors $G(F(C)) = 1_C$ and $F(G(D)) = 1_D$
- algebras with universal constructions
  $\rightarrow$ product (pullback) generalizes limit
  $\rightarrow$ sum (cross-product) generalizes co-limit

Cartesian product example



Tensor product example

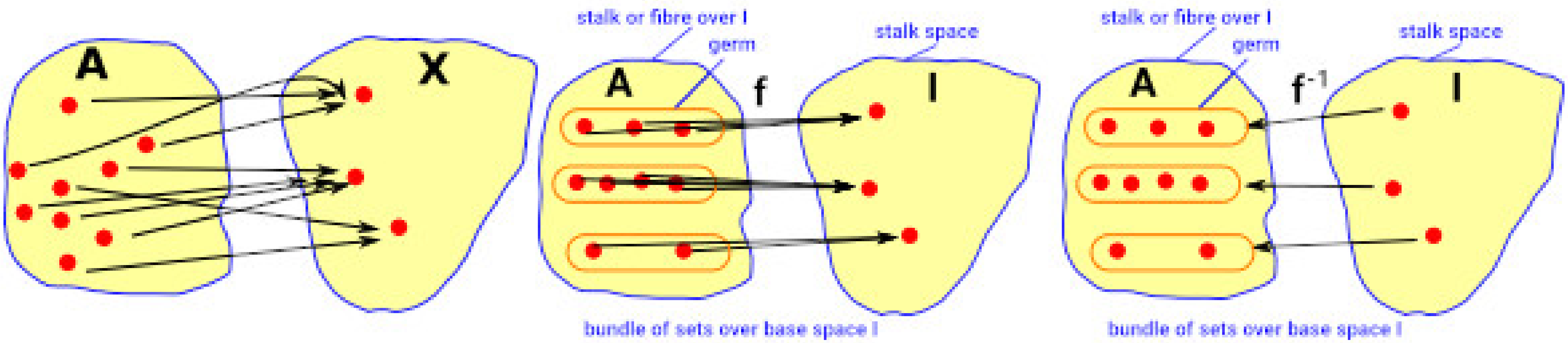


Fibers in Category Theory

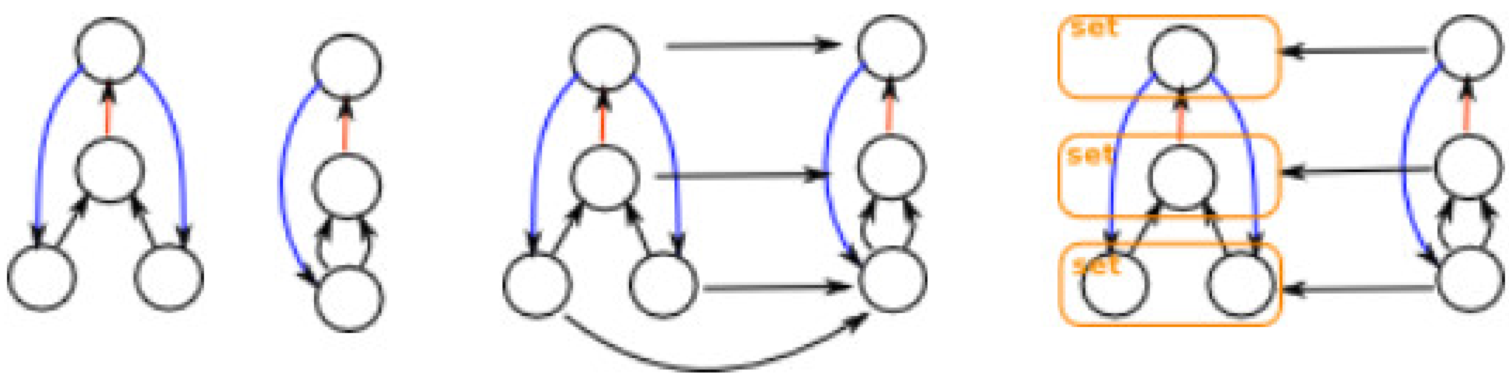
Key idea

- index one category over another category using arrows
- generalizes concepts of
  → projection, pullback and pushout, sorting, partitioning, etc.
- can be further generalized as sheaf
  → track locally defined data attached to open sets in topological space

Sets example



Graphs example



Part 3:
Topological Data Analysis

Topological Data Analysis (TDA)

Aims

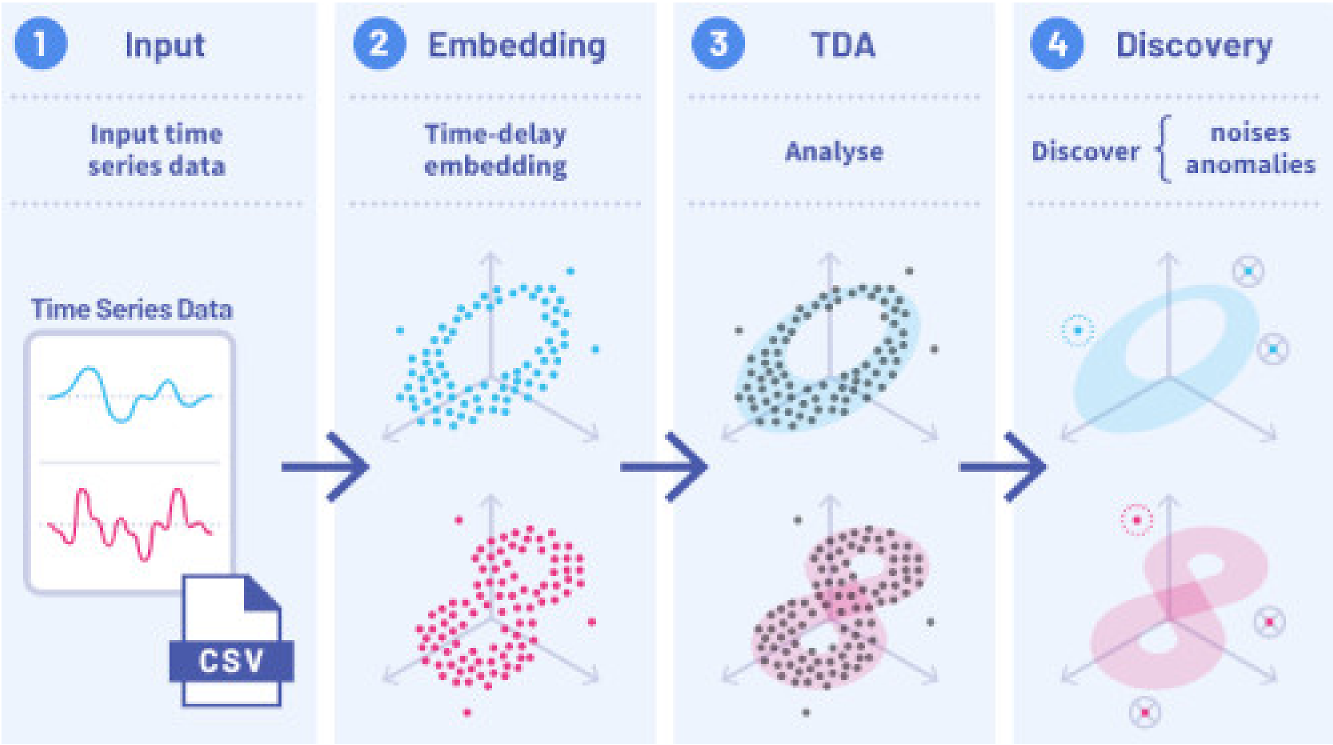
- exploit topology and geometry to define relevant data features
  → summarize and visualize complex data
  → input to machine learning models
- representation of structure of data
  → shape, connectivity, holes/voids

Data

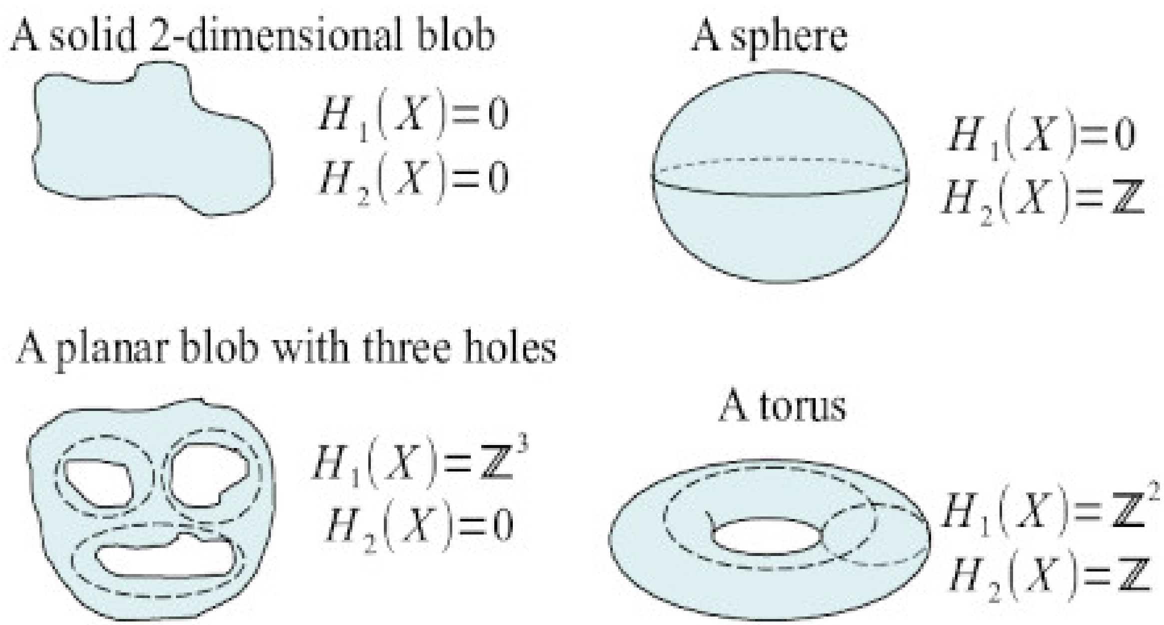
- point clouds
- 2D/3D images
- multivariate series

Basic methods

- clustering
- manifold learning and classification
- nonlinear dimension reduction
- persistent homology



Topological Data Analysis (TDA) (cont.)



Combining methods from

- geometry and topology
- algebraic topology
- differential geometry
- computational geometry
- data analysis

Topological data descriptors

- multi-scale, global/local
- topological invariants
  → robustness against perturbations and outliers

Basic idea

- associate computable algebraic structures to manifolds
  → invariant under homeomorphisms (continuous transformations)
- homology groups are combinatorial representations of manifolds
  → chain complexes

Simplexes

k-simplexes Δ^k

- a convex-hull of $(k+1)$ points, $v_i \in \mathbb{R}^{k+1}$, $i = 0, 1, \ldots, k$
- basic building block for complex simplexes, and simplex chains
- they can be mapped (embedded) to a topological manifold X, i.e., $\sigma : \ \Delta^k \mapsto X$

Face Δ^{k-1} of Δ^k

- delete one point in the hull $[v_0, v_1, \ldots, v_k]$ defining Δ^k
- denoted as, $F_i^k : \ \Delta^{k-1} \mapsto \Delta^k$, $i = 0, 1, \ldots, k$

k-chain

$$\cdots \xrightarrow{\partial_{d+1}} C_d(X) \xrightarrow{\partial_d} C_{d-1}(X) \xrightarrow{\partial_{d-1}} \cdots \xrightarrow{\partial_2} C_1(X) \xrightarrow{\partial_1} C_0(X) \xrightarrow{\partial_0} 0$$

- sequence of vector spaces (complexes), $C_k(X)$
- $C_k(X)$ is linear combination of all k-simplices Δ^k with coefficients in $\mathbb{F}_2$
- $\partial_k : \ C_k(X) \mapsto C_{k-1}(X)$ are boundary maps (homomorphisms)

$$\partial_k(\sigma) = \sum_i \sigma \circ F_i^k = \sum_{i=0}^{k} (-1)^i (v_0, \ldots, \hat{v}_i, \ldots, v_k)$$

- crucially, $\partial_k \circ \partial_{k-1} = 0$ (boundary does not have boundary)

Homology

Homology group (class) of X

$$H_k(X) = Z_k(X) \,/\, B_k(X)$$

- $Z_k(X)$ are k-cycles
- $B_k(X)$ are k-boundaries
- quotient $H_k(X)$ is a type of vector space
- $H_k(X)$ are topological invariants under homeomorphisms

$\beta_0 = 1; \beta_1 = 2$

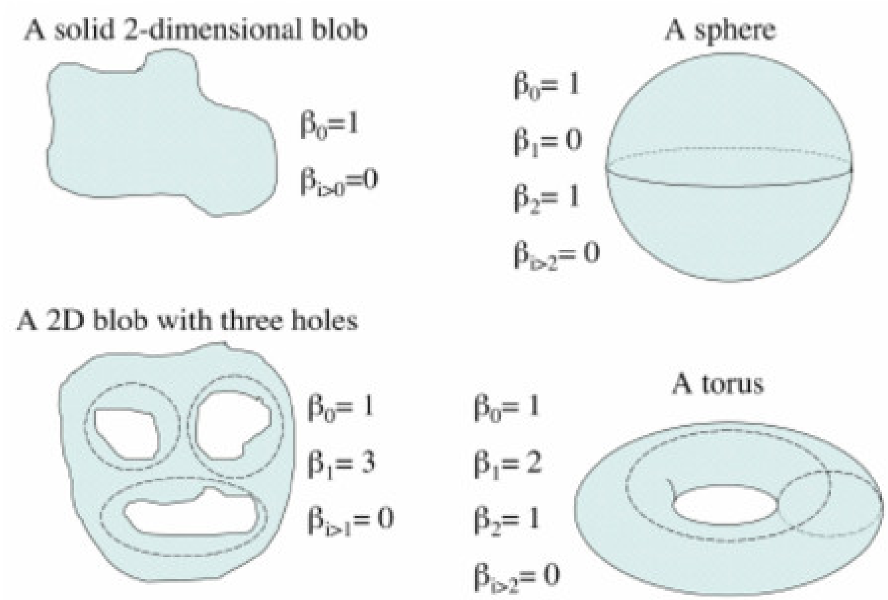
$\beta_0 = 2; \beta_1 = 1$

Betti-numbers

$$\beta_k = \mathrm{rank} H_k(X)$$

or, $$\beta_k = \dim H_k(X)$$

- β_0 is the count of connected components
- β_1 is the count of cycles (1D holes)
- β_2 is the count of voids (2D cavities)



Persistent Homology

Filtration

- homology of filtered chain complexes $C^{\epsilon_0} \subset C^{\epsilon_1} \subset C^{\epsilon_2} \subset \cdots$
- or, homology of inclusion of spaces $X^{\epsilon_0} \subset X^{\epsilon_1} \subset X^{\epsilon_2} \subset \cdots$
- explores object topology across different scales $\epsilon_0 < \epsilon_1 < \epsilon_2 < \cdots$
- homology is functorial: if $\iota^{i,j} : C^{\epsilon_i} \mapsto C^{\epsilon_j}$, then $H_k(\iota^{i,j}) : H_k(C^{\epsilon_i}) \mapsto H_k(C^{\epsilon_j})$

Filtration function

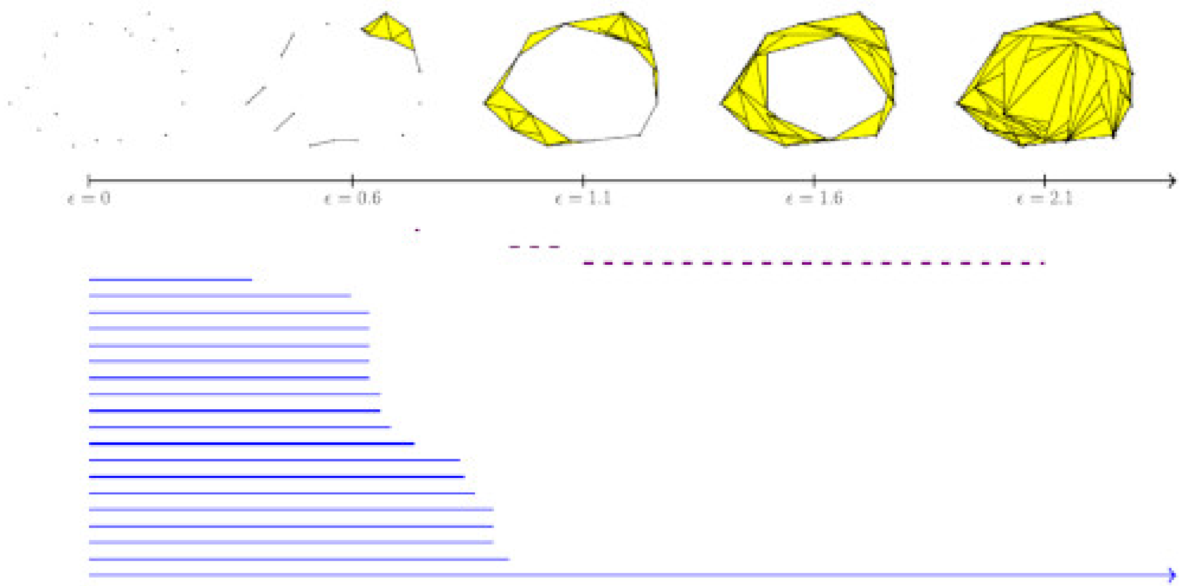
$$f : X \mapsto \mathbb{R} \quad \Rightarrow \quad X^{\epsilon} = f^{-1}(\mathbb{R}_{<\epsilon})$$

Point cloud $Y^{\epsilon} \subset (M, d)$

$$Y^{\epsilon} = \cup_{y \in Y} B_{\epsilon}(y) = g^{-1}(\mathbb{R}_{<\epsilon})$$

- (M, d) is metric space
- $g : M \mapsto \mathbb{R}$ is filtration function, i.e., $g(m) = \min_{y \in Y} d(m, y)$
- $B_{\epsilon}(y)$ is open ball of radius ϵ centred at y
- can consider either Čech complex or Vietoris-Rips complex

Homology Complexes for Point Cloud



Computing Vietoris-Rips complex

- add all points in cloud
- add edges between all points with distance $\leq \epsilon$
- identify triangles
- identify tetrahedrons having all faces (triangles)
- ... and so on (to add higher-order polytopes)

Other complexes

- alpha, witness, cubical, Delaunay, Excursus, ...

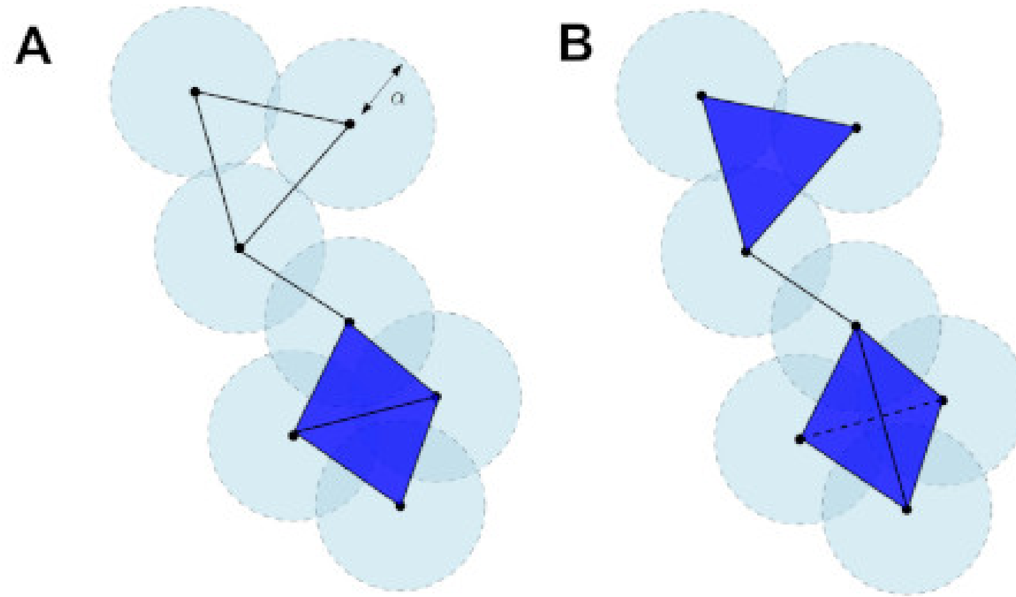
Homology Complexes for Point Cloud (cont.)

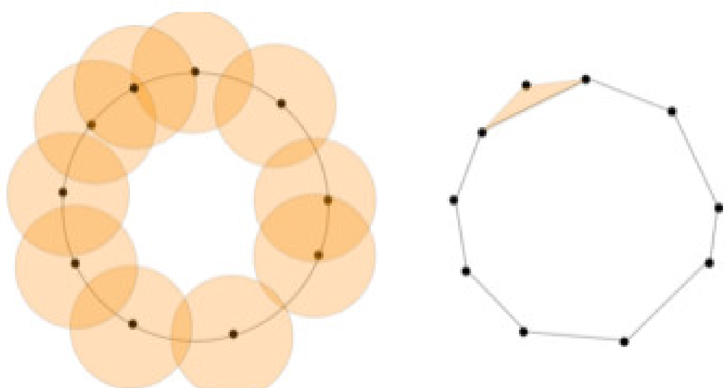
A. Čech complex

- bottom is a union of two adjacent triangles
- it has dimension 2

B. Vietoris-Rips complex

- bottom is a tetrahedron spanned by four vertices and all faces
- it has dimension 3
- simple distance-based filtration





Čech complex

- is sub-complex of Vietoris-Rips complex
- more computationally expensive
  → higher order intersections of balls
  → it is a nerve of balls

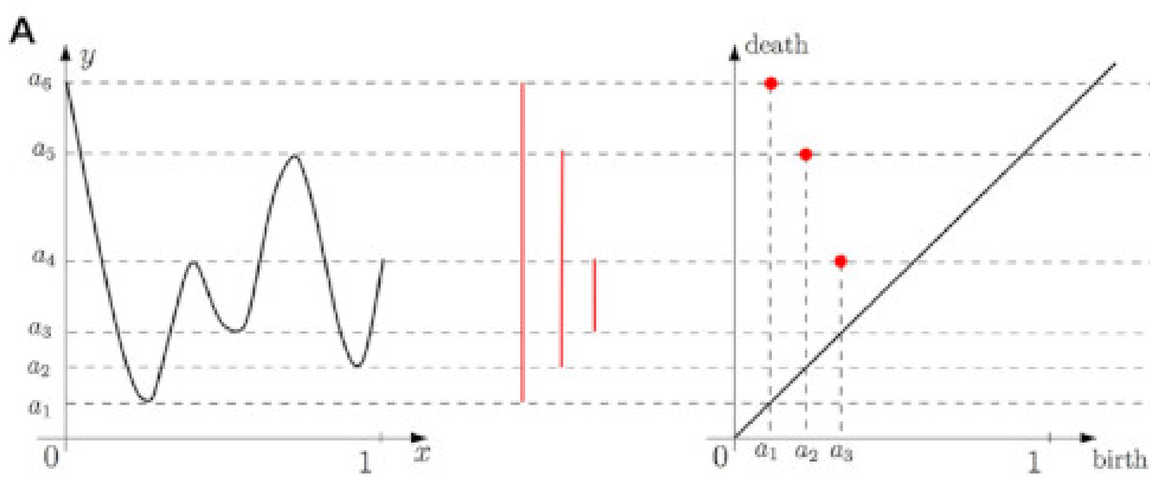
Persistent Homology (cont.)

Life-span of homology

- homology class $\alpha \in H_k(C^{\epsilon_i})$ is born at C^{ϵ_i}, if it is not in $H_k(C^{\epsilon_{i-1}})$
- it dies at C^{ϵ_j}, if it is not in $H_k(C^{\epsilon_{j+1}})$
- $(\epsilon_j - \epsilon_i)$ is persistence of α
  → information how homology (and topology) changes across filtration scales
  → if ϵ_j is infinity or the largest one, homology class is said to be persistent
  → there can be multiple homology classes with the same span $(\epsilon_j - \epsilon_i)$
- the persistence reveals important topological features present across scales
  → robust to perturbations and noise

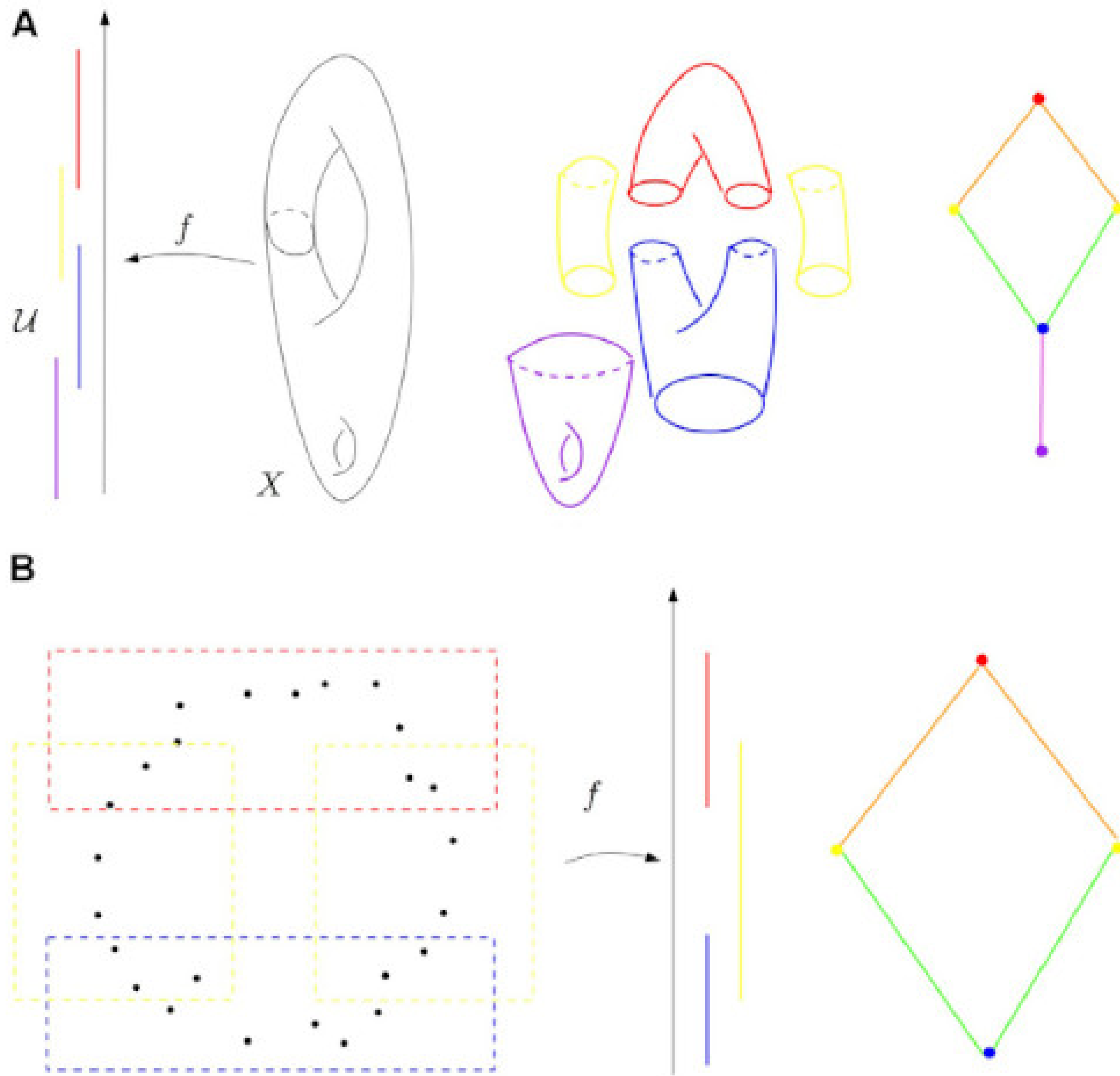
Persistent diagrams

- visualizing persistence topological features
  → scatter plot of birth vs. death values
- one of the descriptors of (data) topology



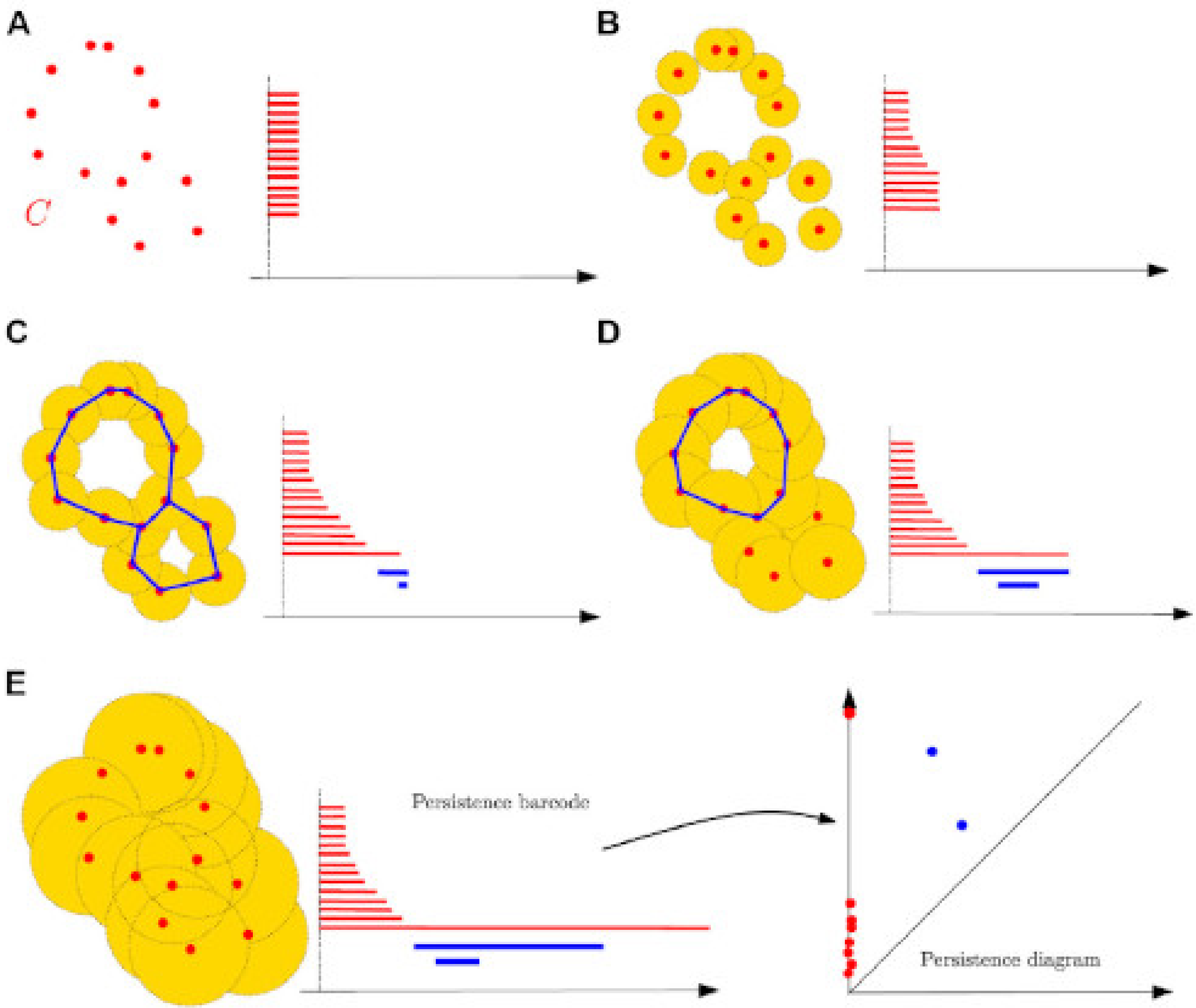
Persistent Homology Examples

Persistent barcodes and nerves



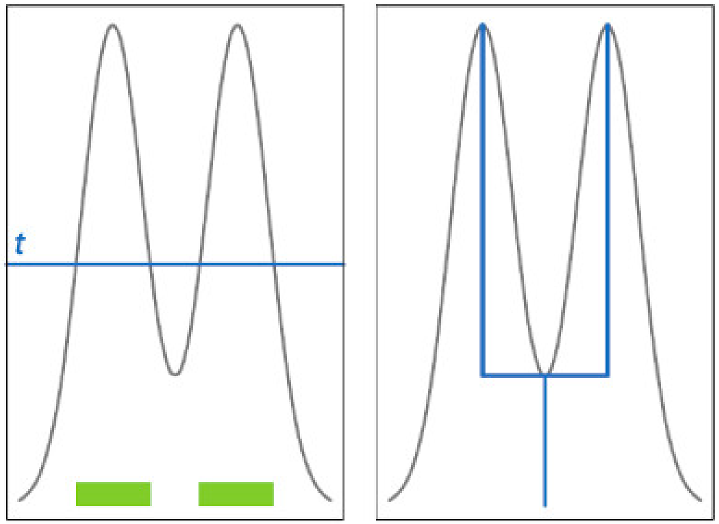
Persistent Homology Examples

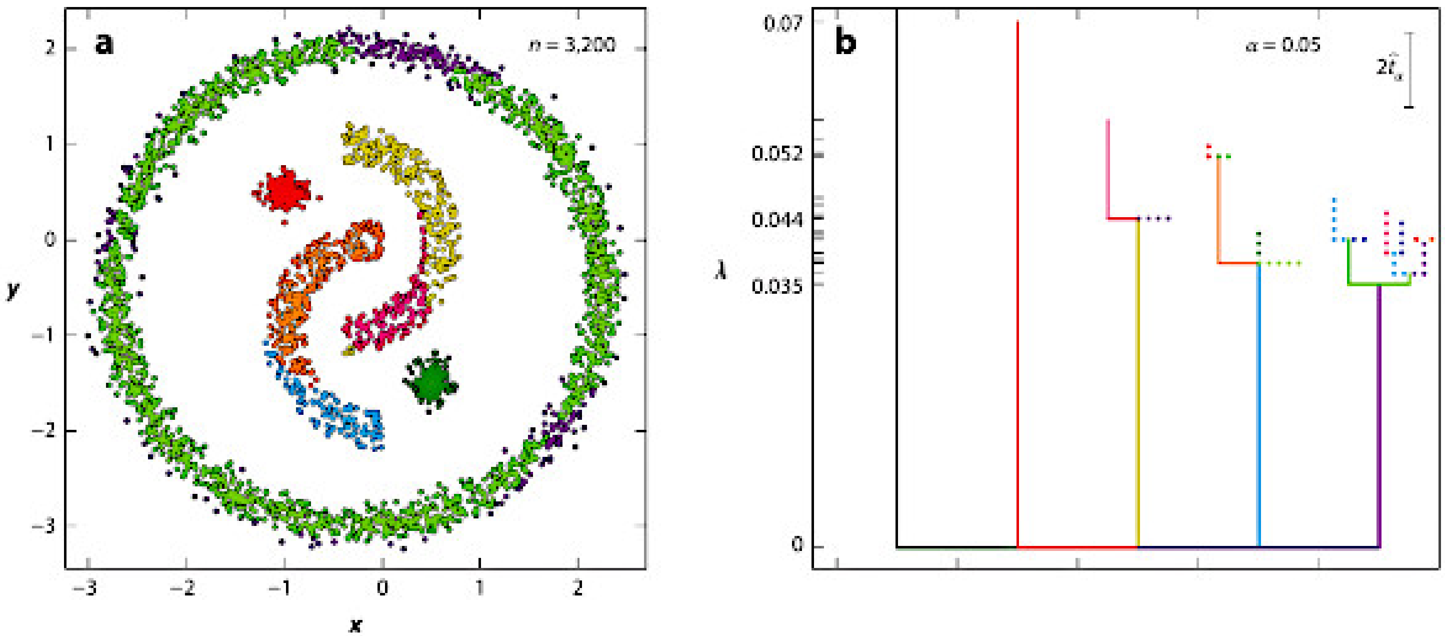
Persistence barcodes and diagrams



Persistent Homology Examples

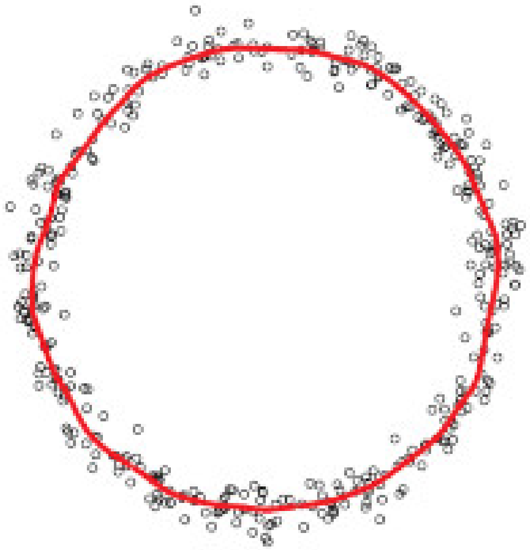
Density trees

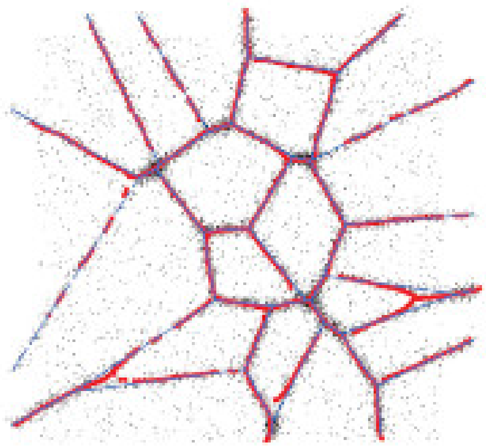


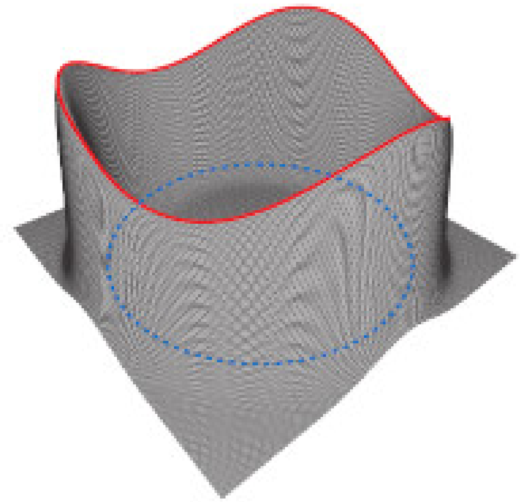


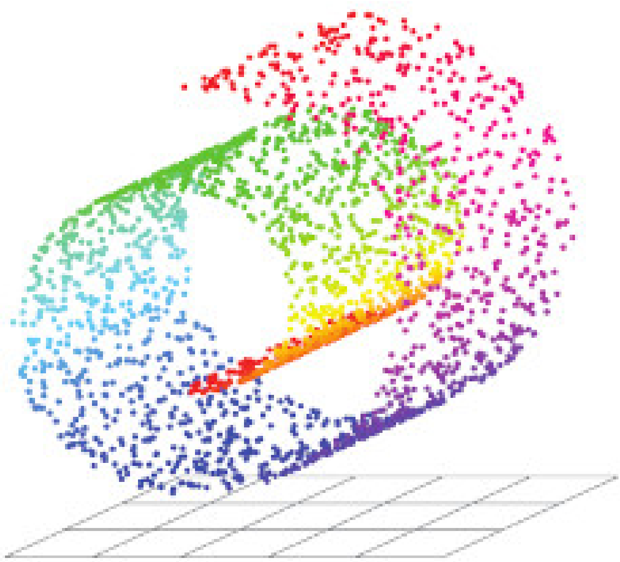
Persistent Homology Examples

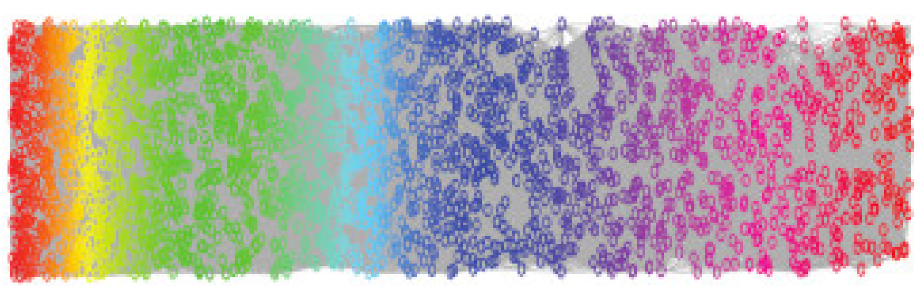
Low-dimensional manifolds, ridges and stratified spaces











Persistent Homology (cont.)

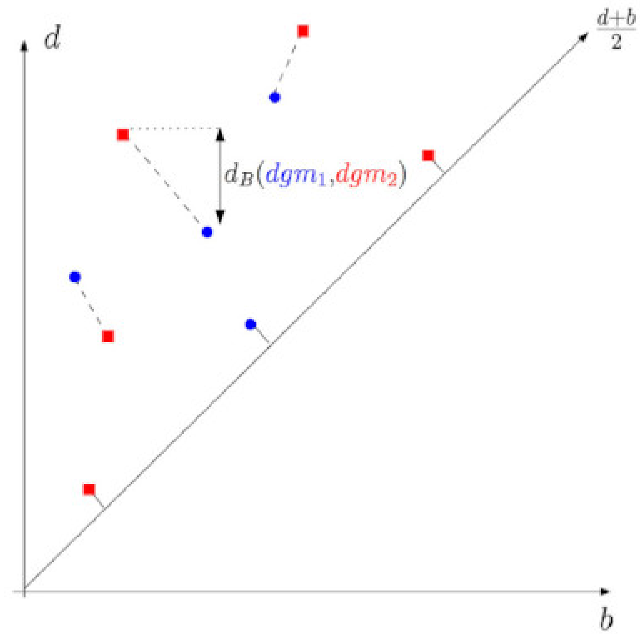
Wasserstein distance

- given real-valued functions f and g with persistence diagrams $\mathcal{D}_f$ and $\mathcal{D}_g$

$$W_p(\mathcal{D}_f, \mathcal{D}_g)^p = \inf_{\text{matching } m} \sum_{(\epsilon,\epsilon')\in m} \left\|\epsilon - \epsilon'\right\|_\infty^p$$

- W_∞ is called bottleneck distance, and by stability theorem

$$W_\infty(\mathcal{D}_f, \mathcal{D}_g) \leq \|f - g\|_\infty$$



Cover of set Y

- collection of subsets of Y whose union is Y
  $\rightarrow$ can be used to define k-simplices in the Čech complex

Nerve of collection of sets

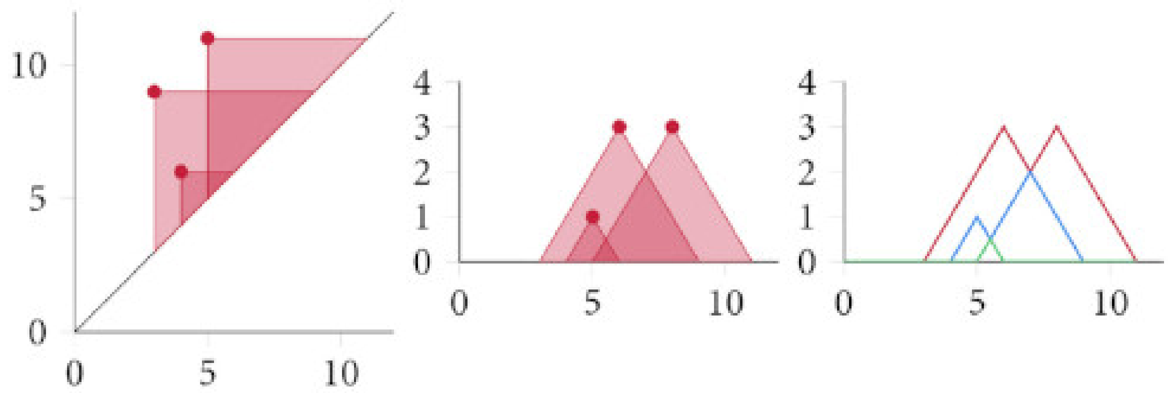
- simplical complex with vertices representing each set

Nerve theorem

- nerve of the cover and the space of Y have the same homology
  $\rightarrow$ if the cover is well-behaved

Persistent Homology (cont.)

Persistence landscape



Other descriptors

- Betti curves, persistence surfaces, persistence modules, persistence images, deep sets, other similarity/distance based methods (e.g. kernels)
  → the aim is robustness and efficiently computable
- representations in vector and function spaces
  → vectorization and discretization
  → mappings are not injective (information loss)
- applications
  → features for machine learning (including regularization)
  → summary statistics for hypothesis testing
  → data model selection and stopping criteria in ML model training
  → anomaly detection in data and in ML models

Persistent Homology - Practical Aspects

Random data

- unknown generating distribution
  → including the support
- possibly small data sizes
  → problem with complex topology/homology
  → complex geometric shapes are difficult to analyze anyway

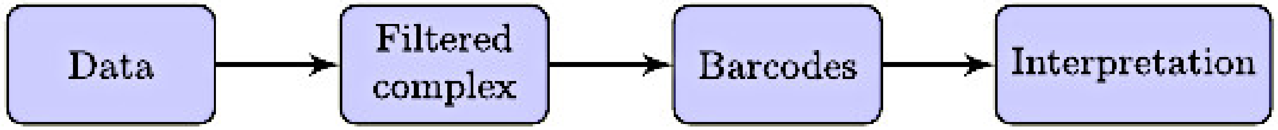
Estimating persistent homology

- estimating persistent diagrams is most common
  → need convergence to true descriptors
  → need to suppress noise in data (observations)
- kernel density estimators
- using projections, transformations and Bayesian methods

Estimating multiple descriptors

- key idea is to compute e.g. persistent diagrams for multiple subsets
- then determine central tendency and confidence regions
  → bootstrapping and asymptotic normality
- or, map persistent diagrams to spaces better suited for statistical evaluations

Computing Persistent Homology



- computationally expensive for large data sets (points, images, graphs)
- computations rely on linear algebra, approximations, and reductions $\rightarrow$ efficient algorithms exist

Alternatives to persistent homology

- Mapper algorithm, Euler calculus, cellular sheaves

Open-source libraries

- Dionysus, DIPHA, Gudhi, Hera, javaPlex, jHoles, Perseus, Persistence Landscape Toolbox, PHAT, Ripser, RIVET, SimpPers, TDA Package

Complex K	**Size of K**	**Theoretical guarantee**
Čech	$2^{\mathcal{O}(N)}$	Nerve theorem
Vietoris–Rips (VR)	$2^{\mathcal{O}(N)}$	Approximates Čech complex
Alpha	$N^{\mathcal{O}(\lceil d/2 \rceil)}$ (N points in $\mathbb{R}^d$)	Nerve theorem
Witness	$2^{\mathcal{O}(\lvert L \rvert)}$	For curves and surfaces in Euclidean space
Graph-induced complex	$2^{\mathcal{O}(\lvert Q \rvert)}$	Approximates VR complex
Sparsified Čech	$\mathcal{O}(N)$	Approximates Čech complex
Sparsified VR	$\mathcal{O}(N)$	Approximates VR complex

Part 4:
Conclusion

Take-Home Messages

1. Homology

- It yields invariant topological descriptors.
- These descriptors are robust, low-dimensional representations of data.
- Topology can be effectively approximated by simplical complexes.
- Simplical complexes can be embedded in vector spaces as features for ML.

2. Persistence homology (PH)

- Filtration defines life-span of topological features over scales.
- PH descriptors include barcodes, diagrams, landscapes and other.
- Interpreting these descriptors can be tricky.
- PH is often computationally demanding, but libraries are available.

3. Topological data analysis TDA)

- TDA is often a synonym for PH.
- Topological descriptors help to visualize structure in data and in ML models.
- For data analysis, topological descriptors must be combined with statistical methods.

Take-Home Messages (cont.)

4. Open research problems

- heavy mathematics behind topology and homology
  → define user-friendly data analysis tools
- homology filtering with multiple parameters
  → basic theorems for PH no longer valid
- choosing parameter values for simplical complexes
  → maximize the number of significant topological features
- find low-dimensional embedding preserving topological features
  → preserve clusters, loops, holes, ...
- exploit topological features in statistical methods
  → general purpose methods
- more generally, fill the gap between mathematics and other disciplines
  → especially engineering

References – Journal Papers

[1] F. Chazal and B. Michel, "An Introduction to Topological Data Analysis: Fundamental and Practical Aspects for Data Scientists," *Frontiers in Artifical Intelligence*, 4:667963, 2021.

[2] R. Ghrist, "Homological Algebra and Data," *The Mathematics of Data*, IAS/Park City Mathematics 25:273-325, 2017.

[3] F. Hensel, M. Moor and B. Rieck, "A Survey of Topological Machine Learning Methods," *Frontiers in Artifical Intelligence*, 4:681108, 2021.

[4] N. Otter, M. A. Porter, U. Tillmann, P. Grindrod and H. Harrington, "A roadmap for the computation of persistent homology," *EPJ Data Science*, 6:17, 2017.

[5] L. Wasserman, "Topological Data Analysis," *Annual Review of Statistics and Its Application*, 5:501–32, 2018.

[6] Jelena Grbić, J. Wu, K. Xia and G.-W. Wei, "Aspects Of Topological Approaches For Data Science," *Foundations of Data Science*, 4(2): 165–216, 2022.

[7] A. Eskenazi and K. You, "A Beginner's Guide to Homological Algebra: A Comprehensive Introduction for Students," ArXiv:2208.11199v2 [math.HO], September 2022.

References – Books

[1] M. Robinson, *Topological Signal Processing*, Springer, 2014.

[2] T. W. Judson and R. A. Beezer, *Abstract Algebra: Theory and Applications*, GNU Free Documentation License, 2022.

[3] J. P. May, *A Concise Course in Algebraic Topology*, University of Chicago Press, 1999.

[4] A. Hatcher, *Algebraic Topology*, Cambridge University Press, 22001.

[5] T. Leinster, *Basic Category Theory*, Cambridge University Press, 2014.

Wikipedia

- https://en.wikipedia.org/wiki/Metric_space
- https://en.wikipedia.org/wiki/Metric_tensor
- https://en.wikipedia.org/wiki/Manifold
- https://en.wikipedia.org/wiki/Simplicial_complex
- https://en.wikipedia.org/wiki/Simplex
- https://en.wikipedia.org/wiki/Cech_complex
- https://en.wikipedia.org/wiki/Vietoris-Rips_complex

Thank you!

pavelloskot@intl.zju.edu.cn