The Simplex Approach

Thomas Walker

January 2024

Abstract

The practice of machine learning is inherently geometrical. Thinking of data as existing in some high-dimensional space, the features of the data set manifest as complex manifolds in this space. Such a perspective is already strongly implemented in the field of data science, with the established domain of topological data analysis. Where the structure of the data is studied directly using methods in algebraic topology. Recently, supervised learning techniques have risen as a ubiquitous tool for data analysis. Largely in part due to its inherent lack of priors. Thus there has been work to incorporate the geometrical priors of topological data analysis into supervised learning methods [3]. Such techniques are broadly referred to as extrinsic topological feature extraction, which is in contrast to intrinsic feature extraction [2]. The latter involves thinking of the application of machine learning more generally as a process distinguishing the complex manifolds corresponding to features of the data that facilitate robust inference from the data. Hence, in addition to encoding geometrical priors using theories such as algebraic topology, we can explore the processes of machine learning through a geometrical lens and motivate architectural choices by their capacity to represent complex topological features [1]. The reason for algebraic topology being the primary tool for exploring the geometrical features of data and machine learning models is for its flexibility in accommodating abstract topological features.

Contents

Part I
Theory of Algebraic Topology

1 Geometrical Structures

1.1 Simplicial Complexes

Definition 1.1.1. *A k-dimensional simplex is the convex hull of $k+1$ affinely independent points $v_0, \ldots, v_k \in \mathbb{R}$, represented by a listing of its vertices denoted $\sigma = [v_0, \ldots, v_k]$.*

Definition 1.1.2. *The faces of a k-simplex are simplicies of dimensions $0, 1, \ldots, k-1$ formed by convex hulls of proper subsets of its vertex set $\{v_0, \ldots, v-k\}$. In particular, we write $\alpha \leq \beta$ to mean that α is a face of β.*

Definition 1.1.3. *A m-dimensional geometrical simplicial complex is an object K in $\mathbb{R}^d$ that is a finite collection of simplices in $\mathbb{R}^d$ of dimensions at most m that are connected along faces, and whose faces are included in K.*

Definition 1.1.4. *An abstract simplicial complex is a list of simplicies $K = \{\sigma_1, \ldots, \sigma_n\}$ with property that if $\tau \subseteq \sigma \in K$ then $\tau \in K$*

Remark 1.1.5. *Note that an abstract simplicial complex K can be thought of as $K \subset \mathcal{P}(S)) \setminus \{\emptyset\}$, where S is a set of affinely independent points. Where K is required to satisfy the condition of Definition 1.1.4.*

Definition 1.1.6. *Let S be a set of affinely independent points. A structure $K \subset \mathcal{P}(S)) \setminus \{\emptyset\}$ is referred to as a hypergraph, with the elements of K being referred to as hyperedges.*

Remark 1.1.7. *Structures such as those defined by Definition 1.1.4 and Definition 1.1.6 are known as higher-order domains.*

Definition 1.1.8. *A rank function on a higher-order domain K is a map $\mathrm{rk} : K \to \mathbb{Z}_{\geq 0}$ such that $\mathrm{rk}(x) \leq \mathrm{rk}(y)$ for all $x, y \in K$.*

1.2 Combinatorial Complexes

Combinatorial complexes are a structure that bridges the gap between simplicial complexes, cell complexes and hypergraphs. More specifically, a combinatorial complex can be viewed as a simplicial complex whose simplices can be missing from the complex, as a cell complex with a relaxed structure, or as a hypergraph with enriched hierarchies induced by a rank function [4].

Definition 1.2.1. *A combinatorial complex is a tripe (S, K, rk), where S is a set of affinely independent points, $K \subset \mathcal{P}(S) \setminus \{\emptyset\}$ is such that $\{s\} \in K$ for all $s \in S$, and $\mathrm{rk} : K \to \mathbb{Z}_{\geq 0}$ is a rank function. Elements of S are called entities, and elements of K are called cells.*

Remark 1.2.2.

1. *The rank function of a combinatorial complex imposes hierarchy relations on the structure. Typically, the rank function is such that* $\mathrm{rk}(\{s\}) = 0$.

2. *For a combinatorial complex* K *we use* x^k *to denote a cell of* K *that has rank* k*, that is a* k*-cell. The set of all* k*-cells of a combinatorial complex* K *will be denoted* $K^{(k)}$.

For brevity, we will typically refer to the combinatorial complex (S, K, rk) as K.

Definition 1.2.3. *The dimension of a combinatorial complex* K *denoted* $\dim(K)$*, is given by the maximal rank amongst its cells.*

Definition 1.2.4. *Let* (S, K, rk) *be a combinatorial complex. For* $r, k \in \mathbb{Z}_{\geq 0}$ *with* $0 \leq r < k \leq \dim(K)$*, the* (r, k)*-incidence matrix* $B_{r,k}$ *between* $K^{(r)}$ *and* $K^{(k)}$ *is the* $\left|K^{(r)}\right| \times \left|K^{(k)}\right|$ *matrix where*

$$(B_{r,k})_{ij} = \begin{cases} 1 & x_i^r \subset x_j^k \\ 0 & \text{otherwise.} \end{cases}$$

Remark 1.2.5. *Cells* $x, y \in K$*, where* K *is a combinatorial complex, are incident if either* $x \subset y$ *or* $y \subset x$ *with* $x \neq y$*. In particular, if* $x \subset y$ *then* x *is in the down-incidence neighbourhood of* y*, whereas* y *is in the up-incidence neighbourhood of* x.

Definition 1.2.6. *Let* (S, K, rk) *be a combinatorial complex. For any* $k \in \mathbb{N}$ *the* k*-adjacency neighbourhood function of* $x \in K$ *is*

$$\mathcal{N}_{a,k}(x) := \{y \in K : \mathrm{rk}(y) = \mathrm{rk}(x), \text{ there exists } z \in K \text{ with } \mathrm{rk}(z) = \mathrm{rk}(x) + k \text{ such that } x, y \subset z\}.$$

Definition 1.2.7. *Let* (S, K, rk) *be a combinatorial complex. For any* $r \in \mathbb{Z}_{\geq 0}$ *and* $k \in \mathbb{Z}_{>0}$ *with* $0 \leq r < r + k \leq \dim(K)$*, the* (r, k)*-adjacency matrix* $A_{r,k}$ *is the* $\left|K^{(r)} \times K^{(r)}\right|$ *matrix where*

$$(A_{r,k})_{ij} = \begin{cases} 1 & x_i^r \in \mathcal{N}_{a,k}\left(x_j^r\right) \\ 0 & \text{otherwise.} \end{cases}$$

2 Homology

Homology is a way of encoding the topology of a space through chains of vector spaces and linear maps. For simplicity, we will consider vector spaces $C_0, C_1, \ldots, C_d$ over $\mathbb{F}_2$. Let $\partial_k : C_k \to C_k k - 1$ be linear maps called boundary operators which satisfy

$$\partial_k \circ \partial_{k-1} = 0,$$

for all $k = 0, \ldots, d$.

2.1 Betti Numbers

Definition 2.1.1. *A chain complex refers to a sequence*

$$0 \xrightarrow{\partial_{d+1}} C_d \xrightarrow{\partial_d} C_{d-1} \xrightarrow{\partial_{d-1}} \ldots \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} \ldots \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0$$

where $C_{d+1} = C_{-1} = 0$ *by convention.*

- *Elements in the image of ∂_k are called boundaries.*
- *Elements in the kernel of ∂_{k-1} are called cycles.*

Moreover, $\ker(\partial_k)$ and $\mathrm{im}(\partial_{k+1})$ are subspaces of C_k with property that

$$B_k := \mathrm{im}(\partial_{k+1}) \subseteq \ker(\partial_k) =: Z_k$$

Definition 2.1.2. *The k^{th} homology group is the quotient group*

$$H_k := Z_k/B_k = \ker(\partial_k)/\mathrm{im}(\partial_{k+1}),$$

for $k = 0, 1, \ldots, d$.

Definition 2.1.3. *The elements of H_k have the form*

$$[z] = z + B_k = \{z + b \in Z_k : b \in B_k\}$$

and are referred to as homology classes.

In particular, $[z] = [z + b]$ for any $b \in B_k$.

Definition 2.1.4. *The k^{th} Betti number, β_k, of H_k is the dimension of H_k as a vector space. That is,*

$$\beta_k := \dim(H_k).$$

2.2 Simplicial Homology

Given an abstract simplicial complex K let $K^{(k)} = \{\sigma_1, \ldots, \sigma_m\}$ be set of all k-dimensional simplices in K. Let $C_k(K)$ be the $\mathbb{F}_2$-vector space whose elements are of the form

$$\sum_{j=1}^{m} n_j \sigma_j$$

for $n_j \in \mathbb{F}_2$. The boundary operators $\partial_k : C_k(K) \to C_{k-1}(K)$ are defined on a k-simplex $\sigma = [v_0, \ldots, v_k]$ by

$$\partial_k \sigma := \sum_{j=1}^{k} [v_0, \ldots, \hat{v}_j, \ldots, v_k]$$

where $\hat{v}_j$ indicates that v_j is omitted. This is extended linearly to $C_k(K)$ by

$$\partial_k \left(\sum_{j-1}^{m} n_j \sigma_j \right) := \sum_{j=1}^{m} n_j \partial_k \sigma_j.$$

Using these one has

$$H_k(K) \cong \mathbb{F}_2^{\beta_k}$$

where

$$\beta_k(K) = \dim(H_k(K)) = \mathrm{null}(\partial_k) - \mathrm{rank}(\partial_{k+1})$$

for $k = 0, \ldots d$. Let $m_k := \left|K^{(k)}\right|$, be the number of k-simplicies in K. In a d-dimensional simplicial complex K we have that

$$|K| \leq \sum_{i=0}^{d} \binom{m_0}{i+1}$$

where m_0 is the size of the vertex set.

Remark 2.2.1. *The cost of computing $\beta(K)$ is about $o(|K|^{2.38})$*

Definition 2.2.2. *For K_1 and K_2 abstract simplicial complexes. A simplicial map is a map $f : K_1^{(0)} \to K_2^{(0)}$ is given by*

$$f([v_0, \dots, v_k]) = [f(v_0), \dots, f(v_1)].$$

Remark 2.2.3.

1. *A simplicial map induces a map between chain complexes $f : C_k(K_1) \to C_k(K_2)$, given by*

$$\sum_{j=1}^{m} n_j \sigma_j \mapsto \sum_{j=1}^{m} n_j f(\sigma_j).$$

   Which in turn induces a map between homologies $H_k(f) : H_k(K_1) \to H_k(K_2)$ given by

$$\sum_{j=1}^{m} n_j \sigma_j \mapsto \sum_{j=1}^{m} n_j f(\sigma_j).$$

2. *For simplicial maps $f : K_1^{(0)} \to K_2^{(0)}$ and $g : K_2^{(0)} \to K_3^{(0)}$, their composition $g \circ f : K_1^{(0)} \to K_3^{(0)}$ is also a simplicial map. Which likewise induces a map between homologies $H_k(g \circ f) : H_k(K_1) \to H_k(K_3)$. In particular, we have that*

$$H_k(g \circ f) = H_k(g) \circ H_k(f).$$

2.3 Vietoris-Rips Complex

Let δ be a metric on $\mathbb{R}^d$. The Vietoris-Rips complex at scale $\epsilon \geq 0$ on X is the abstract simplicial complex

$$\mathrm{VR}_\epsilon := \{[x_0, \dots, x_k] : \delta(x_i, x_j) \leq 2\epsilon,\ x_0, \dots, x_k \in X,\ k = 0, \dots, n\}.$$

From a dense enough sample X from a manifold $M \subseteq \mathbb{R}^d$ and a sufficiently small scale, VR_ϵ recovers the topology of M.

Proposition 2.3.1. *Let $X = \{x_1, \dots, x_n\} \subseteq \mathbb{R}^d$ be $\left(\frac{\epsilon}{2}\right)$-dense in a compact Riemannian manifold $M \subseteq \mathbb{R}^d$. That is, for every $p \in M$ there exists $x \in X$ such that $\|p - x\| < \frac{\epsilon}{2}$. Let τ be the condition number of M. Then for any $\epsilon < \sqrt{\frac{3\tau}{5}}$, the union of balls $V = \bigcup_{i=1}^{n} B_\epsilon(x_i)$ deformation retracts to M. In particular, the homology of V equals the homology of M.*

Remark 2.3.2. *The condition number of a manifold encodes its local and global curvature properties*

2.4 Persistent Homology

Note that $\mathrm{VR}_0(X) = \{[x] : x \in X\}$ is a collection of 0-dimensional simplicies with $\beta_0 = |X|$ and all other Betti numbers equal to 0. That is, $\mathrm{VR}_0(X)$ overfits the data. As ϵ increases, the topology of $\mathrm{VR}_\epsilon(X)$ becomes richer. However, as $\epsilon \to \infty$ all points become vertices of a $|X|$ dimensional simplex, giving a contractible topological space.

Definition 2.4.1. *A persistence barcode is an interval $[\epsilon, \epsilon')$, where at in $\mathrm{VR}_\epsilon(X)$ a topological feature becomes present, and at $\mathrm{VR}_{\epsilon'}(X)$ the feature disappears.*

Remark 2.4.2. *The length of a persistence barcode, that is $\epsilon' - \epsilon$, is often referred to as the persistence of the topological feature.*

Prominent features have large persistence. Suppose that at some scale ϵ_* all prominent features are represented. One may be interested in selection ϵ_* from a finite set of scales, say

$$\epsilon_0 < \epsilon_1 < \cdots < \epsilon_m$$

whose corresponding simplicial complexes are $K_j = \mathrm{VR}_\epsilon(X)$ for $j = 0, \ldots, m$ say. We note that

$$K_0 \subseteq K_1 \subseteq \cdots \subseteq K_m$$

is a filtration of simplicial complexes. Let $f_j : K_j \hookrightarrow K_{j+1}$ be the inclusion maps. The composition $f_{j+p-1} \circ \cdots \circ f_j$ induces a map between $H_k(K_j)$ and $H_k(K_{j+p})$.

- If a homology class in $H_k(K_{j+1})$ is not present in the image of $H_k(K_j)$ then the homology class is said to be born at time j.
- For $i < j$, if different homology classes of $H_k(K_i)$ are mapped to the same homology class in $H_k(K_j)$ then one of the classes is said to have died at time j whilst the other has persisted.

Persistence barcodes keep track of the times at which classes are born and die. To calculate the persistence barcodes for a homology class one can consider the p-persistent k^{th} homology group

$$H_k^{j,p} = Z_k^j / \left(B_k^{j+p} \cap Z_k^j \right)$$

which captures the cycles in $C_k(K_j)$ that contribute to homology in $C_k(K_{j+p})$. One can chose basis elements compatible across homologies $H_k(K_{j+1}), \ldots, H_k(K_{j+p})$ for all k and p. Allowing one to track the persistence of each homology class throughout the filtration.

3 Cochains

Just as we used filtrations on simplicial complexes to discern the structure of the underlying topology, we can use filtration of combinatorial complexes to discern the structure of the underlying topology.

3.1 Cochain Spaces

Definition 3.1.1. *Let $C_k\left(K, \mathbb{R}^d\right)$ be the $\mathbb{R}$-vector space of functions $\mathbf{H}_k : K^{(k)} \to \mathbb{R}^d$, where d is the data dimension. The space $C_k\left(K, \mathbb{R}^d\right)$ is referred to as the k-cochain space, and elements of H_k in $C_k\left(K, \mathbb{R}^d\right)$ are referred to as the k-cochains.*

Remark 3.1.2. *A k-cochain can be thought of as a signal defined on the k-cells of a combinatorial complex K. Equivalently, they can be thought of as feature vectors for the cells.*

3.2 Cochain Maps

Definition 3.2.1. *For $r < k$, an incidence matrix $B_{r,k}$ induces a map $B_{r,k} : C_k(K) \to C_r(K)$ with*

$$\mathbf{H}_k \mapsto B_{r,k}\mathbf{H}_K,$$

where the right-hand side is just matrix-vector multiplication. Similarly, an adjacency matrix $A_{r,k}$ induces a map $A_{r,k} : C_r(K) \to C_r(K)$ with

$$\mathbf{H}_r \mapsto A_{r,k}\mathbf{H}_r,$$

where the right-hand side is just matrix-vector multiplication.

Remark 3.2.2. *The induced maps of Definition 3.2.1 are referred to as cochain maps.*

Cochain maps serve as mechanisms to redistribute the data on a combinatorial complex. They can be used to define higher-order messaging passing and pooling operations.

Part II
The Topology of Machine Learning

4 Vectorisation of Barcodes

Thus far we have seen that for a point cloud of data, we can construct a Vietoris-Rips complex filtration, from which we can compute homology groups. These homology groups consist of homology classes which are born and may die at various points along the filtration. Consequently, we are left with a set of vectors representing these persistence barcodes. These are inherently combinatorial objects that need to be vectorised such that supervised methods of machine learning can be applied to infer structure from these barcodes. Utilising persistence barcodes in this way, to impose geometrical priors, is an extrinsic application of algebraic topology to machine learning techniques. There has been a lot of work trying to determine vectorisation techniques which sufficiently cover the large space of possible barcodes and preserve the underlying geometrical relationships between sets of barcodes [3]. In [3] the main approaches taken to vectorise barcodes are explored and include the following.

1. Statistical vectorisation.
2. Algebraic vectorisation.
3. Curve vectorisation.
4. Functional vectorisation.
5. Ensemble vectorisation.

In [3] it is noted that statistical vectorisation seems the most effective in practice. For a point cloud, we suppose that we have the persistence barcode B. As a persistence barcode is of the form $[p, q]$ for $0 \leq p < q \leq m$ we can regard B as a multi-set of such intervals. Thus, we can consider a multiplicity function $\mu : B \to \mathbb{Z}_{>0}$, where $\mu_{p,q}$ gives the multiplicity of the interval $[p, q]$ in B.

4.1 Statistical Vectorisation

Statistical vectorisation techniques simply use summary statistics from the barcodes to construct their vector representation. One example of such an approach is the persistence statistics vector. The persistence statistics vector $\mu : B \to \mathbb{Z}_{>0}$ contains the following.

1. The mean, standard deviation, median, inter-quartile range, full range, midpoints $\frac{p+q}{2}$, persistence $q - p$ counted with multiplicity. As well as the 10^{th}, 25^{th}, 75^{th} and 90^{th} percentiles of the births p, with similar percentiles for the deaths q.
2. The total number of barcodes.
3. The entropy μ, which is given by

$$E_\mu := -\sum_{[p,q]\in B} \mu_{p,q} \left(\frac{q-p}{L_\mu}\right) \log\left(\frac{q-p}{L_\mu}\right)$$

where

$$L_\mu := \sum_{[p,q]\in B} \mu_{p,q}(q-p).$$

4.2 Algebraic Vectorisation

Algebraic vectorisations use $\mu : B \to \mathbb{Z}_{>0}$ for a barcode B to construct a polynomial map that is then used to generate the vector.

Definition 4.2.1. *The ring of algebraic functions on $\mu : B \to \mathbb{Z}_{>0}$ consist of the real polynomials in variables $\{x_0, y_0, \ldots, x_m, y_m\}$ for three exist polynomials $\{g_i : 0 \leq i \leq m\}$ such that*

$$\frac{\partial f}{\partial x_i} + \frac{\partial f}{\partial y_i} = (x_i - y_i)g_i$$

for every $i = 0, \ldots, m$.

One algebraic vectorisation, referred to as the Adcock-Carlsson coordinates, samples finitely many algebraic functions from the ring of algebraic functions on μ, and then evaluates them at $x_i = p_i$ and $y_i = q_i$ for every i.

4.3 Curve Vectorisation

One can turn barcodes into curves, and then extract feature vectors be sampling a curve using a finite subset.

Definition 4.3.1. *The Betti curve $\mu : B \to \mathbb{Z}_{>0}$ is $\beta_\mu : \mathbb{R} \to \mathbb{R}$ given by*

$$t \mapsto \sum_{[p,q] \in B} \mathbf{1}_{p \leq t < q} \mu_{p,q}.$$

The Betti curve simply counts the number of intervals, including multiplicities, in B which contain t. A similarly inspired curve vectorisation is given by the lifespan curve.

Definition 4.3.2. *The lifespan curve of $\mu : B \to \mathbb{Z}_{>0}$ is $L_\mu : \mathbb{R} \to \mathbb{R}$ given by*

$$t \mapsto \sum_{[p,q] \in} \mathbf{1}_{p \leq t < q} \mu_{p,q}(q - p).$$

4.4 Functional Vectorisation

More generally, we can consider extracting feature vectors from maps whose domain is not $\mathbb{R}$. For our example of a functional vectorisation we let $f : \mathbb{R}^2 \to \mathbb{R}_{>0}$ be a piecewise-differentiable function with $f(x, 0) = 0$ for all $x \in \mathbb{R}$ and

$$\Psi := \{\psi_{p,q} : [p, q] \in B\}$$

where $\psi_{p,q}$ is a probability distribution on $\mathbb{R}^2$ with mean $(p, q - p)$.

Definition 4.4.1. *The persistence surface of $\mu : B \to \mathbb{Z}_{>0}$ with respect to f and Ψ is $\rho^\mu_{f,\Psi} : \mathbb{R}^2, \mathbb{R}$ given by*

$$(x, y) \mapsto \sum_{[p,q] \in B} \mu_{p,q} f(p, q - p) \psi_{p,q}(x, y).$$

Moreover, the persistence image $I^\mu_{f,\Psi} : \mathcal{P}(\mathbb{R}) \to \mathbb{R}$ is given by

$$Z \mapsto \iint_Z \rho^\mu_{f,\Psi}(x, y) \, \mathrm{d}x \, \mathrm{d}y.$$

To obtain a feature vector from a persistence image, one lets Z range over grid pixels in a rectangular subset of $\mathbb{R}^2$ and then normalises to produce a greyscale image. Standard choices for f and Ψ include

$$f(x, y) = \begin{cases} 0 & t \leq 0 \\ \frac{t}{\lambda_{\max}} & 0 < t < \lambda_{\max} \\ 1 & t \geq \lambda_{\max} \end{cases}$$

where $\lambda_{\max}$ is the largest persistence, and

$$\psi_{p,q}(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(x-p)^2+(y-(q-p))^2}{2\sigma^2}\right)$$

for some σ.

4.5 Ensemble Vectorisation

Ensemble methods take a large collection of training barcodes to generate a vectorisation mechanism that can then be applied to a separate barcode. More specifically, let $\mu_i : B_i \to \mathbb{Z}_{>0}$ be a collection of training barcodes and fix a dimension parameter $b \in \mathbb{Z}_{>0}$. The automatic topology-oriented learning vectorisation maps each B_i to a vector space in $\mathbb{R}^b$.

1. Let
$$P_i := \big\{(p,q) \in \mathbb{R}^2 : [p,q] \in B_i,\, q > p\big\}$$
define discrete measures on $\mathbb{R}^2$. From which estimate their average measure, E.

2. Let $\mathbf{z} = (z_1, \ldots, z_b)$ be a point sample in $\mathbb{R}^2$ drawn along E. Let
$$\sigma_i(\mathbf{z}) := \frac{1}{2}\max_{j\neq i}\|z_j - z_i\|_2$$
for $1 \leq i \leq b$.

Now construct the contrast functions $\big(\Omega_i : \mathbb{R}^2 \to \mathbb{R}\big)_{1\leq i\leq b}$ where

$$\Omega_i(x) = \exp\left(-\frac{\|x-z_i\|}{\sigma_i(\mathbf{z})})\right).$$

Then the automatic topology-oriented learning vectorisation for a given barcode $\mu : B \to \mathbb{Z}_{>0}$ is $(\Omega_1^\mu, \ldots, \Omega_b^\mu)$ where

$$\Omega_i^\mu := \sum_{[p,q]\in B} \mu_{p,q}\Omega_i(p,q).$$

5 Classifying Distributions on Manifolds

Suppose that we want to classify different probability distributions supported on disjoint manifolds $M_a, M_b \subseteq \mathbb{R}^d$. In particular, we assume that $\inf\{\|x-y\| : x \in M_a,\ y \in M_b\}$ can be made arbitrarily small, but not zero [1]. To proceed we sample a finite set of points $T \subseteq M_a \cup M_b$ uniformly and densely so that the corresponding Betti numbers can be obtained from $T \cap M_a$ and $T \cap M_b$ respectively. Each $x \in T$ is labelled as to whether $x \in M_a$ or $\in M_b$. It is natural to apply a feed-forward neural network $\nu : \mathbb{R}^d \to [0,1]$ to this classification problem. In particular,

$$\nu = s \circ f_l \circ \cdots \circ f_2 \circ f_1.$$

- Layer $f_j : \mathbb{R}^{n_j} \to \mathbb{R}^{n_{j+1}}$, is an affine map $\rho_j : \mathbb{R}^{n_j} \to \mathbb{R}^{n_{j+1}}$ given by $x \mapsto A_j x + b_j$, composed with an activation function $\sigma : \mathbb{R}^{n_j} \to \mathbb{R}^{nj+1}$.
- $s : \mathbb{R}^{n_{l+1}} \to [0,1]$ is a score function. This is a linear classifier that defines a hyperplane in $\mathbb{R}^p$, which is used to determine the output of the network.
- n_j is the number of nodes in the j^{th} layer. In particular, $n_1 = d$ and $n_{l+1} = p$.
- Let $\nu_j := f_j \circ \cdots \circ f_2 \circ f_1$, $\nu = s \circ v_l$ and $\nu_{l+1} := s$. By convention, we take ν_0 to be the identity function. Consequently, $\beta(M) = \beta(\nu(M))$.

We can then train the the l-layer neural network $\nu : \mathbb{R}^d \to [0,1]$ on $T \subset M_a \cup M_b$ to classify samples as a or b. The network is said to be well-trained if it correctly classifies all of $x \in T$. One can imagine, and indeed it is observed in practice, that $M_a \cup M_b$ has complicated topologies.

1. Individually the manifolds M_a and M_b could have complicated topologies.
2. M_a and M_b could individually have simple topologies, but be entangled in a complicated manner. That is, one of the following statements could hold.
   (a) M_a and M_b cannot be separated by a hyperplane.
   (b) Any decision boundary that can separate M_a and M_b will necessarily have a complicated topology.

The neural network is trained to disentangle these manifolds through its layers such that M_a and M_b have relatively simple individual topologies that can be differentiated more robustly with a hyperplane. Indeed, using the tools of homology, one can observe how the topologies of these manifolds progress through the neural network. In particular, one focuses on how the Betti number evolves through the layers, by using Vietoris-Rips complexes at each layer. By understanding the evolution of topologies through neural networks, one can inform architectural design to complement the structure of the raw point cloud data. More specifically, we can find out how parameters such as the number of layers, layer width, and activation functions impact the network's ability to extract certain topological features. This is an intrinsic extraction of topological features using algebraic topology.

6 Enhancing Graphical Methods

It is possible to utilise algebraic topology to inspire machine learning techniques' architectural design, beyond hyper-parameter tuning. The utility of algebraic topology is to work with non-euclidean discrete data domains, and so here we study its application to the graph domain. In particular, we explore the combinatorial complex neural network (CCNN), which is a generalisation of the graph neural networks (GNN) [4]. CCNN utilises topology to extrapolate graphs into a higher-order representation, facilitating the abstraction of robust features, and understanding the higher-level patterns of the data. More specifically, for graph-based data, the higher-order representations allow for higher-order interactions to be performed, which is particularly powerful for performing intra-graph tasks.

6.1 Combinatorial Complex Neural Network

We can form a combinatorial complex on a set of points, with the corresponding cochain maps, to extract higher-order relationships to help perform inference on the set of points. For instance, with the combinatorial complex, one can perform cell classification, complex classification, and cell prediction [4]. A combinatorial complex neural network is a class of neural network architectures that are supported on combinatorial complexes.

Definition 6.1.1. *Let K be a combinatorial complex. Let $C_{i_1} \times \cdots \times C_{i_m}$ and $C_{j_1} \times \cdots \times C_{j_n}$ be Cartesian products of cochain spaces on K. Then a combinatorial complex neural network is a function of the form*

$$\mathrm{CCNN} : C_{i_1} \times \ldots C_{i_m} \to C_{j_1} \times C_{j_n}.$$

Remark 6.1.2. *A combinatorial complex neural network according to Definition 6.1.1 takes a vector of cochains and returns another vector of cochains. One can see how this can be generally attributed to a message-passing or pooling mechanism on a combinatorial complex.*

We can construct the precise mapping executed by a combinatorial complex neural network using tensor diagrams. More specifically, we can think of the cochain spaces as nodes, and consider different operations between these nodes. Stacking these operations in a consequential manner will build a mapping between Cartesian products of cochain spaces and result in a combinatorial complex neural network. Indeed, the operations between nodes are inspired by wanting to perform cell classification, complex classification and cell prediction as we detailed previously. To construct these operations between nodes we need a push-forward operation.

Definition 6.1.3. *Let K be a combinatorial complex. Let $G : C_i(K) \to C_j(K)$ be a cochain map, and let $\mathbf{H}_i$ be in $C_i(K)$. Then the cochain push-forward induced by G is $\mathcal{F}_G : C_i(K) \to C_j(K)$ is given by*

$$\mathbf{H}_i \to \mathbf{K}_i := \left(\mathbf{k}_{y_1^j}, \ldots, \mathbf{k}_{y^j_{\left|K^{(j)}\right|}} \right)$$

such that for $k = 1, \ldots, \left|K^{(j)}\right|$ we have

$$\mathbf{k}_{y_k^j} = \bigoplus_{x_l^i \in \mathcal{N}_{G^T}\left(y_k^j\right)} \alpha_G \left(\mathbf{h}_{k_l^i} \right)$$

with $\bigoplus$ being a permutation-invariant aggregation function and α_G a differentiable function.

Remark 6.1.4. *The operator $\mathcal{F}_G$ pushes forward an i-cochain supported on $K^{(i)}$ to a j-cochain supported on $K^{(j)}$, by aggregating the feature vectors in the neighbourhood of each cell.*

6.1.1 Nodes

With the push-forward mechanism, we can construct operations between nodes, that is operations between cochain spaces.

Definition 6.1.5. *Let K be a combinatorial complex. Let $G_1 : C_{i_1}(K) \to C_j(K)$ and $G_2 : C_{i_2}(K) \to C_j(K)$ be cochain maps. A merge node $\mathcal{M}_{G_1,G_2} : C_{i_1} \times C_{i_2} \to C_j$ is given by*

$$(\mathbf{H}_{i_1}, \mathbf{H}_{i_2}) \mapsto \beta\left(\mathcal{F}_{G_1}(\mathbf{H}_{i_1}) \bigotimes \mathcal{F}_{G_2}(\mathbf{H}_{i_2})\right)$$

where $\bigotimes : C_j \times C_j \to C_j$ is an aggregation function, $\mathcal{F}_{G_1}$ and $\mathcal{F}_{G_2}$ are the induced push-forward operators as given by Definition 6.1.3 and β is an activation function.

Definition 6.1.6. *Let K be a combinatorial complex. Let $G_1 : C_j(K) \to C_{i_1}(K)$ and $G_2 : C_j(K) \to C_{i_2}(K)$ be cochain maps. A split node $\mathcal{S}_{G_1,G_2} : C_j \to C_{i_1} \times C_{i_2}$ is given by*

$$\mathbf{H}_j \mapsto (\beta_1(\mathcal{F}_{G_1}(\mathbf{H}_j)), \beta_2(\mathcal{F}_{G_2}(\mathbf{H}_j)))$$

where for $i = 1, 2$ the operator $\mathcal{F}_{G_i}$ is the induced push-forward operator as given by Definition 6.1.3 and β_i is an activation function.

Remark 6.1.7. *The push-forward, merge node, and split node are referred to collectively as elementary tensor operations.*

Beyond the elementary tensor operators, there exist convolutional and attention-based operators. To consider such operators requires augmenting Definition 6.1.3. Doing so [4] can define combinatorial complex convolutional neural networks and combinatorial complex attention neural networks, that lift more sophisticated machine learning techniques into the domain of combinatorial complex data.

References

[1] Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. *Topology of deep neural networks*. Apr. 13, 2020. DOI: 10.48550/arXiv.2004.06093. arXiv: 2004.06093[cs,math,stat]. URL: http://arxiv.org/abs/2004.06093 (visited on 12/31/2023).

[2] Felix Hensel, Michael Moor, and Bastian Rieck. "A Survey of Topological Machine Learning Methods". In: *Frontiers in Artificial Intelligence* 4 (2021). ISSN: 2624-8212. URL: https://www.frontiersin.org/articles/10.3389/frai.2021.681108 (visited on 12/31/2023).

[3] Dashti Ali, Aras Asaad, Maria-Jose Jimenez, Vidit Nanda, Eduardo Paluzo-Hidalgo, and Manuel Soriano-Trigueros. *A Survey of Vectorization Methods in Topological Data Analysis*. Dec. 19, 2022. DOI: 10.48550/arXiv.2212.09703. arXiv: 2212.09703[math]. URL: http://arxiv.org/abs/2212.09703 (visited on 12/31/2023).

[4] Mustafa Hajij, Ghada Zamzmi, Theodore Papamarkou, Nina Miolane, Aldo Guzmán-Sáenz, Karthikeyan Natesan Ramamurthy, Tolga Birdal, Tamal K. Dey, Soham Mukherjee, Shreyas N. Samaga, Neal Livesay, Robin Walters, Paul Rosen, and Michael T. Schaub. *Topological Deep Learning: Going Beyond Graph Data*. May 19, 2023. DOI: 10.48550/arXiv.2206.00606. arXiv: 2206.00606[cs,math,stat]. URL: http://arxiv.org/abs/2206.00606 (visited on 12/31/2023).