Data-Centric AI Requires Rethinking Data Notion

Mustafa Hajij*
,Santa Clara University
mhajij@scu.edu

Ghada Zamzmi*
University of South Florida
ghadh@mail.usf.edu

Karthikeyan Natesan Ramamurthy
IBM Research
knatesa@us.ibm.com

Aldo Guzman Saenz
IBM Research
Aldo.Guzman.Saenz@ibm.com

Abstract

The transition towards data-centric AI requires revisiting data notions from mathematical and implementational standpoints to obtain unified data-centric machine learning packages. Towards this end, we propose unifying principles offered by categorical and cochain notions of data, and discuss the importance of these principles in data-centric AI transition. In the categorical notion, data is viewed as a mathematical structure that we act upon via morphisms to preserve this structure. As for cochain notion, data can be viewed as a function defined in a discrete domain of interest and acted upon via operators. While these notions are almost orthogonal, they provide a unifying definition to view data, ultimately impacting the way machine learning packages are developed, implemented, and utilized by practitioners.

1 Introduction

The common notion of data, which is overwhelmingly preferred by machine learning practitioners, is rooted in statistical science [9]. Within this notion, data is described as a sample obtained from a probability density function (PDF) defined over a domain of interest[1]. Consider, for instance, the dimensionality reduction problem in which one is given a data $X \subset \mathcal{X}$, and the aim is to find a mapping $f : \mathcal{X} \to \mathcal{Y}$ such that some mathematical structure related to X is preserved via f; $\mathcal{X}$ and $\mathcal{Y}$ denote the data space and the target space, respectively. While the data X is typically sampled from an unknown probability distribution, from the perspective of the dimensionality reduction algorithm, the interesting aspect of data lies in the mathematical structure the function f is designed to preserve.

The past decade has witnessed an enormous interest in the mathematical aspects of machine learning models, their properties, limitations, and generalizations. As the interest in the machine learning community is shifting towards a data-centric AI, we believe that the commonly used notion of data must be carefully revisited and systematically studied. Systematic and cohesive mathematical definition of data notions can impact how data is understood and how machine learning packages are implemented. Also, it can ultimately expand the number of practitioners, who can potentially benefit from emerging machine learning technologies. As an example, we give the analogy of the popular scikit-learn package [21], which has a simple API, yet unified across all underlying algorithms.

To provide a concrete definition of data, we distinguish between two main notions: *categorical* and *cochain*. In the *categorical notion* (Section 2), the *invariants* of data are often the interesting object of study. Invariants of data are the structural properties of interest that are preserved under a particular type of *morphisms* (structure-preserving maps). For instance, one invariant can be the metric space structure, encoded in the distance matrix or the persistence diagrams, that is invariant under metric preserving functions (morphism). In the *cochain notion* (Section 3), the domain upon which the data is defined is the main element of interest. Within this perspective, we think of data

---

* indicates joint authorship.

[1]There are several others notions rooted in physical and social sciences that we do not focus on here.

as a function defined on the elementary building blocks of the domain of interest; this function is acted upon by operators defined on these domains. An example of this type is the solution of a partial differential equation on a surface. While these two notions are almost orthogonal, they provide a unifying definition to view the data, ultimately impacting the way machine learning packages are designed, implemented, and utilized by various practitioners.

2 Categorical Notion

Here, we consider data as a mathematical structure (e.g., a metric space) that we act upon via *morphisms* to preserve this structure. Specifically, the categorical setting seeks *invariants* in the data, a well-defined mathematical quantity extracted from the data and is preserved under a particular type of morphisms. In practical settings, a loss function is typically utilized to preserve the structure as much as possible; however, completely preserving the mathematical structure might not be feasible. In what follows, we provide examples of mathematical structures along with our categorical definitions.

2.1 Data as a Metric Space

In the multi-dimensional scaling algorithm (MDS) [4], the data $X \subset \mathcal{X}$ is given as a finite metric space (X, d_X). The purpose of the MDS algorithm is to construct a mapping $f : \mathcal{X} \to \mathcal{Y}$ that preserves the metric structure defined on X as much as possible. Specifically, we define a cost function:

$\mathcal{L}_{MDS} : Obj(\mathrm{Mat}) \times Obj(\mathrm{Mat}) \to \mathbb{R}$ defined via :

$$\mathcal{L}_{MDS}(((X, d_X), (f(X), d_Y))) = \sum_{(x_1, x_2) \in X \times X} l_{MDS}(d_X(x_1, x_2)), d_Y(f(x_1), f(x_2)) \quad (1)$$

where Mat denotes the category[2] of metric spaces, $Obj(\mathrm{Mat})$ denotes the objects of Mat, and $l_{MDS} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is an appropriate loss function (e.g., absolute value). In simple terms, the above cost function defines an approximate notion of equivalence in the category of metric spaces.

2.2 Data as a Probabilistic Structure

T-SNE algorithm [15] considers a matrix P that encodes a *probabilistic structure* extracted from the data X; i.e., given a point cloud $X \subset \mathcal{X}$, the entry P_{ij} encodes the similarity of points x_i and x_j in the data X. For brevity, we omit the precise definition of the matrix P. For T-SNE, the structure of interest to be preserved when mapping this data to another space is encoded in the matrix P.

The algorithm seeks morphisms that (approximately) preserve this structure, hence achieving approximate invariance. We realize the T-SNE algorithm in our weak categorical setting. Namely, the objects are tuples of the form (X, P_X), where $X \subset \mathcal{X}$ is a finite set and P_X is a symmetric matrix such that the summation of its entries add up to 1. The morphisms are functions of the form $f : \mathcal{X} \to \mathcal{Y}$ that preserves the probabilistic structure of X as much as possible. We define a cost function:

$\mathcal{L}_{TSNE} : Obj(\mathrm{Prob}) \times Obj(\mathrm{Prob}) \to \mathbb{R}$ defined via :

$$\mathcal{L}_{TSNE}(((X, P_X), (f(X), P_{f(X)}))) = KL(P_X \| P_{f(X)}) \quad (2)$$

where $KL(A \| B)$ is the KL divergence between two probability distributions A and B.

2.3 Data as a Homological Structure

Given a point cloud $X \subset \mathcal{X}$, persistent homology [8] provides a mechanism to construct a finite sequence of topological fingerprints of X, which we denote by $\mathrm{PH}_X := (\mathbf{D}_X = \{D_1, D_2, ...\}, \boldsymbol{\pi}_X = \{\pi_1, \pi_2, ...\})$. Here, D_i are the *persistence pairings* and π_i are the *persistence diagrams* induced by the Vietoris–Rips of (X, d_X) [3]. These topological fingerprints encode a structure about the data that is relevant in many machine learning-related tasks such as classification and dimensionality reduction [19, 17]. Consider the scenario where we construct a mapping $f : \mathcal{X} \to \mathcal{Y}$ such that f preserves

---

[2]A category is a collection of 'objects' (e.g., metric spaces and sets) that are linked by 'arrows' (the maps between these spaces) such that (1) arrows are composable in the same way we compose functions and (2) every object has an identity arrow.

[3]We only discuss persistence diagrams induced by the Vietoris–Rips filtrations, in such case the persistence computation depends only on the distance matrix d_X. The treatment for general filtration is essentially similar.

these fingerprints as much as possible. Preserving this structure is equivalent to preserving particular entries in the distance matrices d_X and d_Y, and these entries depend on the persistence pairings π [8]. The weak equivalence is a function of the form $\mathcal{L}_{PH} : Obj(Per) \times Obj(Per) \to \mathbb{R}$ defined by

$$\mathcal{L}_{PH}(\mathrm{PH}_X, \mathrm{PH}_{f(X)}) = d(\mathrm{PH}_X, \mathrm{PH}_{f(X)}) \tag{3}$$

where d is some appropriately chosen distance between the persistence structures of X and $f(X)$ (e.g., the Sinkhorn distance) [10]. See [17] for recent works on preserving the persistence diagram.

3 Cochain Notion

Orthogonal to the categorical notion, data can be viewed as a function defined in a discrete domain of interest and acted upon via *operators*. Consider, for instance, a discrete domain $\mathcal{M}$, say a graph or a simplicial complex [14]. Roughly speaking, data in this case is defined in the form $f : \mathcal{M} \to \mathcal{X}$, where f associates to every building block $v \in \mathcal{M}$ a data element $f(v) \in \mathcal{X}$; e.g., a scalar or a vector in some Euclidean space. An important example of this setting is Topological Data Analysis (TDA) [8], where one studies the topological properties of the domain $\mathcal{M}$ by studying the properties of functions defined in it. Another example is the Graph Signal Processing (GSP) [20], where one is interested in the scalar signal defined on the node or the edge sets of a given graph. Similar to the categorical notion, cochain notion provides an abstraction that comes with important virtues: (1) it unifies notions in a single cohesive and mathematically elegant perspective, (2) it becomes evident how to create new similar structures, and (3) it impacts the way machine learning packages are designed and implemented. In what follows, we restrict our work to simplicial complexes and assume these complexes are oriented and finite. Moreover, we use $\mathcal{C}_k(\mathcal{M})$ to denote the space of all linear combinations of all the k-simplices in $\mathcal{M}$; elements in $\mathcal{C}_k(\mathcal{M})$ are called chains. Dually, $\mathcal{C}^k(\mathcal{M})$ is the vector space spanned by all real-valued functions defined on k-simplices, and its elements are called cochains.

3.1 Deep Learning on Complexes and Discrete Exterior Calculus

We argue that the cochain perspective provides a unified framework between two subjects, namely topological deep learning on complexes [12, 7, 3] and discrete exterior calculus (DEC) on complexes [16, 5, 22], that are typically treated separately (see Section 3.1.2).

3.1.1 Discrete exterior calculus

In the context of discrete exterior calculus, a linear operator $\mathcal{A} : \mathcal{C}^i(\mathcal{M}) \to \mathcal{C}^j(\mathcal{M})$, may act on a cochain f to produce another cochain $\mathcal{A}(f)$:

$$f \xrightarrow{\mathcal{A}} \mathcal{A}(f) \tag{4}$$

An example of an operator $\mathcal{A}$ is the graph Laplacian. Namely, given a graph $G(V,E)$, the graph Laplacian is a matrix $L = A - D$, where A is the adjacency matrix of the graph and D is the degree matrix. As an operator, the graph Laplacian $L : \mathcal{C}^0 \to \mathcal{C}^0$ takes a signal $f : V \to \mathbb{R}$, a scalar function defined on the node set graph G, or an element in $\in \mathcal{C}^0$, and produces another signal $L(f) \in \mathcal{C}^0$.

The (primitive) discrete exterior operators are exactly seven operators that include the exterior derivative (coboundary map), the hodge star, and the wedge product. We refer the reader to [6] for a complete list of these operators. In addition to these operators, there are other complex operators that can be built from them. These operators are the analog of the classical differential operators such as the gradient, divergence, and curl operators. Together cochains and the operators that act on them allow a concrete framework that facilitates computing a cochain of interest such as a cochain obtained by solving a partial differential equation on a discrete surface. To keep our treatment concrete, more examples are provided.

Example 3.1. *One of the most primitive operators is the (co)boundary maps. Consider a simplicial complex surface $\mathcal{M}$, and denote its sets of nodes, edges and faces by V,E,F. The boundary operator $\partial_k : \mathcal{C}_k \to \mathcal{C}_{k-1}$ is defined to be $\partial_k \{v_0 v_1,...,v_k\} = \sum_{i=1}^{k} (-1)^k \{v_0,...,\hat{v}_i,...,v_k\}$, where $\hat{v}_i$ indicates that v_i is missing from the sequence that determines the simplex. Dually, the coboundary maps are defined to be the transpose ∂_k^T of the boundary maps. The coboundary map ∂_k^T is also referred to by the kth discrete exterior derivative and is denoted by d^k. In particular, $d^0(f)$, $d^1(f)$, and $d^2(f)$ of some 0, 1, 2 cochains defined on $\mathcal{M}$ are the discrete analogs of the gradient ∇f, the curl $\nabla \times f$, and divergence $\nabla \cdot f$ of a smooth function defined on a smooth surface.*

Example 3.2. *The (co)boundary can be used to define the k-Hodge Laplacian $L_k = \partial_k^\top \partial_k + \partial_{k+1} \partial_{k+1}^\top$:. As an operator, the $k-$ Hodge Laplacian maps a k-cochain to another k-cochain.*

3.1.2 Topological Nets: deep learning on complexes and a unifying perspective

To show that the cochain notion provides a unified framework between topological deep learning on complexes and DEC on complexes, we introduce topological networks (TNs).

Let f be a cochain defined on a finite complex $\mathcal{M}$. We denote by $\mathcal{C}(\mathcal{M})$ to the $\mathbb{R}$-vector space of all linear combinations of such maps. Given an ordering on the cells in $\mathcal{M}$, these cells can be identified with the canonical orthonormal basis in $\mathbb{R}^{|\mathcal{M}|}$. We assume this identification in what follows. A linear operator $\mathcal{A} : \mathcal{C}(\mathcal{M}) \to \mathcal{C}(\mathcal{M})$ induces a *topological network* $TN_\mathcal{A} : \mathcal{C}(\mathcal{M}) \to \mathcal{C}(\mathcal{M})$ defined via

$$f \xrightarrow{TN_\mathcal{A}} \phi(\mathcal{A}fW) \tag{5}$$

where W is a trainable weight matrix and ϕ is a non-linearity. The Topological Networks (TNs) given in equation (5) are deceptively simple yet they are the essence of our general unifying framework. Observe first that equation (5) generalizes the familiar convolutional graph neural network [18]. Moreover, it is easy to show that the setting given in equation (5) is equivalent to a message passing protocol (determined by the matrix $\mathcal{A}$). See [11] for a proof in a graph neural network, which easily carries over to higher order complexes such as cell and simplicial complexes. More importantly, we view equation (5) from the exterior calculus perspective. Specifically, we emphasize the operator and cochain natures of both $\mathcal{A}$ and f respectively since this perspective allows for a unified treatment of deep learning on complexes with exterior calculus. In particular, when the matrix W is the (untrainable) identity matrix and the map ϕ is the identity map then equation (5) is equivalent to the linear setting given in equation (4) and discussed in Section 3.1.1. In other words, DEC which is generally summarized by operations of the form given via equation (5) corresponds precisely to a message function in the deep learning setting without performing the update function.

This unified framework offers multiple advantages. First, it is designed to express and train mathematical expressions that combine linear and non-linear forms on complexes. For instance, it makes sense to solve, or train, an expression of the form $d_1(TN_{d_0}(f)) = L_2(g)$, where f a cochain in $\mathcal{C}^0$ and g is a given cochain in $\mathcal{C}^2$. Second, this form generalizes (with the right choice of the matrix $\mathcal{A}$) all existing message passing protocols defined on complexes including [12, 23, 26, 25, 1, 7], and hence offering a unifying framework to write deep learning on graphs, surfaces, and higher order complexes. Note that we are defining $\mathcal{A}$ to operate on a cochain defined on the entire complex $\mathcal{M}$. This means that the message passing can occur among simplices of different dimensions simultaneously. In practice, the operator $\mathcal{A}$ is a block matrix whose block elements are the discrete operators we introduced in Section 3.1.1 or operators that can be built from them (e.g. the exterior derivative, the Laplacian, etc.). Third, from a practical standpoint, this setting creates a concrete mathematical foundation for a unifying deep learning API on complexes as well as DEC. This can ultimately invite practitioners from both fields to push the state-of-the-arts in new directions. With the right abstraction of data, we envision a unifying API across multiple algorithms and disciplines related to data science.

4 Related Works

Spivak et al. [27] introduced a categorical framework for knowledge representation. Another effort that focuses on creating concrete foundations for geometric deep learning is proposed by Bronstein et al. [2]. Our work approaches the notion of data in general terms instead of focusing on deep models in particular. Moreover, our work emphasizes the role re-examining data notion plays in understanding models from a theoretical and practical standpoints. Recent works on non-linear graph signal processing and its generalization to higher order complexes [23, 12, 13, 1, 26, 24] implicitly discuss the cochain notion of data. In [26], Schaub et al. emphasized the relationship between linear signal processing on complex (determined by linear operators) and non-linear filtering defined via a neural network on higher order networks. The goal of these previous works is to study signal processing on higher order complexes. Our goal here is to emphasize the unifying principles offered by the categorical and cochain notions and its importance in the data-centric AI transition. In particular, we emphasize the theoretical advantages, practical ramifications, and the evolution of general understanding of these principles.

5   Acknowledgement

References

[1] C. Bick, E. Gross, H. A. Harrington, and M. T. Schaub. What are higher-order networks? *arXiv preprint arXiv:2104.11329*, 2021.

[2] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[3] E. Bunch, Q. You, G. Fung, and V. Singh. Simplicial 2-complex convolutional neural nets. *NeurIPS workshop on Topological Data Analysis and Beyond*, 2020.

[4] M. A. Cox and T. F. Cox. Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer, 2008.

[5] F. De Goes, A. Butts, and M. Desbrun. Discrete differential operators on polygonal meshes. *ACM Transactions on Graphics (TOG)*, 39(4):110–1, 2020.

[6] M. Desbrun, E. Kanso, and Y. Tong. Discrete differential forms for computational modeling. In *Discrete differential geometry*, pages 287–324. Springer, 2008.

[7] S. Ebli, M. Defferrard, and G. Spreemann. Simplicial neural networks. *NeurIPS 2020 Workshop TDA and Beyond*, 2020.

[8] H. Edelsbrunner and J. Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

[9] J. Friedman, T. Hastie, R. Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[10] C. Frogner, C. Zhang, H. Mobahi, M. Araya-Polo, and T. Poggio. Learning with a wasserstein loss. *arXiv preprint arXiv:1506.05439*, 2015.

[11] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[12] M. Hajij, K. Istvan, and G. Zamzmi. Cell complex neural networks. *NeurIPS 2020 Workshop TDA and Beyond*, 2020.

[13] M. Hajij, G. Zamzmi, and X. Cai. Simplicial complex representation learning. *arXiv preprint arXiv:2103.04046*, 2021.

[14] A. Hatcher. *Algebraic topology*. 2005.

[15] G. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *NIPS*, volume 15, pages 833–840. Citeseer, 2002.

[16] A. N. Hirani. *Discrete exterior calculus*. California Institute of Technology, 2003.

[17] O. Kachan. Persistent homology-based projection pursuit. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 856–857, 2020.

[18] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[19] M. Moor, M. Horn, B. Rieck, and K. Borgwardt. Topological autoencoders. In *International conference on machine learning*, pages 7045–7054. PMLR, 2020.

[20] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[22] L. Ptáčková and L. Velho. A simple and complete discrete exterior calculus on general polygonal meshes. *Computer Aided Geometric Design*, 88:102002, 2021.

[23] T. M. Roddenberry, N. Glaze, and S. Segarra. Principled simplicial neural networks for trajectory prediction. In *International Conference on Machine Learning*, pages 9020–9029. PMLR, 2021.

[24] T. M. Roddenberry and S. Segarra. Hodgenet: Graph neural networks for edge data. In *2019 53rd Asilomar Conference on Signals, Systems, and Computers*, pages 220–224. IEEE, 2019.

[25] M. T. Schaub, A. R. Benson, P. Horn, G. Lippner, and A. Jadbabaie. Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Review*, 62(2):353–391, 2020.

[26] M. T. Schaub, Y. Zhu, J.-B. Seby, T. M. Roddenberry, and S. Segarra. Signal processing on higher-order networks: Livin' on the edge... and beyond. *Signal Processing*, 187:108149, 2021.

[27] D. I. Spivak and R. E. Kent. Ologs: a categorical framework for knowledge representation. *PloS one*, 7(1):e24274, 2012.