Simplicial Neural Networks

David S. Hippocampus[*]
Department of Computer Science
Cranberry-Lemon University
Pittsburgh, PA 15213
hippo@cs.cranberry-lemon.edu

Abstract

We generalize graph convolutional neural networks to data that live on a class of topological spaces called simplicial complexes. These are natural multi-dimensional generalizations of graphs which encode more than just pairwise relationships, namely higher-order relationships between nodes (represented geometrically as filled triangles, tetrahedra, and so forth). We define an appropriate notion of convolutions for such data, which we leverage to construct the desired convolutional neural networks. This allows us to consider richer data than traditional methods, including n-fold collaboration networks and vector field data.

1 Introduction

- Graph Neural Networks intro
- Extension to simplicial complexes.
- Simplicial complex represent collaboration on which we have data like citations.
- Preliminary results on vector field data. Where we believe this method can be applied and work?
- Cite work on hypergraphs, message passing.
- Advantage our work: low computational cost, Laplacian an operator encodes topological information has been extensively used in other application eg page ranking. In higher dimension we believe it carries additional information on the simplicial complex.

2 Proposed Technique

FIXME

2.1 Simplicial Complexes

A *simplicial complex* is a collection of finite sets closed under taking subsets. We refer to a set in a simplicial complex as a *simplex* of *dimension* p if it has cardinality $p+1$. Such a p-simplex has $p+1$ *faces* of dimension $p-1$, namely the sets omitting one element, which we will denote as $[v_0, \dots, \hat{v}_i, \dots, v_p]$ when omitting the i'th element. While this definition is entirely combinatorial, we will soon see that there is a geometric interpretation, and it will make sense to refer to and think of 0-simplices as *vertices*, 1-simplices as *edges*, 2-simplices as *triangles*, 3-simplices as *tetrahedra*, and so forth (see Figure 1).

[*]Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

the Topological Data Analysis and Beyond Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Let $C_p(K)$ be the free real vector space with basis K_p, the set of p-simplices in a simplicial complex K. The elements of $C_p(K)$ are called *p-chains*. The p-cochain (vector) space $C^p(K)$ is defined as the dual of $C_p(K)$, i.e. $C^p(K) = \{\hom(K, \mathbb{R})\}$. The basis of C^p is given by the dual basis K_p^*. These vector spaces come equipped with *coboundary maps*, namely linear maps defined by

$$\delta_p : C_p \to C_{p+1}$$

$$\delta_p(f([v_0, \dots, v_{p+1}])) = \sum_{i=0}^{p+1} (-1)^i f([v_0, \dots, \hat{v}_i, \dots, v_{p+1}])$$

where $\hat{v}_i$ denotes that the i-th vertex has been omitted.

[[**Stefania says:** Change above with definition of cochain. Cochain in in a grid are pixels. Any diemnsion are the features of our simplices]]

Papers	Authors	Citations
Paper I	A, B, C	100
Paper II	A, B	50
Paper III	C, D	4

Table 1: Data

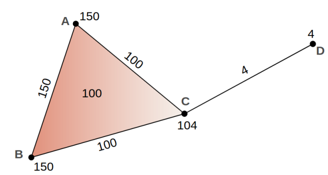

Figure 1: Co-authorship complex

Figure 2: Example of co-authorship complex build from data

2.2 Combinatorial Laplacians

We are in this paper concerned with finite simplicial complexes, and assume that they are built in a way that encodes useful information about the data being studied. In particular, it is not necessary for it to come equipped with some embedding into Euclidean space, nor do we demand that it triangulates a Riemannian manifold. Therefore dualities like the Hodge star, which is used to construct the Hodge–de Rham Laplacian in the smooth setting [2] that motivates us, are unavailable for our method. The same is true for discrete versions of the Hodge star, such as that of Hirani [1]. To define a discrete version of the Laplacian for simplicial complexes, we simply take the linear adjoint of the coboundary operator with respect to the inner product, defining $\delta_i^* : C^{i+1} \to C_i$ by

$$\langle \delta_i f_1, f_2 \rangle_{i+1} = \langle f_1, \delta_i^* f_2 \rangle_i \quad \forall f_1 \in C^i(K), f_2 \in C^{i+1}(K).$$

In analogy with Hodge–de Rham theory, we then define the *degree-i simplicial Laplacian* of a simplicial complex K as the linear operator $\mathcal{L}_i : C^i(K) \to C_i(K)$ such that

$$\mathcal{L}_i = \mathcal{L}_i^{\text{up}} + \mathcal{L}_i^{\text{down}}$$

$$\mathcal{L}_i^{\text{up}} = \delta_i^* \circ \delta_i : C^i(K) \to C^i(K)$$

$$\mathcal{L}_i^{\text{down}} = \delta_{i-1} \circ \delta_{i-1}^* : C^i(K) \to C^i(K).$$

[[**Stefania says:** Write with co-boundary]] In case $i = 0$, $\mathcal{L}_0$ corresponds to the classical graph Laplacian. Observe that there are p Laplacians for a complex of dimension p. In most practical applications, the matrices for the Laplacians are very sparse and can easily be computed as a product of sparse boundary matrices and their transposes.

[[**Stefania says:** Laplacian encodes important topological information (just add references)]] [[**Stefania says:** Laplacian can be seen as a message passing function on the simplices]]

Our contribution is a notion of convolution for simplicial complexes using the Laplacian.

2.3 Simplicial Neural Networks

- Goal: building a convolutional NN whose input is an arbitrary p-cochain on a fixed simplicial ocmplex K
- Definition of Fourier Transform

Table 2: Number of simplices

Dimension	0	1	2	3	4	5	6	7	8	9	10
CC1	352	1474	3285	5019	5559	4547	2732	1175	343	61	5
CC2	1126	5059	11840	18822	21472	17896	10847	4673	1357	238	19

- Convolutional Filters: low degree polynomials in the frequency domain
- Implemented using Chebyshev polynomials
- Computational cost: good the p-Laplacian is localized and sparse.

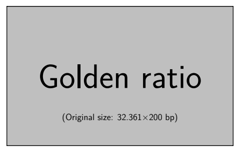

Figure 3: Maybe figures SNN

3 Experimental results

In this section we present experimental results for the simplicial neural network. The datasets we analyze have been extracted from the Semantic Scholar datasets. The data consists of XXX papers together with their authors and number of citations. We retain paper with more than 5 citations and at most 10 authors.

An important step in preprocessing many kinds of input data in TDA is constructing a simplicial complex. Our work focus on *collaboration complexes* (or *co-authorship complexes*) [[**Stefania says:** cite paper Alice]], simplicial complexes where a paper with k authors is represented by a $(k-1)$-simplex. We constructed different co-authorship complexes by considering sub-samplings from the papers set of the Semantic Scholar dataset. The sub-samplings were obtained by performing random walks on the nodes of the graph which vertices corresponds to the papers and edges connect papers sharing at least one author. The co-authorship complexes obtained from each sub-sampling have corresponding k co-chain given by the number of shared citations.

We evaluate the performance of the SNN on the task of predicting missed input data. Specifically, given a fixed co-authorship complex missing data is introduced at random on the training co-chains at 4 levels: $10\%, 20\%, 30\%$, and 50%. As in a typical pipeline for this task, in our approach missing data is first replaced by some values. In our case the training input is given by the citations on the co-authorship complex where the random missing data is substituted by the median of the known data. We trained a SNN composed by 3-layers with 30 convolutional filters of degree 5. We used the L_1 norm as reconstruction loss over the known elements an the Adam optimizer with learning rate of 1×10^{-3}. The SNN was trained for 1000 iterations. We then test the performance of the network on its accuracy in predicting the missing data. A predicted citation is considered correct if the predicted value differs of at most 1 from the actual number of citations. The prediction error is the absolute value of the difference between the predicted citation and the actual value of the citation.

Figure 4 shows the accuracy of the SNN in prediction missing citations on CC1 (Co-authorship Complex 1, Table **??**). The distribution of the prediction error is shown in Figure 5.

As a second assessment of our network we use transfer learning . In particular, we test how accurately a SNN pretrained on a co-authorship complex can predict citations on a different complex. Figure shows the test accuracy on predicting missing values of the SNN on CC2 (Co-authorship Complex 2, see Table) and evaluate in predicting the missing values of CC1.

[[**Stefania says:** Say computations are done up to dimension 3]]

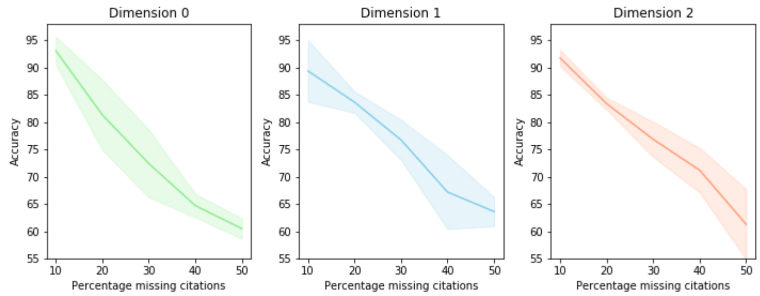

Figure 4: Accuracy of SNN in predicting missing citations

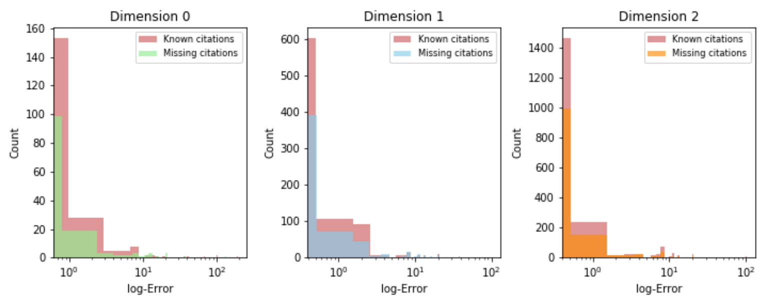

Figure 5: Distribution of the prediction's error

4 Conlusion and Future Work

FIXME

- Define pooling
- Better baseline for the above work
- Preliminary results on vector field data. Where we believe this method can be applied?
- transfer learning accuracy is it linked to similar structure of complexes

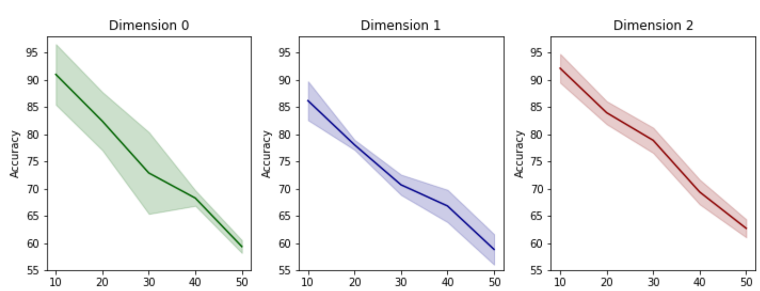

Figure 6: Accuracy in predicting missing citations with a pretrained SNN

References

[1] Anil N. Hirani. "Discrete Exterior Calculus". PhD thesis. California Insitute of Technology, 2003.

[2] Ib Madsen and Jørgen Tornehave. *From calculus to cohomology: de Rham cohomology and characteristic classes*. Cambridge University Press, 1997.