PERSISTENT HOMOLOGY AND GRAPHS REPRESENTATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

This article aims to study the topological invariant properties encoded in node graph representational embeddings by utilizing tools available in persistent homology. Specifically, given a node embedding representation algorithm, we consider the case when these embeddings are real-valued. By viewing these embeddings as scalar functions on a domain of interest, we can utilize the tools available in persistent homology to study the topological information encoded in these representations. Our construction effectively defines a unique persistence-based graph descriptor, on both the graph and node levels, for every node representation algorithm. To demonstrate the effectiveness of the proposed method, we study the topological descriptors induced by DeepWalk, Node2Vec and Diff2Vec.

1 INTRODUCTION

Graphs are among the most ubiquitous models in computer science with an array of applications in drug discovery (Takigawa & Mamitsuka (2013)), biological protein-protein networks (Sun et al. (2017)), data representation and organization (Gross & Yellen (2005)), recommendation systems (Kutty et al. (2014)), and social networks (Nettleton (2013)). Due to the necessity of operating a typical machine learning pipeline with Euclidean data, there has been a surge of interest in learning graph representation. Graph representation aims to learn a map that embeds the nodes of a graph, or subgraphs of it, to some Euclidean space such that this embedding captures some useful information about the graph. In particular, *node representation learning* (Zhang et al. (2018)) has been mostly inspired by the success of Word2Vec (Mikolov et al. (2013)), a method to embed a corpus of words into the Euclidean space based on their context in sentences. Similar to Word2Vec, node representation embeddings learn a latent space representation of the nodes of a given graph and are very useful when performing downstream machine learning tasks on graphs. This includes node and graph classification, community detection, link prediction, and graph similarity (Chen et al. (2020)). Despite the practical popularity of node graph representations, their expressive and discriminative power has not been explored extensively.

This article aims to understand and study the information encoded in node representation embeddings. Specifically, we aim to understand the expressive and discriminative powers as well as the limitations of these embeddings from the perspective of graph classification. Our proposed method utilizes persistent homology (Edelsbrunner & Harer (2010)) to extract the topological invariants from a given latent representation; i.e., for a given node representational embedding $f : V_G \to \mathbb{R}^d$, we are interested in the case when the dimension d of the latent space $\mathbb{R}^d$ is one. In particular, we consider the lower-star filtration obtained from such scalar functions and utilize its induced invariant of G, the so called persistence diagrams, obtained via this filtration. The final persistence-based representation that we obtain is a function of the chosen node-embedding algorithm and the input graph.

Recent years have witnessed an increased interest in the role topology plays in machine learning and data science (Carlsson (2009)). Topology provides a set of natural tools that allows the formulation of many long-standing problems in these fields. In particular, persistent homology (Edelsbrunner & Harer (2010)) has been successful at finding solutions to a large array of complex problems (Attene et al. (2003); Bajaj et al. (1997); Kweon & Kanade (1994); Lee et al. (2011a;b); Lum et al. (2013); Rosen et al. (2017)). These sets of tools have matured and become an area of research known as *Topological Data Analysis* (TDA) (Edelsbrunner & Harer (2010); Carlsson (2009)). TDA-based

methods have shown excellent performance in several applications including neuroscience (Lee et al. (2012)), bioscience (Chan et al. (2013); DeWoskin et al. (2010); Hajij et al. (2020b)), in the study of graphs (Petri et al. (2013a;b)), time-varying data (Edelsbrunner et al. (2004); Hajij et al. (2018)) among others.

The major contribution of this work is the utilization of the tools available in persistent homology to study the topological information encoded in node embedding representations. For every node representation algorithm, our construction effectively defines a unique graph descriptor. We show the effectiveness of the proposed method by studying the topological descriptors induced by DeepWalk (Perozzi et al. (2014)), Node2Vec (Grover & Leskovec (2016)) and Diff2Vec (Rozemberczki & Sarkar (2018)). Our preliminary results are promising and indicate that the information extracted by our method can be effectively utilized for graph classification. The rest of the paper is organized as follows. Section 2 reviews the necessary tools for our constructions, persistent homology, and graph autoencoders. In Section 3, we describe the graph topological embeddings obtained using persistent homology and node embeddings. Section 4 is devoted to providing insights on the advantages of our method. Finally, we demonstrate our results in Section 5.

2 Setting the Scene

In this section, we review the necessary tools to build our machinery. Persistent Homology is reviewed in 2.1 whereas graph autoencoders are reviewed in 2.2.

2.1 Persistent Homology

Before proceeding, we assume the reader has basic familiarity with simplicial complexes and simplicial homology. We also would like to note that although we only introduce persistent homology for simplicial complexes, this work can be easily extended to more general domains[1]. Let K be a simplicial complex. We will denote the vertices of K by $V(K)$. Let S be an ordered sequence $\sigma_1, \cdots, \sigma_n$ of all simplices in K, such that for any simplex $\sigma \in K$ every face of σ appears before S. Then S induces a nested sequence of subcomplexes called a *filtration*: $\phi = K_0 \subset K_1 \subset ... \subset K_n = K$. A d-homology class $\alpha \in H_d(K_i)$ is said to be *born* at the time i if it appears for the first time as a homology class in $H_d(K_i)$. A class α *dies* at time j if it is trivial in $H_d(K_j)$ but not trivial in $H_d(K_{j-1})$. The *persistence* of α is defined to be $j - i$. Persistent homology captures the birth and death events in a given filtration and summarizes them in a multi-set structure called the *persistence diagram* $P^d(\phi)$. Specifically, the persistence diagram of the a filtration ϕ is a collection of pairs (i, j) in the plane where each (i, j) indicates a d-homology class that is created at time i in the filtration ϕ and killed entering time j. Persistent homology can be defined for *any filtration*. For the purpose of this work, we assume that the input is a piecewise linear function $f : |K| \longrightarrow \mathbb{R}$ defined on the vertices of complex K. Further, we assume the function f has different values on different nodes of K. Then, any such function induces the *lower-star* filtration as follows. Let $V = \{v_1, \cdots, v_n\}$ be the set of vertices of K sorted in non-decreasing order of their f-values, and let $K_i := \{\sigma \in K | \max_{v \in \sigma} f(v) \leq f(v_i)\}$. The lower-star filtration is defined as:

$$\mathcal{F}_f(K) : \phi = K_0 \subset K_1 \subset ... \subset K_n = K. \tag{1}$$

The interpretation of the lower-star filtration is rather intuitive; viz., the lower-star filtration reflects the topological information encoded in the scalar function f in the sense that the persistence homology induced by the filtration 1 is identical to the persistent homology of the sublevel sets of the function f. The lower-star filtration is our tool to extract the topological information encoded in node embeddings. We describe this in the following section.

2.2 Graph Autoencoders and Node Representation Learning

Representational learning on graphs has witnessed a surge of interest over the past few years. The main idea is to learn a mapping that embeds the nodes of a graph inside some latent space while preserving the nodes structural information. Although representational learning methods differ in what they learn from data, one may conceptualize various representational learning strategies via

[1] AutoEncoders on generalized simplicial complexes are given in (Hajij et al. (2020a)).

graph autoencoders (Hamilton et al. (2017)). We briefly review this abstraction here as it is needed in later sections. Let $G(V, E)$ be a graph, possibly weighted. An *encoder* on G is a function of the form:

$$enc : V_G \to \mathbb{R}^d. \tag{2}$$

This encoder associates to every node v in G a feature vector $\mathbf{z}_v \in \mathbb{R}^d$ that encodes the structure of v and its relationship to other nodes in G. A *decoder* is a function of the form: $dec : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$. The decoder associates to every pair of node embeddings a similarity score that quantifies some notion of relationship between these nodes. The pair (enc, dec) on G is called a *graph AutoEncoder* on G. The functions enc and dec are typically trainable functions that are optimized using user-defined similarity measure and loss function. The abstraction above encompasses most popular method for node representation learning including Graph Factorization (Angles & Gutierrez (2008)), Node2Vec (Grover & Leskovec (2016)), and DeepWalk (Perozzi et al. (2014)). In our context, we are interested in the case when the latent space in (2) is $\mathbb{R}$. Specifically, we are interested in the case when the encoder $enc : V_G \to \mathbb{R}$ is a scalar function on G. By utilizing the lower-star filtration on an encoder of interest, we obtain a topological descriptor on the graph G that reflects the topological information encoded in this learned representation.

3 Obtaining topological embeddings using Persistent Homology and Node Representation Learning

Given a learned node representation $enc : V_G \to \mathbb{R}$ as described in the previous section, one may extract the topological information encoded by computing the persistence diagrams of its lower-star filtration. The persistence diagram can then be converted to a feature vector inside some feature space to perform a downstream machine learning task of interest. This process of converting the persistence-based representation to a vector is called vectorization of the persistence diagram. Many vectorization schemes have been suggested recently. Examples of these schemes include betti curve (Umeda (2017)), the persistence landscape (Bubenik (2015)), the persistence image (Adams et al. (2017)) and many other (Chen et al. (2015); Berry et al. (2020); Kusano et al. (2016)). In our context, the persistence diagram of a learned representation embedding can be converted to a vector using any of the vectorization methods referenced above.

3.1 Persistence-based Node Embeddings

The above procedure can be altered to obtain a persistence-based node embedding that induces an encoder of the form 2. To this end, we need to introduce the *ego-network* of a vertex. Let w be given vertex G. The *k-hop ego network* of w, is a subgraph $G_w^k(\mathcal{N}^k(w), E_w^k)$ of G. The vertex set $\mathcal{N}^k(w)$ consists of the node w and all other nodes in V_G that can be reached in k steps from w. The edges E_w^k consists of every edge of G whose endpoints are in $\mathcal{N}^k(w)$. Given a node $w \in V_G$ and an encoder enc as described in 2, one may define the following restriction: $enc_w : \mathcal{N}^k(w) \to \mathbb{R}$, and then compute the lower-star filtration on this restriction to obtain a node-level topological representation of the node w.

One might wonder why we need to create the node-level persistence-based representation for a node w and what are the advantages of having this representation over the "raw" representation already contained in the encoder? We provide a justification for this point next.

4 Why Using Persistent Homology To obtain Descriptors from Node embeddings ?

Node representation embeddings naturally induce graph descriptors by directly using the node embedding readily available from the representation algorithm. For instance, one may simply add all node embeddings of a given graph to obtain a graph descriptor. This begs for the question, why do we need to utilize persistent homology to obtain a graph descriptor from the node embeddings?

Recently, there has been multiple efforts to provide a partial answer for the above question. For example, several studies Nielson et al. (2015); Joshi & Joshi (2019) reported that the topological features extracted by the persistence diagram can 1) track different information from the original

encodings obtained from the raw data Dey et al. (2017); Hajij et al. (2021) and 2) determine which predictors are more related to the outcome. In addition, the topological analysis of a graph allows to extract features that are invariant to the spatial transformation and more robust to noise Zheng (2015); Bae et al. (2017). For instance, DeepWalk and Node2Vec are stochastic node embeddings by design and different runs of the algorithm on the same graph yield different outputs. Our results in Section 5 indicates that while this is the case, the topological information extracted from the persistence diagram yielded a feature vector that is robust against such stochasticity. See Section 5 for more details. In addition, the fact that persistence homology tracks different information from the original deep learning-based representations (Dey et al. (2017); Hajij et al. (2021)) suggests that combining the encodings obtained from a persistent homology and other invariants obtained directly from node embedding should increase the quality of the learning task at hand. Further, persistent homology provides a clear, concise and rigorous method to quantify the level of expressiveness of the embeddings as we shall illustrate in Section 5.

5 Preliminary Results

To evaluate the proposed method, we used a mesh dataset (Sumner & Popović (2004)) that consists of 60 meshes divided into 6 categories: cat, elephant, face, head, horse, and lion (see Figure 1). Each category contains ten triangulated meshes[2]. We compared the proposed method with three node representation embeddings: DeepWalk Perozzi et al. (2014), Node2Vec Grover & Leskovec (2016) and Diff2Vec Rozemberczki & Sarkar (2018). For each mesh in the dataset, we compute the node representation vectors and then compute their respective $0-$persistence diagrams. Given two persistence diagrams obtained using the same node representation embedding, we measure the distance between them using the Wasserstein distance Carriere et al. (2017). The results are reported in Figure 1. This procedure gives us a Wasserstein distance matrix on the persistence diagrams induced by the lower-star filtration of the node-representation embedding method. Finally, we visualized the resulting discrete metric space using a 2d t-SNE projection (van der Maaten & Hinton (2008)) as shown in Figure 1. These results are promising and suggest that the information extracted in this method can be effectively utilized for a downstream machine learning pipeline on graphs. Moreover, observe that this study demonstrates that the expressive power of the obtained representation varies across the three chosen node embeddings.

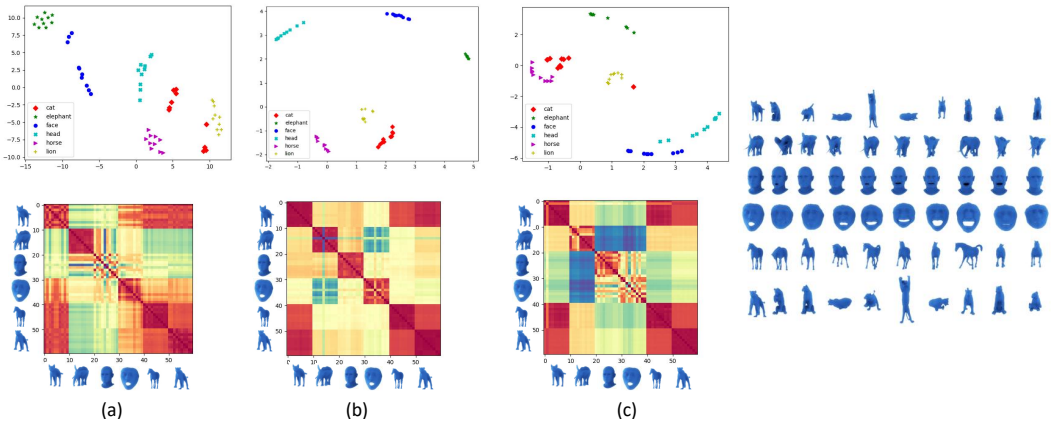

Figure 1: Figure (a) shows the 2d-TSNE projection on the top and the Wasserstein distance matrix on the bottom for Node2Vec, whereas figure (b) and (c) show the results for DeepWalk and Diff2Vec respectively. Note that each descriptor obtained this way clearly provides us with distinct clustering on the level of the metric space. Furthermore, meshes with similar structures, such as horses and lions have clusters that are close to each other.

[2]Each mesh in the dataset is converted to a weighted graph where the weight on an edge is simply the Euclidean distance between its nodes.

REFERENCES

Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *The Journal of Machine Learning Research*, 18(1):218–252, 2017.

Renzo Angles and Claudio Gutierrez. Survey of graph database models. *ACM Computing Surveys (CSUR)*, 40(1):1–39, 2008.

Marco Attene, Silvia Biasotti, and Michela Spagnuolo. Shape understanding by contour-driven retiling. *The Visual Computer*, 19(2):127–138, 2003.

Woong Bae, Jaejun Yoo, and Jong Chul Ye. Beyond deep residual learning for image restoration: Persistent homology-guided manifold simplification. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 145–153, 2017.

Chandrajit L Bajaj, Valerio Pascucci, and Daniel R Schikore. The contour spectrum. In *Proceedings of the 8th Conference on Visualization'97*, pp. 167–ff. IEEE Computer Society Press, 1997.

Eric Berry, Yen-Chi Chen, Jessi Cisewski-Kehe, and Brittany Terese Fasy. Functional summaries of persistence diagrams. *J. Appl. Comput. Topol.*, 4(2):211–262, 2020.

Peter Bubenik. Statistical topological data analysis using persistence landscapes. *The Journal of Machine Learning Research*, 16(1):77–102, 2015.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

Mathieu Carriere, Marco Cuturi, and Steve Oudot. Sliced wasserstein kernel for persistence diagrams. In *International Conference on Machine Learning*, pp. 664–673. PMLR, 2017.

Joseph Minhow Chan, Gunnar Carlsson, and Raul Rabadan. Topology of viral evolution. *Proceedings of the National Academy of Sciences*, 110(46):18566–18571, 2013.

Fenxiao Chen, Yun-Cheng Wang, Bin Wang, and C-C Jay Kuo. Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing*, 9, 2020.

Yen-Chi Chen, Daren Wang, Alessandro Rinaldo, and Larry Wasserman. Statistical analysis of persistence intensity functions. *arXiv preprint arXiv:1510.02502*, 2015.

D DeWoskin, J Climent, I Cruz-White, M Vazquez, C Park, and J Arsuaga. Applications of computational homology to the analysis of treatment response in breast cancer patients. *Topology and its Applications*, 157(1):157–164, 2010.

Tamal Dey, Sayan Mandal, and William Varcho. Improved image classification using topological persistence. In *Proceedings of the conference on Vision, Modeling and Visualization*, pp. 161–168, 2017.

Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.

Herbert Edelsbrunner, John Harer, Ajith Mascarenhas, and Valerio Pascucci. Time-varying reeb graphs for continuous space-time data. In *Proceedings of the twentieth annual symposium on Computational geometry*, pp. 366–372. ACM, 2004.

Jonathan L Gross and Jay Yellen. *Graph theory and its applications*. CRC press, 2005.

Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, 2016.

Mustafa Hajij, Bei Wang, Carlos Scheidegger, and Paul Rosen. Visual detection of structural changes in time-varying graphs using persistent homology. In *2018 IEEE Pacific Visualization Symposium (PacificVis)*, pp. 125–134. IEEE, 2018.

Mustafa Hajij, Kyle Istvan, and Ghada Zamzami. Cell complex neural networks. *NeurIPS 2020 Workshop TDA and Beyond.*, 2020a.

Mustafa Hajij, Nataša Jonoska, Denys Kukushkin, and Masahico Saito. Graph based analysis for gene segment organization in a scrambled genome. *Journal of Theoretical Biology*, pp. 110215, 2020b.

Mustafa Hajij, Ghada Zamzmi, and Fawwaz Batayneh. Tda-net: Fusion of persistent homology and deep learning features for covid-19 detection in chest x-ray images. *arXiv preprint arXiv:2101.08398*, 2021.

William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

Milan Joshi and Dhananjay Joshi. A survey of topological data analysis methods for big data in healthcare intelligence. *Int. J. Appl. Eng. Res*, 14:584–588, 2019.

Genki Kusano, Yasuaki Hiraoka, and Kenji Fukumizu. Persistence weighted gaussian kernel for topological data analysis. In *International Conference on Machine Learning*, pp. 2004–2013, 2016.

Sangeetha Kutty, Richi Nayak, and Lin Chen. A people-to-people matching system using graph mining techniques. *World Wide Web*, 17(3):311–349, 2014.

In So Kweon and Takeo Kanade. Extracting topographic terrain features from elevation maps. *CVGIP: image understanding*, 59(2):171–182, 1994.

Hyekyoung Lee, Moo K. Chung, Hyejin Kang, Boong-Nyun Kim, and Dong Soo Lee. Computing the shape of brain networks using graph filtration and gromov-hausdorff metric. *International Conference on Medical Image Computing and Computer Assisted Intervention*, pp. 302–309, 2011a.

Hyekyoung Lee, Moo K. Chung, Hyejin Kang, Bung-Nyun Kim, and Dong Soo Lee. Discriminative persistent homology of brain networks. *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pp. 841–844, 2011b.

Hyekyoung Lee, Hyejin Kang, Moo K. Chung, Bung-Nyun Kim, and Dong Soo Lee. Persistent brain network homology from the perspective of dendrogram. *IEEE Transactions on Medical Imaging*, 31(12):2267–2277, 2012.

PY Lum, G Singh, A Lehman, T Ishkanov, Mikael Vejdemo-Johansson, M Alagappan, J Carlsson, and G Carlsson. Extracting insights from the shape of complex data using topology. *Scientific reports*, 3:1236, 2013.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

David F Nettleton. Data mining of social networks represented as graphs. *Computer Science Review*, 7:1–34, 2013.

Jessica L Nielson, Jesse Paquette, Aiwen W Liu, Cristian F Guandique, C Amy Tovar, Tomoo Inoue, Karen-Amanda Irvine, John C Gensel, Jennifer Kloke, Tanya C Petrossian, et al. Topological data analysis for discovery in preclinical spinal cord injury and traumatic brain injury. *Nature communications*, 6(1):1–12, 2015.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710, 2014.

Giovanni Petri, Martina Scolamiero, Irene Donato, and Francesco Vaccarino. Networks and cycles: A persistent homology approach to complex networks. *Proceedings European Conference on Complex Systems 2012, Springer Proceedings in Complexity*, pp. 93–99, 2013a.

Giovanni Petri, Martina Scolamiero, Irene Donato, and Francesco Vaccarino. Topological strata of weighted complex networks. *PLoS ONE*, 8(6):e66506, 2013b.

Paul Rosen, Bei Wang, Anil Seth, Betsy Mills, Adam Ginsburg, Julia Kamenetzky, Jeff Kern, and Chris R Johnson. Using contour trees in the analysis and visualization of radio astronomy data cubes. *arXiv preprint arXiv:1704.04561*, 2017.

Benedek Rozemberczki and Rik Sarkar. Fast sequence-based embedding with diffusion graphs. In *International Workshop on Complex Networks*, pp. 99–107. Springer, 2018.

Robert W Sumner and Jovan Popović. Deformation transfer for triangle meshes. *ACM Transactions on graphics (TOG)*, 23(3):399–405, 2004.

Tanlin Sun, Bo Zhou, Luhua Lai, and Jianfeng Pei. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC bioinformatics*, 18(1):1–8, 2017.

Ichigaku Takigawa and Hiroshi Mamitsuka. Graph mining: procedure, application to drug discovery and recent advances. *Drug discovery today*, 18(1-2):50–57, 2013.

Yuhei Umeda. Time series classification via topological data analysis. *Information and Media Technologies*, 12:228–239, 2017.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. URL `http://www.jmlr.org/papers/v9/vandermaaten08a.html`.

Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE transactions on Big Data*, 6(1):3–28, 2018.

Yi Zheng. *Application of Persistent Homology in Signal and Image Denoising*. PhD thesis, Niedersächsische Staats-und Universitätsbibliothek Göttingen, 2015.