A Persistent Homology Perspective to the Link Prediction Problem

Sumit Bhatia[1], Bapi Chatterjee[2⋆], Deepak Nathani[3], and Manohar Kaul[3]

IBM Research AI, India,sumitbhatia@in.ibm.com
Institute of Science and Technology, Austria, bapi.chatterjee@ist.ac.at
Indian Institute of Technology, Hyderabad, {me15btech11009,mkaul}@iith.ac.in

Abstract. Persistent homology is a powerful tool in Topological Data Analysis (TDA) to capture topological properties of data succinctly at different spatial resolutions. For graphical data, shape and structure of the neighborhood of individual data items (nodes) is an essential means of characterizing their properties. We propose the use of persistent homology methods to capture structural and topological properties of graphs and use it to address the problem of link prediction. We achieve encouraging results on nine different real-world datasets that attest to the potential of persistent homology based methods for network analysis.

1 Introduction

A graph structure representing pairwise relations or interactions among individuals or entities recurs in diverse real-world applications such as social and professional networks, biological phenomena such as protein-protein interactions [10], and citation and collaboration networks [4]. In all these applications, understanding how the network evolves and the ability to predict the formation of new, hitherto non-existent links is extremely useful and has crucial applications such as predicting target genes for cancer research [31], social network analysis, and recommendation systems.

The Link Prediction Problem: Let U denote the set of all *possible* edges in graph $G = (V, E)$ with V as the vertex set, and E as the edge set. If G is undirected, $|U| = C(n, 2) = n(n-1)/2$, whereas, if G is directed, $|U| = 2{\times}C(n, 2) = n(n-1)$. The set $U - E$ is called the set of *potential* links. Often, in real-world settings, only a small subset of links $u \in U$ will materialize in future with $|u| << |U|$. For example, in a typical social network that has hundreds of millions of users (nodes), each user may only be friends (form an edge) with only a few hundred users. Given $G = (V; E)$, the task of identifying the edges $e \in u$ is challenging and requires understanding and modelling the differences between the sets u and $U - u$.

Why Persistent Homology for Link Prediction?: *Persistent homology* (PH) [11,12] is an algebraic tool for describing the structural features of a topological space at different spatial resolutions. By embedding a high-dimensional dataset in a topological space, PH allows us to extract and study crucial information about the *structure* and *shape* of the

⋆ Supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No. 754411 (ISTPlus).

dataset in a succinct manner. Since understanding the evolution and formation of edges in networks involves analyzing the structure and shape of the underlying networks, we posit that PH offers a theoretically sound framework to study such toplogical properties of networks. As an emerging technique in data mining, PH has been successfully applied in various applications such as text analysis [42], image analysis [8], temporal network analysis [33,16], and network classification [7].

Persistence Diagram: A popular tool from the realm of PH is *persistence diagram* (PD). *Homology* of a point set X roughly characterizes it in terms of *shape-features* like connected components, tunnels and voids. Given a graph $G = (V, E)$, mapping the nodes $v_i \in V$ to the points $\{x_i\}_{1 \leq i \leq n} \in X$, homology of X exhibits G in terms of the shape-features formed by its nodes and edges. However, these features depend a lot on the resolution or the scale at which they are studied, and it is crucial to study them across a spectrum of spatial resolutions. The features that persist across resolutions constitute its *persistent homology* (PH) represented by its PDs. PD is depicted as a set of points in a two-dimensional space whose indices correspond to the resolutions at which the topological features are "born" and subsequently, "die". Differences between PHs of two graphs (or subgraphs) can be captured by *dissimilarity measure* such as The *Wasserstein* or *Bottleneck* [12, Chapter VIII] distance between their corresponding PDs. Using such dissimilarity measures between PDs, we understand how an adaptive-sized extended neighborhood of query nodes changes in terms with regards to their PH when an edge is added (removed) to (from) the graph.

Our Contributions: We describe a novel approach for predicting links in networks by utilizing the Persistence Diagrams of different neighborhood sub-graphs around the query nodes. Specifically, we characterize the existence of a potential link between a pair of query nodes in terms of a dissimilarity measure between a number of specially constructed neighborhoods. We first present the necessary mathematical notions to describe our method: the PD of a graph and the distance measures between PDs (Section 2). We then argue and explain that for a pair of nodes, the PDs of the subgraph induced by their extended neighborhood should *not* change much by addition or removal of a naturally existing edge. We also provide a theoretical insight into the working of our approach (Section 3). We describe and discuss the experiments conducted using nine different real-world network datasets that provide strong empirical evidence for the potential of application of PH for link prediction, and network analysis in general. Our proposed approach achieves robust performance across all the datasets when compared with six commonly used baseline methods for link prediction (Section 4).

Overview of and Comparison with Related Work: Most methods for link prediction utilize the structural properties of the underlying network to predict formation of new edges. Some of the most frequently used methods [1,29] utilize the intuition that the likelihood of a link between two nodes is high if they share many common neighbors. Despite being widely adopted due to their intuitive nature and ease of computation, such methods are limited to the second order neighborhood of the source node and ignore the global structural information about the underlying network. On the other hand, studying the shape features of the graph at varying resolutions enables us to capture the global structure information.

Different other approaches that consider global information for link prediction include measures based on an ensemble of all paths (such as the Katz score [21]), measures derived from conducting random walks over the graph [2,18], and learning continuous vector representations of nodes in the graph such that the nodes sharing similar structural properties are mapped close to each other in the latent space (e.g., DeepWalk [34], LINE [38], node2vec [15], struc2vec [35]). Ensemble methods that complement the network information with external information such as text documents have also been proposed [6]. In contrast to these methods that need to explore the entire graph for capturing global information, our approach is adaptive: we only study the combined neighborhood whose size varies depending on the sparsity of the graph. Thus, we can also avoid the large cost of exploring the entire graph.

2 Persistent Homology of a Graph

For a self-contained exposition, we briefly present the definitions of main concepts used in this work. For a detailed description, a reader can refer to any well-known book on computational topology [12]. A quick yet sufficient introduction to some more basic concepts can also be found in the extended pre-print of this paper [5].

Persistence Diagram: Let Δ be a finite *abstract simplicial complex* and $\{\Gamma_i\}_{i\in I}$ s.t. $\emptyset = \Gamma_0 \subsetneq \Gamma_1 \subsetneq \Gamma_2 \ldots \subsetneq \Gamma_p = \Delta$ be a *filtration* of Δ. For a pair $i, j\ s.t.\ 0 \leq i \leq j \leq p$, this inclusion relation among Γ_is induces a homomorphism on the simplicial homology group of each dimension $n \in \mathbb{Z}$ given by $f_n^{i,j} : H_n(\Gamma_i) \rightarrow H_n(\Gamma_j)$.

The n^{th} *persistent homology (PH) group* is the image of the homomorphism $f_n^{i,j}$ given by $Im(f_n^{i,j})$. In turn, the n^{th} *persistent Betti number* is defines as the rank of $Im(f_n^{i,j})$ given by $\beta_n^{i,j} = rank(Im(f_n^{i,j}))$.

The n^{th} persistent Betti number counts how many *homology classes* of dimension n survives a passage from Γ_i to Γ_j. We say that a homology class $\alpha \in H_n(\Gamma_i)$ is *born* at *resolution* i if it did not come from a previous *sub-complex*: $\alpha \notin Im(f_n^{i-1,i})$. Similarly, we say that a homology class dies at resolution j if it does not belong to the sub-complex Γ_j and belonged to previous sub-complexes.

A *persistence diagram (PD)* is a plotting of the points (i, j) corresponding to the birth and death resolutions, respectively, for each of the homology classes. Because a homology class can not die before it is born, every point (i, j) lies above the diagonal $x = y$. If a homology class does not die after its birth, we draw a vertical line starting from the diagonal in correspondence to its birth. For practical purposes, we take a *persistence threshold* τ, and assume that every homology class dies at the resolution τ. A typical PD is shown in Figure 1.

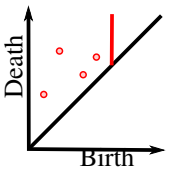

Fig. 1. PD

Distance between PDs: Let P_1 and P_2 be two PDs. Let η be a bijection between the points in the two diagrams. We define the following two

distance measures:

$$\text{(a) } \textit{Wasserstein-q distance} : W_q(P_1, P_2) = (\inf_{\eta: P_1 \to P_2} \sum_{p \in P_1} ||p - \eta(p)||_\infty^q)^{\frac{1}{q}} \quad (1)$$

$$\text{(b) } \textit{Bottleneck Distance} : W_\infty(P_1, P_2) = \inf_{\eta: P_1 \to P_2} \sup_{p \in P_1} ||p - \eta(p)||_\infty \quad (2)$$

The Wasserstein-q distance is sensitive to small differences in the PDs, whereas, the Bottleneck distance captures relatively large differences.

Rips Complex: A Vietoris-Rips Complex, also called a *Rips complex* is an abstract simplicial complex defined over a finite set of points $X = \{x_i\}_{i=1}^n \subseteq \mathcal{X}$ in a metric space $(\mathcal{X}, d)$. Given X and a real number $r > 0, r \in \mathbb{R}$, a Rips complex $R(X, r)$ is formed by connecting the points for which the balls of radius $\frac{r}{2}$ centered at them intersect. In the context of the same point set, we use R_r to denote $R(X, r)$. A 1-simplex is formed by connecting two such points and corresponds to an edge. A 2-simplex is formed by 3 such points and corresponds to a triangular face.

Rips Filtration: Given a set of points $X = \{x_i\}_{i=1}^n \subseteq \mathcal{X}$, let $0 = r_0 \leq r_1 \leq r_2 \ldots \leq r_m$ denote a finite sequence of increasing real numbers, which we use to construct Rips complexes $\{R_{r_i}\}_{i=1}^m$ as defined above. Clearly, by construction of Rips complexes the sequence $\{R_{r_i}\}_{i=1}^m$ is nested and thus provides a filtration of R_{r_m}:

$$\emptyset = R_{r_0} \subsetneq R_{r_1} \subsetneq R_{r_2} \ldots \subsetneq R_{r_m}$$

Deriving the PH groups via homomorphism over a Rips filtration, we obtain a PD associated with the point set X. Please note that to compute the Rips filtration associated with a point set X we need only the relative pairwise distances between the points $x_i \in X$. Essentially, we need a symmetric distance matrix $D = \{d(x_i, x_j)\}_{i=1,j=1}^{n,n}$ to compute the PD of X. Next, we will use this method to compute the PD of a graph.

Remark: Without going in details, we would like to mention that there are many choices for filtrations and distance metric available when applying PH to a graph[1], however, for this application, computational simplicity and well-developed software that could scale to real world datasets were the main factors for us to decide on Rips filtration with shortest-path metric.

2.1 Persistence Diagram of a Graph

Consider a graph $G = (V, E)$, where $V = \{v_i\}_{i=1}^n$ is the node set and $E = \{e_i\}_{i=1}^m$ is the edge set. We associate a positive weight $w_{e_i} \in \mathbb{R}, w_{e_i} > 0$ with each of the elements $e_i \in E$. For an *unweighted* graph, $w_{e_i} = 1, \forall e_i \in E$. If two nodes are not connected by an edge, we take the (virtual) edge-weight between them as ∞, which for practical purposes is taken as a large positive real number $M \in \mathbb{R}$. The *shortest-path distance* $D_{sp}(v_i, v_j)$ between the nodes $v_i, v_j \in V$ is defined as the sum of weights of the edges on the path starting at v_i and terminating at v_j.

[1] https://topology.ima.umn.edu/node/53

Now consider the metric space $(\mathcal{X}, d)$ equipped with a metric d. Let $X = \{x_i\}_{i=1}^n$ be a set of points in $(\mathcal{X}, d)$ such that the points in X correspond to the nodes in $V = \{v_i\}_{i=1}^n$. In an *undirected* graph, where the shortest-path distance D_{sp} between any two nodes is symmetric, it makes a natural choice for a metric. We can verify that D_{sp} satisfies all the properties of a metric: for arbitrary $v_i, v_j, v_k \in V$, (a) $D_{sp}(v_i, v_j) \geq 0$, (b) $D_{sp}(v_i, v_j) = 0 \iff v_i = v_j$, (c) $D_{sp}(v_i, v_j) = D_{sp}(v_j, v_i)$ and (d) $D_{sp}(v_i, v_j) + D_{sp}(v_j, v_k) \geq D_{sp}(v_i, v_k)$. Therefore, for points $x_i, x_j \in X$, which correspond to $v_i, v_j \in V$, we take the metric as $d(x_i, x_j) = D_{sp}(v_i, v_j)$.

For a *directed* graph, the shortest-path distance between two nodes is not symmetric. In this case, $d(x_i, x_j) = D_{sp}(v_i, v_j)$ provides a *quasi-metric*: it satisfies (a), (b) and (d) as described above. From a quasi-metric $d(x_i, x_j)$, we derive a metric as follows:

$$f_a(x_i, x_j) = a \times d(x_i, x_j) + (1 - a) \times d(x_j, x_i)$$

where $a \in [0, 1/2]$ [40]. For $a = \frac{1}{2}$, $f_a(x_i, x_j)$ is the average of the two directed distances. In this work, for a metric space representation of a directed graph, we take $d(x_j, x_i) = \frac{D_{sp}(v_i, v_j) + D_{sp}(v_j, v_i)}{2}$, where $x_i, x_j \in X$ correspond to $v_i, v_j \in V$.

Computing the all-pair-shortest-path (APSP) in an undirected graph [19] gives a symmetric *distance matrix* $D = \{d_{ij}\}_{i=1,j=1}^{n,n}$. For a directed graph, the distance matrix is not symmetric; therefore, to impose a metric structure we apply the aforementioned method: $d_{ij} = d_{ji} = \frac{d_{ij} + d_{ji}}{2}$. With that, we have a complete pipeline to compare the shape-features of two graphs (or subgraphs) using PH.

3 Link Prediction via Persistent Homology

Having discussed the background to compute the quantitative differences between a pair of subgraphs with respect to their shape-features, we describe how to use that to understand and predict the existence of a potential link. First, we summarize the entire pipeline of computing the PD for a graph G.

PD Computation: To start with, we compute the all-pair-shortest-path distance matrix D using Johnson's algorithm [19]. In case G is directed, D is made symmetric as described in the section 2.1. Thereafter, D and a persistence-threshold τ are used to compute the PD of G. Efficient implementations for PD computations, such as the one by Bauer [3], could be used for this purpose.

Now consider the cases of combined-neighborhood of nodes u and v as shown in Figure 2. We consider two scenarios with respect to reasonably extended neighborhoods of the two nodes, as shown in Figure 2 (a) and (b). A potential link is shown by the dotted curve.

Essentially, a case of predicting a link between an arbitrary pair of nodes lies on the spectrum of scenarios starting at the one shown in the fig. 2 (a) and stretches towards the ones similar to the fig. 2 (b). As we explained before, the existence of a possible link has higher chances as we move away from the case of the fig. 2 (a) on this spectrum.

With that observation, we explore and understand how the difference in shape-measures, as provided by the distances in the PDs of a number of subgraphs induced by

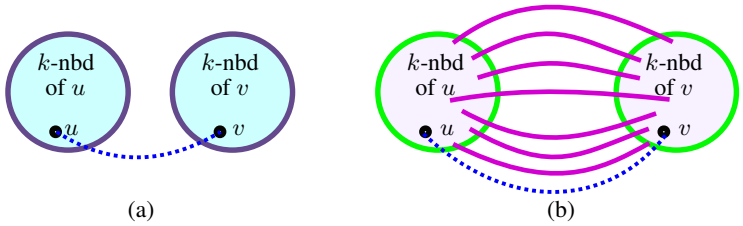

Fig. 2. Combined-neighborhood of u and v, when they have (a) no edges connecting (b) multiple edges connecting.

the combined-neighborhood of u and v, varies when we examine the cases of arbitrary pair of nodes. This is presented in the Algorithm 1.

Given a graph $G = (\{v_i\}_{i=1}^{n}, \{e_k\}_{k=1}^{m})$, for a $k \leq n$, first, we compute the subgraph of G induced by the i-hop neighbors of u and v, where $1 \leq i \leq k$, see lines 2 and 3. Thereafter, we compute the subgraph induced by an i-hop combined-neighborhood the two nodes, where where $1 \leq i \leq r$, see line 4. The radius of the individual and combined neighborhoods, k and r, respectively, are chosen such that there could be a positive probability of covering of the combined-neighborhood by the union of the two individual neighborhoods, and therefore, $k \leq 2r$. From this subgraph, we induce two subgraphs corresponding to the existence and non-existence of a link between the query nodes, see lines 5 and 6. Following our intuition, a missing link in a complete graph has high chances of existence, therefore, we also construct a complete graph over the nodes of the combined neighborhood, line 7. Having collected these subgraphs, we compute their PDs as described previously.

In the PDs, we have considered only 0^{th} PH groups. This is because the cycles in a graph, which correspond to its 1^{st} PH group, are never destroyed as there are no 2-faces. Thus, for our purpose, distances between the 1-dimensional PDs of the subgraphs would not help much. In the subsequent discussion, by the topological features we shall mean the 0^{th} dimensional features i.e. the number of connected components.

We compute the Wasserstein-2: d_1, d_2, d_3 and d_4, and the Bottleneck: d_5, d_6, d_7 and d_8 distances between the PDs, as shown in the lines 11 to 15. They signify how much the induced *subgraphs are dissimilar* with respect to their shape-features. We use d_is, $1 \leq i \leq 8$, in our experiments to perform link-prediction as a ranking task (Section 4).

Computational cost: To implement Algorithm 1, we leveraged parallelization as much as possible. For example, for shortest-path computation, we use a simple shared-memory thread-based parallelization of applying Dijkstra's algorithm, which runs in $\tilde{O}(|V|^2)$ (assuming $|V| > |E|$), for each of the nodes, and thus pay roughly $\tilde{O}(|V|^3/p)$, where p is the number of threads, and store the APSP matrix in a database. The neighborhood and combined neighborhood computation steps are linear in the maximum degree, thus $O(|V|)$. The PD computation is performed by reduction of the APSP matrix to cost $O(|V|^3)$ arithmetic operations. W_q and W_∞ distance computation steps are linear in the size of PDs. Effectively, Algorithm 1 costs $O(|V|^3)$. Next, we sketch a theoretical justification of our approach.

Algorithm 1 Bottleneck and Wasserstein-2 dist. computation.

Input: Graph G, Nodes u, v, Neighborhood radius k, Combined-neighborhood radius r, persistence-threshold τ, a boolean isD to indicate if directed.

Algorithm GETDIST(G, u, v, k, r, τ, isD)
  $N_u^k \leftarrow$ GETNBRS(u, k); ▷Induced subgraph over i-hop neighbors of u, where $1 \leq i \leq k$.
  $N_v^k \leftarrow$ GETNBRS(v, k);
  $N_{u,v}^r \leftarrow$ GETCOMBINEDNBRS(u, v, r); ▷Induced subgraph over i-neighbors of u or v or both, where $1 \leq i \leq r$.
  $N_{u,v}^{r+} \leftarrow N_{u,v}^r \cup (u, v)$; ▷Induced subgraph $N_{u,v}^r$ augmented with the edge (u, v).
  $N_{u,v}^{r-} \leftarrow N_{u,v}^r - (u, v)$; ▷Induced subgraph $N_{u,v}^r$ without the edge (u, v).
  $C(N_{u,v}^r) \leftarrow$ MAKECOMPLETE$(N_{u,v}^r)$; ▷The complete graph over the nodes of $N_{u,v}^r$.
  $P_u \leftarrow$ PD(N_u^k, τ, isD); ▷Persistence diagram of the subgraph induced by N_u^k.
  $P_v \leftarrow$ PD(N_v^k, τ, isD); $P^+ \leftarrow$ PD$(N_{u,v}^{r+}, \tau, isD)$;
  $P^- \leftarrow$ PD$(N_{u,v}^{r-}, \tau, isD)$; $P^c \leftarrow$ PD$(C(N_{u,v}^r), \tau, isD)$;
  $d_1 \leftarrow$ W-2-DIS(P^+, P^-); $d_2 \leftarrow$ W-2-DIS(P^+, P^c);
  $d_3 \leftarrow$ W-2-DIS(P^+, P_u); $d_4 \leftarrow$ W-2-DIS(P^+, P_v);
▷Wasserstein-2 distances between the Ps
  $d_5 \leftarrow$ B-DIS(P^+, P^-); $d_6 \leftarrow$ B-DIS(P^+, P^c);
  $d_7 \leftarrow$ B-DIS(P^+, P_u); $d_8 \leftarrow$ B-DIS(P^+, P_v);
▷Bottleneck distances between the Ps
  $\tilde{d} \leftarrow \{d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8\}$; ▷A vector of the eight distances.
  Output $\tilde{d}$;
end Algorithm

3.1 Why this algorithm works?

While the commonly used link-prediction heuristics [1,29,23,39], have been empirically validated, to the best of our knowledge, only a limited number of works [36,9] have explored why such methods should work. McPherson et al. [28] suggest that the network of real-life interactions stem from *homophily*. Hoff et al. [17] introduced a statistical model for such networks, that was extended by Sarkar et al. [36]. Essentially, all these models represent a graph-node by a point in a *latent* d-dimensional Euclidean space and suggest that the probability of the existence of a link between two query nodes u and v can be defined in terms of a parameterized logistic function of the distance between the corresponding points as follow [36]:

$$P(u \sim v | d_{uv}) = \frac{1}{1 + e^{\alpha(d_{uv} - r)}} \tag{3}$$

where $u \sim v$ denotes the existence of a link between the nodes u, v, and α and r are the parameters of *function-sharpness* and *sociability of the nodes*, respectively. Thus, a smaller distance d_{uv} in the latent space implies a higher probability of link between u and v. Under the constraints of space, we now explain how decreasing the distances d_1 to d_8 in Algorithm 1 corresponds to decreasing d_{uv} in the eq. (3).

First, note that the distances d_is, $i \leq 1 \leq 8$, are essentially based on the optimal matchings between the PDs and behave very differently from the Euclidean metrics. See eqs. (1) and (2): higher the value of $\eta(p)$ for each $p \in P_1$, lower are the $W_q(P_1, P_2)$ and $W_\infty(P_1, P_2)$. $\eta(p)$, as a bijection, represents matchings between the PDs P_1 and P_2. Thus, the lower values of d_is reflect higher matchings between the PDs indicating that the compared subgraphs have more similar topological features. An attentive reader would also have noticed that the PDs that we compare to generate d_is, correspond to the simplicial sub-complexes over the subsets of the same dataset obtained by the embedding of a graph in a metric space. It is easy to observe that these subsets overlap by virtue of the construction of the subgraphs induced by the combined neighborhoods of the query nodes. In this setting, a higher matching in the PDs indicates highly similar topological features and these similar features are over the common subset of the subgraphs. Now, we discuss the individual subgraph comparison summaries captured by the d_is:
(a) d_1 and d_5: smaller values of d_1 and d_5 indicate that augmenting a possible edge between the query nodes does not change the topological features of the subgraph induced by the combined neighborhood. **(b)** d_3 and d_7: their smaller values imply that the combined neighborhood itself is not much different from the neighborhood of the first node in terms of the topological features. **(c)** d_4 and d_8: same as (b) for the second node. **(d)** d_2 and d_6: smaller values of d_2 and d_6 indicate that the subgraph induced by the combined-neighborhood is closer to a complete graph in terms of topological features.

Let $n_l(u, v)$ denote the number of paths of length l between the nodes u and v. From the above summary, in general terms, it can be inferred that smaller the values of d_is, $i \leq 1 \leq 8$, (a) the combined-neighborhood lies closer to the structure shown in Figure 2(b) on the spectrum of the scenarios mentioned in Section 3. For example, smaller d_2 and d_6 would indicate that the combined-neighborhood is similar to a complete graph in which the likelihood of completion of a missing link is very high, and (b) because of the fact that higher overlap of neighborhood subgraphs, $n_l(u, v)$ is non-zero for increasing number of small path-lengths l.

In our method, the metric space embedding of the graph translates it into a point cloud in Euclidean space where even though the points are at non-deterministic positions, the distance between them is deterministic. Essentially, it aligns to the deterministic model, (see Sections 3 and 4 of [36], with (a) identical radii for unweighted graphs and (b) non-identical radii of weighted graphs. Thus in the spirit of the discussion in Section 5 for the bounds over d_{uv}, the Lemma 5.7, and Theorem 5.8 in the paper by Sarkar et al. [36], and inferring from the point (b) in the previous paragraph, $P(u \sim v|d_{uv})$ increases with decrease in the values of d_is, $i \leq 1 \leq 8$.

4 Experiments

4.1 Experimental Protocol

Datasets: Table 1 lists the nine publicly available datasets that were used for evaluating our proposed approach. The datasets selected are from different domains and widely used in the study of complex networks.

Baselines: We compare the performance of our approach with six frequently used methods for link prediction. We consider Common Neighbors (CN), Adamic-Adar (AA) [1] and Milne-Witten (MW) [29] as representative local methods . We chose Preferential Attachment (PA), node2vec (N2V) [15], and struc2vec (S2V) [35] as representative global methods.

Implementation: We implemented our approach in C++ using the Ripser library [3] for computing PDs. We used the publicly available code[3] of Kerber et al. [22] to compute W_2 and W_∞ distances. We fixed the persistence threshold $\tau = 4$. It was empirically found that beyond $\tau = 4$ PD did not change. The neighborhood and combined-neighborhood radii k and r are taken as $\lceil \frac{L}{4} \rceil$ and $\lceil \frac{L}{2} \rceil$, respectively, where L is the shortest path distance between the two query nodes. This selection of k and r is adaptable to the position of query nodes and ensures that there is a reasonable intersection of their neighborhoods. Empirically we found that increasing this value did not change the distance d_i's but only increased the computation time. We implemented the baselines AA, MW, CN, and PA in C++ and used author provided source code for node2vec and struc2vec. For all the datasets, we removed 5% of edges making sure that the residual graph remains connected. We then compare the performance of different methods to recover the removed edges using information from the residual graph (Sec 4.2). All the datasets and our source code are available for download[4].

	# nodes	# edges	N/w Type
DC [32]	112	425	Word Co-occurrence n/w
ATC[2]	1226	2615	Air Traffic n/w
Cora [26]	2708	5429	Citation n/w
Euroad [37]	1174	1417	Road n/w
Figeyes [14]	2239	6452	Protein interaction n/w
Yeast [10]	1870	2277	Protein interaction n/w
Power [41]	4941	6594	Power Grid n/w
arXiv [24]	5242	14496	Collaboration n/w
Twitter [27]	23370	33101	Social N/w

Table 1. Different datasets used in experiments

4.2 Results and Discussions

Traditionally, the problem of link prediction has been addressed as a ranking problem where given a source node, a ranked list of target nodes is produced ordered by the likelihood of a potential link being formed between the source and the target nodes [25,20]. The baselines CN, AA, MW, and PA by definition, output a score between the source and target node that can be used as the ranking function. The other two baselines – N2V and S2V – learn continuous vector representations for each node in the graph. A typical way to rank target nodes given a source node is to rank them based on their distance from the source node [30]. Hence, for these methods, given a source node, we produce a ranked list of all the other nodes in the graph ordered by the Euclidean distance between the source and target node vectors. Given a pair of source and target nodes our proposed approach produces eight different distance values (Algorithm 1) capturing different topological properties. In order to produce a ranked list that combines these different properties captured by the different distance functions, we use the

[3] https://bitbucket.org/grey_narn/hera/src/master/

[4] https://github.com/sumit-research/persistent-homology-link-prediction

	Hits @ 10							Hits @ 50							Hits @ 100						
	CN	**AA**	**MW**	**PA**	**S2V**	**N2V**	**PH**	**CN**	**AA**	**MW**	**PA**	**S2V**	**N2V**	**PH**	**CN**	**AA**	**MW**	**PA**	**S2V**	**N2V**	**PH**
DC	.190	**.285**	.142	.333	.142	.095	.000	.571	.666	.619	.571	.476	.476	**.761**	.714	.714	.714	**1.00**	.952	.952	**1.00**
ATC	**.100**	.061	.053	.023	.038	.061	.077	.161	.076	.092	.130	.138	.238	**.263**	.161	.076	.092	.215	.184	**.384**	.372
Cora	.180	.080	.074	.016	.028	.048	**.332**	.232	.080	.074	.038	.052	.118	**.338**	.252	.080	.074	.040	.072	.144	**.338**
Euroad	.085	.085	.085	.014	.000	.100	**.185**	.085	.085	.085	.028	.114	.557	**.600**	.085	.085	.085	.071	.214	**.742**	.728
Figeyes	.000	.006	.000	.000	.006	**.012**	.003	.012	.006	.018	.003	.018	.024	**.027**	.015	.006	.024	.015	.043	.043	**.046**
Yeast	.212	**.247**	.159	.008	.017	.150	.183	.256	.292	.283	.079	.053	.292	**.339**	.256	.292	.292	.159	.106	.362	**.385**
Power	.227	.209	.182	.000	.015	.246	**.267**	.255	.255	.255	.009	.039	.574	**.595**	.255	.255	.255	.030	.072	.680	**.747**
Arxiv	.580	**.587**	.135	.015	.122	.480	.237	.849	**.874**	.526	.070	.219	.823	.723	.904	**.918**	.709	.114	.238	.897	.865
Twitter	**.055**	.046	.047	.000	.003	.000	.003	.085	.053	.161	.002	.010	.001	**.117**	.087	.053	.236	.011	.015	.001	**.276**

Table 2. Performance of different methods on nine different datasets for the link prediction task. Hits at ranks 10,50, and 100 are reported. For each dataset, the best method achieving highest hits at a given rank is highlighted in bold.

rank product metric [13] to combine the ranked lists produced by individual distance functions to obtain the final ranking of target nodes with respect to a given source node. For a node i, the rank product is computed as $rp_i = (\prod_{j=1}^{m} r_{ij})^{1/m}$ where r_{ij} is the rank of node i in the j^{th} ranked list.

Table 2 summarizes the results achieved by the six baselines and our proposed approach (PH). We report *Hit Rate*@N (for $N = \{10, 50, 100\}$), – the proportion of edges for which the correct target node was ranked in the top N positions. Observe that our approach outperforms the baselines in most cases, and is a close second in others. Also note that while the methods based on immediate neighborhood achieve the best values for five out of nine datasets in terms of $Hits@10$, the methods that utilize global network information generally outperform the local methods at higher ranks. This is expected as the local methods work in a small, though highly relevant, search space of nodes in the immediate neighborhood of query nodes. Thus, they are able to predict the links for a few test cases that lie in this small search space. However, they fail for hard test cases that lie outside this search space. For instance, in the `euroad` dataset, only 6 out of 70 test cases lie in the first order neighborhood of query nodes, resulting in poor performance of local methods. On the other hand, the global methods (N2V, S2V, PH) outperform at higher ranks as they are not limited to this small search space.

The robust performance achieved by the proposed approach, for all the datasets and at different ranks, is commendable given that the proposed approach uses only eight features (distance functions comparing the topological properties) that can be computed with relative ease compared to computationally expensive learning of vector representations (as is the case with node2vec and struc2vec). Further, unlike the CN, AA, MW, and PA baselines, that are also easier to compute, the proposed approach is built upon the solid theoretical foundations and is not limited to the immediate neighborhood of query nodes.

5 Conclusions and Future Work

We proposed an approach inspired from persistent homology to model link formation in graphs and use it to predict missing links. Our approach achieved robust and stable per-

formance across nine datasets, outperforming many frequently used baseline methods despite being relatively simple and computationally less expensive. Given that the topological features succinctly capture information about shape and structure of the network and can be computed without the need of extensive training, it will be worth exploring how these features can be combined with other techniques for network analysis.

References

1. Adamic, L.A., Adar, E.: Friends and neighbors on the web. Social Networks 25(3), 211 – 230 (2003)
2. Backstrom, L., Leskovec, J.: Supervised random walks: Predicting and recommending links in social networks. In: WSDM '11 (2011)
3. Bauer, U.: Ripser. https://github.com/Ripser/ripser (2018)
4. Bhatia, S., Caragea, C., Chen, H.H., Wu, J., Treeratpituk, P., Wu, Z., Khabsa, M., Mitra, P., Giles, C.L.: Specialized research datasets in the citeseer^x digital library. D-Lib Magazine 18(7/8) (2012)
5. Bhatia, S., Chatterjee, B., Nathani, D., Kaul, M.: Understanding and predicting links in graphs: A persistent homology perspective. arXiv preprint arXiv:1811.04049 (2018)
6. Bhatia, S., Vishwakarma, H.: Know thy neighbors, and more!: Studying the role of context in entity recommendation. In: Hypertext (HT). pp. 87–95 (2018)
7. Carstens, C.J., Horadam, K.J.: Persistent homology of collaboration networks. Mathematical problems in engineering 2013 (2013)
8. Chung, M.K., Bubenik, P., Kim, P.T.: Persistence diagrams of cortical surface data. In: International Conference on Information Processing in Medical Imaging. pp. 386–397 (2009)
9. Cohen, S., Zohar, A.: An axiomatic approach to link prediction. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
10. Coulomb, S., Bauer, M., Bernard, D., Marsolier-Kergoat, M.C.: Gene essentiality and the topology of protein interaction networks. Proceedings of the Royal Society B: Biological Sciences 272(1573), 1721–1725 (2005)
11. Edelsbrunner, H., Harer, J.: Persistent homology-a survey. Contemporary mathematics 453, 257–282 (2008)
12. Edelsbrunner, H., Harer, J.: Computational Topology - an Introduction. American Mathematical Society (2010)
13. Eisinga, R., Breitling, R., Heskes, T.: The exact probability distribution of the rank product statistics for replicated experiments. FEBS Letters 587(6), 677 – 682 (2013)
14. Ewing, R.M., Chu, P., Elisma, F., Li, H., Taylor, P., Climie, S., McBroom-Cerajewski, L., Robinson, M.D., O'Connor, L., Li, M., et al.: Large-scale mapping of human protein–protein interactions by mass spectrometry. Molecular systems biology 3(1), 89 (2007)
15. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD. pp. 855–864 (2016)
16. Hajij, M., Wang, B., Scheidegger, C., Rosen, P.: Visual detection of structural changes in time-varying graphs using persistent homology. In: PacificVis. pp. 125–134. IEEE (2018)
17. Hoff, P.D., Raftery, A.E., Handcock, M.S.: Latent space approaches to social network analysis. Journal of the american Statistical association 97(460), 1090–1098 (2002)
18. Jeh, G., Widom, J.: Simrank: a measure of structural-context similarity. pp. 538–543. ACM (2002)
19. Johnson, D.B.: Efficient algorithms for shortest paths in sparse networks. Journal of the ACM (JACM) 24(1), 1–13 (1977)

20. Kataria, S., Mitra, P., Bhatia, S.: Utilizing context in generative bayesian models for linked corpus. In: AAAI. vol. 10, p. 1 (2010)
21. Katz, L.: A new status index derived from sociometric analysis. Psychometrika 18(1), 39–43 (Mar 1953)
22. Kerber, M., Morozov, D., Nigmetov, A.: Geometry helps to compare persistence diagrams. In: 2016 Proceedings of the Eighteenth Workshop on Algorithm Engineering and Experiments (ALENEX). pp. 103–112. SIAM (2016)
23. Leskovec, J., Backstrom, L., Kumar, R., Tomkins, A.: Microscopic evolution of social networks. In: KDD. pp. 462–470 (2008)
24. Leskovec, J., Kleinberg, J., Faloutsos, C.: Graph evolution: Densification and shrinking diameters. ACM Trans. Knowl. Discov. Data 1(1) (Mar 2007)
25. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. Journal of the American society for information science and technology 58(7), 1019–1031 (2007)
26. Lu, Q., Getoor, L.: Link-based classification. In: Fawcett, T., Mishra, N. (eds.) ICML. pp. 496–503. AAAI Press (2003), http://www.aaai.org/Library/ICML/2003/icml03-066.php
27. McAuley, J., Leskovec, J.: Learning to discover social circles in ego networks. In: NIPS, pp. 548–556 (2012)
28. McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in social networks. Annual review of sociology 27(1), 415–444 (2001)
29. Milne, D., Witten, I.: An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy. pp. 25–30 (2008)
30. Misra, V., Bhatia, S.: Bernoulli embeddings for graphs. In: AAAI. pp. 3812–3819 (2018)
31. Nagarajan, M., et al.: Predicting future scientific discoveries based on a networked analysis of the past literature. In: KDD. pp. 2019–2028. ACM (2015)
32. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Physical Review E 74(3) (2006)
33. Pal, S., Moore, T.J., Ramanathan, R., Swami, A.: Comparative topological signatures of growing collaboration networks. In: Workshop on Complex Networks CompleNet. pp. 201–209. Springer (2017)
34. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: KDD. pp. 701–710 (2014)
35. Ribeiro, L.F., Saverese, P.H., Figueiredo, D.R.: struc2vec: Learning node representations from structural identity. In: KDD. pp. 385–394 (2017)
36. Sarkar, P., Chakrabarti, D., Moore, A.W.: Theoretical justification of popular link prediction heuristics. In: IJCAI (2011)
37. Šubelj, L., Bajec, M.: Robust network community detection using balanced propagation. The European Physical Journal B 81(3), 353–362 (Jun 2011)
38. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: WWW. pp. 1067–1077 (2015)
39. Tang, L., Liu, H.: Relational learning via latent social dimensions. In: KDD. pp. 817–826 (2009)
40. Turner, K.: Generalizations of the rips filtration for quasi-metric spaces with persistent homology stability results. arXiv preprint arXiv:1608.00365 (2016)
41. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. nature 393(6684), 440 (1998)
42. Zhu, X.: Persistent homology: An introduction and a new text representation for natural language processing. In: IJCAI (2013)