DIRECTION AWARE POSITIONAL AND STRUCTURAL ENCODING FOR DIRECTED GRAPH NEURAL NETWORKS

Yonas Sium 1  Georgios Kollias 2  Tsuyoshi Idé 2  Payel Das 2  Naoki Abe 2  Aurélie Lozano 2  Qi Li 1

1 Iowa State University
2 IBM Research, T. J. Watson Research Center

ABSTRACT

We propose a novel method for computing joint 2-node structural representations for link prediction in directed graphs. Existing approaches can be grouped into two families. The first group of methods learn structural embeddings of individual nodes in the entire graph through a directed Graph Neural Network (GNNs), and then combine pairs of the encodings to get a representation for the respective node pairs. Methods in the second group compute a representation of the subgraph enclosing the two nodes by employing GNNs initialized with positional encodings and consider these as their potential edge embeddings. Both families of link prediction techniques suffer from considerable shortcomings: The former fail to differentiate two distant nodes with similar neighborhoods; The latter, although properly appropriate for learning edge representations, adopt undirected GNNs, positional encodings, and subgraphs, so the edge direction signal is inevitably lost. Our proposal is also based on the idea of enclosing subgraphs, but the subgraphs are assumed directed, and directed Graph Neural Networks (GNNs) are used to learn their node encodings and initial positional embeddings are direction-aware. Our emphasis on capturing the direction of edges is reflected in superior performance in the link prediction task against baselines with undirected GNNs on symmetrized enclosing subgraphs and existing directed GNNs over a collection of benchmark graph datasets.

1. INTRODUCTION

In recent years graph neural networks (GNNs) have attracted considerable attention as an effective deep learning model for analyzing graph data[1]. Link prediction (predicting the existence of a link between two nodes) is one of the graph learning tasks that are broadly studied in different domains: e.g. in social networks [2], biological networks [3], recommender systems [4] and knowledge graphs [5]. In many applications the direction of the predicted edge is critical: In knowledge graphs, subject nodes point to object nodes; In citation graphs a derivative work cannot be pointed by earlier research; In causal graphs a node representing an effect should not precede its cause. Failure to capture edge directionality could also have disruptive effects on important downstream application tasks: e.g. conflicts in induced time orderings, blocked paths to root causes, distorted flow computations between node sets, etc.

The majority of existing GNN models learn the representation of a link by combining the representation learned of the two linked nodes. The linked nodes get their representations through message passing schemes [6] through aggregation of information from the neighboring nodes combined with their own features [7, 8, 9]. With this approach, nodes that have the same local neighborhood get the same representation, and hence, node based link prediction may fail for similar nodes with similar topological structure. Earlier works that proposed directed GNNs [10, 11, 12] adopted the above approach so they inherit this conceptual flaw.

In [13] the authors identified and formalized two types of encodings for graph nodes: structural and positional. GNN produce structural embeddings, which are highly effective for node labeling tasks: we want nodes with similar neighborhoods to be close in the representation space. As mentioned earlier, however, they fail to induce proper link representations: two nodes with identical local structure may be undistinguishable and hence lead to a wrong link prediction. They then show that joint 2-node structural representations are required for link prediction. On the other hand, positional encodings are commonly produced by lower-dimensional projections of the graph adjacency information. Such encodings for nodes that are close neighbors in the graph are close to each other, implying that links should most probably exist between nodes with minimal distances in the corresponding representation space. This is desirable for the link prediction task.

Our task is directed link prediction, so according to [13] we would need a scheme of generating either positional or joint 2-node structural encodings for graph nodes. Our approach blends these two perspectives in an innovative manner, while emphatically preserving the asymmetry arising from directed edges in the original graph. Notably: (i) we use GNN to learn structural encodings but do not use them directly in link prediction; (ii) we compute positional encodings but these are only used for GNN initialization.

** Structural encodings:** We extract a k-hop directed subgraph $G_k(u, v)$ around potentially linked nodes $u, v$, hence satisfying the joint 2-node structural representation requirement for link prediction. Directed GNN on the subgraph produce structural encodings for individual nodes in the subgraph, so using them pairwise for link prediction would be suboptimal. Hence, after computing the encodings of all nodes, we pool them into a single vector summarizing the subgraph as a whole and we assign it to the node pair.

**Positional encodings:** We initialize GNN based on spectral vectors which are positional encodings. They could be used directly for predicting links, but we only use them as input to the directed GNNs to produce structural encodings for the nodes. Note that spectral representation for a directed subgraph is different from its symmetrized version and thus our input encodings are direction-aware.

We provide theoretical justification that our directional-aware positional encodings are valid for learning structural encodings of pairs of nodes (edges) in directed graphs and empirically validate its effectiveness by testing it on the link prediction task on directed graphs. Experiments show our proposed approach improves over benchmark baselines including undirected models with distance-based encoding, existing methods for node representation in directed graphs and ex-
isting models enhanced with our proposed structural and positional encodings.

2. RELATED WORK

The idea of representing an edge by the encoding of its enclosing subgraph appears in Weisfeiler-Lehman Neural Machine (WLN) [14] and the follow up work in SEAL framework [1]. In WLN, special emphasis is placed on vertex ordering for the adjacency matrix of the input subgraph, using the Weisfeiler-Lehman (WL) coloring procedure for its computation. In SEAL, they introduce a distance-based encoding of nodes in the subgraph and they critically improve on WLN by leveraging GNN for node embedding learning. However, WLN and SEAL do not address directed graphs, which is our focus here. We use spectral encodings for input node labeling which are inherently direction-aware and we process enclosing directed subgraphs using directed GNN. A further generalization to multi-node representations, rather than two-node ones which is effectively our case with graph edges, is studied in [15].

Our focus is on capturing the directionality of predicted links, so we train directed GNN for computing the node encodings of the directed subgraphs. In particular, we use digraph convolution (DiGCN) and its multiscale generalization (DiGCNIB) from [11] and Magnet from [10]. The directed GNN we use are based on specific definitions of the Laplacian operator for directed graphs, followed by the standard graph convolution approach in [16]. In [11] they define a symmetric Laplacian for digraphs in terms of the transition matrix for personalized PageRank, which is then extended to higher-order proximity between graph nodes. In [10] a complex-valued, Hermitian Laplacian matrix is introduced. Another stream of research considers improving the power of GNN towards getting better representation for a link by fusing positional and structural encoding [15, 17, 18, 13], however not including link direction in their analysis or reported results.

3. MODEL

3.1. Description

Given a pair of nodes \((u, v)\) we (i) extract the directed subgraph enclosing of them; (ii) compute spectral, positional encodings for the directed subgraph nodes by leveraging truncated SVD; (iii) do the same with HITS algorithms; and (iv) concatenate these encodings and use them as initial inputs in training a directed GNN to compute subgraph node encodings, which are pooled at the end to represent the \((u, v)\) pair for the link prediction task. Figure 1 illustrates the overall approach.

(i) Directed Subgraph Extraction: We extract a \(k\)-hop directed subgraph \(G_k(u, v)\) around the endpoints \((u, v)\) of the potential directed edge \(u \rightarrow v\). Let \(N^+(i) = \{j \mid (i, j) \in E\}\) be the set of nodes that point to, \(N^+(i)\) the set of source nodes targetting \(i\) and \(N^-(i) = N^+(i) \cup N^-(i)\) the set of all incident nodes. We recursively define \(G_k(u, v) = (V_k(u, v), E_k(u, v))\), by setting \(G_0 = (V_0, E_0) = (\{u, v\}, \emptyset)\) and \(G_l = (V_l, E_l)\) where \(V_l = \bigcup_{j=1}^{l} N^-(j)\) and \(E_l = \{(i, j) \in E[i, j] \in V_l\}\), for \(l = 1, 2, \ldots, k\); \((u, v)\) dropped for notational convenience.

(ii) Direction Aware Positional Encoding Truncated SVD: We compute a variant of truncated SVD of the adjacency matrix \(A_k(u, v) \in \mathbb{R}^{n \times n}\) of subgraph \(G_k(u, v)\). More specifically, we compute an approximation \(U_d \in \mathbb{R}^{n \times d}\) of its left singular subspace corresponding to its \(d\) top singular values in \(\Sigma_d = \text{diag}[\sigma_1, \sigma_2, \ldots, \sigma_d]\), where \(\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_d\). Then the SVD-based encoding for node \(i\) in the subgraph will be the corresponding row of \(U_d \Sigma_d\). This is also the projection of the matrix row of \(A_k(u, v)\) for on the top \(d\) right singular vectors, which are columns of \(\Sigma_d\). Let us call this encoding \(\text{svd}[i] \in \mathbb{R}^d\).

(iii) Direction Aware Positional Encoding: HITS Our second, rank based positional encoding is the combination of the authority value \(a\) and the hub value \(h\). We compute the authority and hub scores for all nodes in \(G_k(u, v)\) using HITS algorithm [19], which is a popular method for ranking nodes in a directed network. These are scalar centrality values respectively capturing the importance of nodes as targets or as sources of directed links and ranking them accordingly; we organize them in vectors \(a, h \in \mathbb{R}^n\). \(a\) is computed by iteratively applying \(x \leftarrow A_i^k a_i x\), initialising with any vector \(x\) and normalizing at each step, where we have set \(A_k(u, v) = A_k\). Iterating \(x \leftarrow A_k^2 A_i x\) we converge to \(h\) for similar initialization and normalization strategies. Recalling that truncated SVD is \(A_k = U_k \Sigma_k \Sigma_k^T\), it follows that \(A_k^2 \Sigma_k^T = U_k \Sigma_k^T U_k\), so the iterative procedure for computing hub scores \(h\) is essentially the power method [20] used for finding the dominant eigenvector of matrix \(A_i A_i^T\), which is clearly the first column in left singular vectors \(U_k\). With analogous arguments, authority scores \(a\) is the first column in right singular vectors \(V_k\). Let us concateenate authority and hub scores for a node \(i \in V_k\) as \((a[i], h[i])\) and call this encoding \(\text{hits}[i] \in \mathbb{R}^2\).

(iv) Directed GNN-based Link Prediction: For all nodes \(i \in V_k(u, v)\), we concatenate HITS and truncated SVD-based positional encodings, \(\text{pe}([i]) = (\text{hits}[i] \mid \text{svd}[i]) \in \mathbb{R}^{d+2}\), use them as initial inputs for training a graph neural network (GNN), compute new node embeddings and finally pool them to yield a single vector representation for the whole subgraph \(G_k(u, v)\).

3.2. Theoretical Analysis

In [15] they prove that, with a valid labeling trick, a node-most-expressive GNN, defined as giving different representations to non-isomorphic nodes, can learn structural representations of a target node set \(S\). They detail two conditions for a node labeling scheme to actually be a labeling trick and a sufficient condition for each of them to hold. In addition, for directed graphs, where the order in the target set \(S\) is important they comment that it suffices to define a labeling trick respecting the order of \(S\). The two sufficient conditions read as (i) “the target nodes \(S\) have distinct labels from the rest of the nodes, so that \(S\) is distinguishable from others” and (ii) “the labeling function is permutation equivariant”. In our case \(S = (u, v)\) is the ordered set of two nodes, which we want to represent. We recall that the indicator function for \(S\), \(\chi_S(i)\), evaluates to 1 if node \(i \in S\) and is zero otherwise. \(\chi_S(i)\) is the simplest valid labeling trick [15], referred to as zero-one labelling trick.

**Theorem 3.1** \(\{\text{pe}([i]) \mid \chi_S(i)\}\) is a labelling trick.

**Proof:** The zero-one component \(\chi_S(i)\) in the encoding clearly distinguishes nodes in \(S = (u, v)\) from those in \(V_k(u, v) - S\), so the first sufficient condition holds. The second condition is also satisfied: the positional encoding component \(\text{pe}([i])\) is based on SVD and SVD is equivariant; zero-one component \(\chi_S(i)\) is also equivariant from [15]. More specifically, for any permutation \(\pi\) in the set \(\Pi_n\) of permutations over \(n\) symbols, \(\pi \in \Pi_n\), we can define the respective permutation matrix \(P\) with \(P_{(i, j)} = 1\) and vanishing elsewhere. Then the permuted adjacency matrix corresponding to a permutation of nodes will read: \(P A \Pi P^T = (P(U_d) \Sigma_d (P V_d))^T\) and this means that the node encodings are also permuted under the same \(\pi\). Q.E.D.
4. EXPERIMENTS

We conducted extensive experiments to evaluate the effectiveness of our positional encoding for directed link prediction on three GNN models designed for directed graphs and three other GNN models designed for undirected graphs. We also compare the performance with node based variants of the directed GNN models. Additionally, we compare our direction aware positional encoding with other structural encodings mainly designed for undirected graphs. We test on WebKB and citation network datasets, which are directed graphs.

4.1. Datasets and Baselines

We use directed WebKB (Cornell, Texas, Wisconsin) datasets [21]; nodes represent web pages, and edges represent hyperlinks between them. We also use the directed citation network (Cora-ML, CiteSeer datasets[22]); nodes represent articles while edges represent citations between them. We used the indegree and outdegree as the node features. The statistics of the datasets are summarized in Table 1.

**Baselines:** The baseline can be grouped into node based link prediction methods on GNNs designed for directed graphs, which are Magnet [10], DGCN [12], DiGraphIB [23] and subgraph based link prediction methods on GNNs designed for undirected graphs, which are GCN [16], GIN [24], GraphSage [25]. GIN and GraphSage are implemented by adapting the original paper [15]. In all baselines, ReLU nonlinearities are used in the output of each hidden layer. To aggregate the neighborhood sum-pooling aggregation is used.

4.2. Experimental Setup

We train three two-layer subgraph based GNNs designed for directed graphs for directed link prediction, which are Magnet, DGCN and DiGraphIB. We also train three two-layer subgraph based GNNs designed for undirected graphs for directed link prediction. In both cases, we used our proposed truncated SVD and Rank positional encodings.

For all the experiments, we use AUC as our evaluation metric. We repeat the experiments 4 times and report the average AUC results. We use 32 as the batch size and train for 500 epochs. We tune learning rate $\eta \in \{10^{-3}, 1, 2, 3\}$; $\eta = 10^{-3}$ gives the best performance. We randomly split the edges of each graph into train/validation/test sets at 85/15/10 percent. They are balanced sets of $k$-hop subgraphs around pairs of nodes $u, v$ which are either connected (positive samples) or not (negative samples). For the Magnet[10] model, we take the optimal hyperparameter value of $q$, which is provided in the original paper for each dataset.

4.3. Results and Discussion

The results of experiments using our truncated SVD and Rank positional encodings are shown in Table 2. The results when truncated SVD and rank positional encoding are used independently are shown in Table 3 and Table 4. In all cases the bold font is used to indicate the best performing models.

**Directed v.s. undirected GNNs with truncated SVD and Rank positional encodings:** In both cases the performance improved significantly when truncated SVD and Rank positional encodings are used as input features. The GNNs designed for directed graphs, however, are exhibiting better improvement than the GNNs designed for undirected graphs. For example, for CoraML and Cornell datasets, Magnet and DGCN directed GNNs are the top and second best performing, with 3.13% and 1.28% improvement on Cornell and 5.57% and 2.07% improvement on CoraML over GIN model designed for undirected graphs. Except for Texas dataset, Multitask directed GNN is performing the best for all datasets. This indicates that truncated SVD and Rank positional encodings are successfully capturing the directional structural information.

**Directed GNNs with both truncated SVD and Rank positional encodings v.s. without.** We compare the performance of Magnet, DiGraphIB, DGCN directed GNNs with v.s. without positional encodings. Table 2 shows clearly that using our encodings improves the performance by a large margin. This indicates that, though the models are designed to capture the directional structural information, the global directed structural information captured by our positional...
Comparison truncated SVD and Rank with DRNL positional encodings. When we use DRNL, which is designed for undirected graphs, on directed graphs, the results are the worst. This can be seen by comparing the results in Table 5 and Table 2. This implies that the existing positional encodings for undirected graph are not suitable for capturing the global directed structural information.

4.4. Ablation Studies

We conduct ablation studies to examine to what extent truncated SVD and Rank can independently direct the structural information of a directed graph. Comparison between Tables 2, 3 and 4 shows that combining truncated SVD and Rank, which are both variants of SVD, improves the performance further.

5. CONCLUSIONS AND FUTURE WORK

We combine specially designed, direction-aware, positional encodings of enclosing subgraph nodes with directed GNN learning in order to generate provably high-performance encodings for the link prediction problem in directed graphs. Experiments demonstrate the sustained efficacy of our approach. Predicting edges and their direction can be critical in graph applications and is a special case of the more general problem of co-encoding ordered node sets, which we plan to research within the context of higher-order representations (graph-products, tensorial adjacencies).
6. REFERENCES


