struc2vec: Learning Node Representations from Structural Identity

Overview:

Structural properties of the nodes and in a graph are critical features for many applications such as node classification, knowledge completion, link prediction etc. The conventional metrics used to represent these properties are node degree, clustering coefficients, neighborhood overlap, among many others. Even though such metrics precisely describe the properties of nodes in a graph, their scope is limited to the problem at hand and cannot be transferred to other domain/problem. Advancements in neural networks are providing methods to overcome such impediments by learning node properties in a continuous high-dimensional space.

Problem of interest:

Authors focus on the problem of developing an encoder function which can map every node in a graph to a vector in a continuous space, i.e., for a given graph

We want to develop encoder such that,

Published methodologies have adhered, in many cases, to a common two-step theme. First, generating set of sequences of nodes via random (or biased random) walk on the given graph. Followed by learning node representations from the set of sequences by maximizing dot product of two node embeddings for a pair of co-occurring nodes. While skip-gram structure remains prevalent for the second step, the first step i.e., random walk generations methods, have been vastly varied and experimented with.

In published studies, random walk generation is altered with few parameters such as, length of the sequence, maximum distance/hops traveled from the start node, exploration strategy (breadth first search, depth first search or combination of both) and total number of sequences generated. But generation of random walks on the original graph encodes more local information than global. In other words, truncating the length of random walk to, say three, would result in sequences where only neighbors appear together. Such methods cannot recognize similarity between the nodes which are far apart in network but structurally similar. For example, consider two published papers ‘u’ and ‘v’ that received 10 citations in last year. We can certainly see the similarity here! But if ‘u’ is a biological science paper and ‘v’ is an English literature paper then in the graph of published papers ‘u’ and ‘v’ will be placed far apart and never will appear in one random walk.

Authors of struc2vec present a unique random walk generation method which (highly likely) captures structurally similar nodes in a walk and hence, encodes global structural information in the node embeddings.

Strengths:

Encoding the global information in random walks by creating the multi-layered graph is the salient property of the method presented in this study. Struc2vec not only ensures a high similarity between structurally similar node embeddings for nodes placed far apart in a graph but also achieves a high similarity between vectors for nodes in disconnected components of the graph. Performance of the embeddings in clustering structurally similar nodes in Barbell graph and Karate network is shown to be much better than the other optional methods. Robustness of the method is tested successfully with edge removal exercise. Lastly, problem domains (air-traffic networks) are selected where the structural properties matter for node classification and in those scenarios struc2vec outperform other embedding methods.

Contribution:

In the pool of embedding methods aiming to develop encoder functions that map a node to a vector encapsuling the node and edge features only, struc2vec certainly bridges the gap to develop embedding representative of the structural properties of the node. The novel random walk sampling technique presented in the paper successfully captures enough information for node vector to represent structural properties in latent space. Development of such method proves critical in the operations problem where the structural properties of the node are equally important to the features of the nodes.

Limitations:

The primary limitation of the method presented in this study is that the similarity measured between nodes is strictly structural. The features of the node or that of the edges are not considered in calculating the similarity score. Hence, application of this method to heterogeneous graphs should be carefully done as the heterogeneity information will be lost during creation of the multi-layered graph. Even after considering improvements to lessen the computational expense, the presented method is significantly expensive for large graph, mainly because of creation of the multi-layered graph. Lastly, the node embedding method is transductive i.e., any unseen node or any addition of the node in the original graph cannot be readily mapped into embedding vector.

Appendix:

1. Method

The methodology presented in the paper can be broken into four main steps:

1. Measure structural similarity between node pairs

Authors measure the similarity between nodes ‘u’ and ‘v’ in a hierarchical manner. Starting from ‘u’ and ‘v’, lists of nodes are calculated at a 1-hop distance and both lists are sorted based on the degree of each node in the list. Similarity is calculated between two lists (not necessarily of the same size) based on the Dynamic Time Warping (DTW) algorithm. The process is repeated for k\* number of hops, where k\* is diameter of the graph. Therefore, we can write the expression as follows,

Where,

islist of nodes at k-hops from ‘u’ and is sorted version of the same list,

calculates similarity between ‘u’ and ‘v’ based on DTW

is called structural distance between ‘u’ and ‘v’ when considering k-hop neighbors

1. Construct a multi-layered graph from the measured structural similarities

For a fixed value of ‘k’, in previous step structural distance is calculated for every node pair in the graph. Therefore, we have a matrix of size |V| X |V| of distance values for a fixed ‘k’. From this distance matric a complete, undirected and weighted graph is created. Creation of such graph is repeated for each value of ‘k’. The edge weight between node ‘u’ and ‘v’ when considering neighbors at k-hops is calculated as follows,

This facilitates complete graphs. Every node in these graphs are connected to the same node in graphs for previous and next value of . These edges connecting different layers are given weights are follows,

1. Sample biased random walks

For a value of i.e., a layer in multi-layered graph, transitions probabilities are calculated as normalized exponentials of the structural similarity values, as follows,

In addition to the same layer transitions between nodes, transitions to ‘k+1’ and ‘k-1’ layers are also possible. With probability ‘1-q’ random walk decides to change layer with a probability calculated as follows,

1. Use sampled walks to learn node embeddings

The set of sequences generated from the biased random walk on the multi-layered graph created (in the second step mentioned above) are used to learn node embeddings. With a window size ‘w’ context of a node is defined and a skip-gram model is used to learn embeddings by maximizing the likelihood of the context nodes given a specific node.