A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications

This article makes a survey over more than 150 articles involving embedding to find definitions and standard that can extend the knowledge base of this part of the scientific study. They successfully find standards of input, embedding output and methods in between explaining what each part is and where they are most useful.

Problem

Problem 1 (Graph embedding). Given the input of a graph G = (V,E), and a predefined dimensionality of the embedding d (d << |V|), the problem of graph embedding is to convert G into a d-dimensional space, in which the graph property is preserved as much as possible. The graph property can be quantified using proximity measures such as the first- and higher-order proximity. Each graph is represented as either a d-dimensional vector (for a whole graph) or a set of d-dimensional vectors with each vector representing the embedding of part of the graph (e.g., node, edge, substructure).

Input graphs

Homogeneous Graph

Homogeneous graphs are graphs where all nodes are of one type and all edges are of one type. If the graph is unweighted and undirected then everything can be treated equally. If there is a weight on an edge, then nodes should be closer together in an embedded graph. In graphs with direction nodes which depend on each other should be closer together

Heterogeneous Graph

Heterogeneous graphs are graphs that can contain nodes and/or edges of multiple types. this is often seen in community question answering sites, multimedia networks or knowledge graphs where edges represent a specific connection and nodes represent a specific entity. To embed these graphs a Minimizing Margin-Based Ranking Loss function is used to minimize loss when embedding and finding the similarities and position of the embedding vectors.

Graph with Auxiliary Information

Graph with Auxiliary Information are graphs with nodes or edges that has more information than just a type. this information can be labels, attributes, node features, information propagation and knowledge base. This type graph is often hard to embed because of all the additional information that needs to be preserved but at the other hand there is much more information that can be used to find similarities.

Graph Constructed from Non-Relational Data

These graphs are often made form data that might have connections that might be found through means such as knn , association or other algorithms meant to classify. This type of graph meads a lot of the challenges the others does with the added challenge of trying to find meaningful connections.

Output graphs

Node Embedding

Node embedding is the act of finding closed between nodes that has something in common and find their closeness.

Edge Embedding

Edge embedding are more complicated than node embedding because it strives to compress the information of the nodes and edges between them into vectors and find the closeness between the relations.

Hybrid Embedding

Hybrid embedding tries to take substructures in a graph both nodes and edges, combining them into a vector representing a given structure this allows an analyst to find structures that might be in opposite ends of a graph and compare their similarities.

Whole-Graph Embedding

Whole-Graph embedding takes graphs and makes a vector representation of it. This allows for comparisons between whole graphs. This can be extremely hard to do because each of the embedding methods try to be reversable and by representing a whole graph as a vector there can be losses.

GRAPH EMBEDDING TECHNIQUES

Each embedding method has some advantages and disadvantages that can be considered when embedding is needed.

Matrix Factorization

When using matrix factorization as an embedding method then there is a huge advantage of considering global proximity because this gives an analyst a more precise estimation of one vector distance to all other vectors. The disadvantage of this method is that the method scales linearly and can therefor be quite time consuming to use and the matrixes will be quite big and consume a lot of space.

Deep Learning

Most deep learning methods are very effective and robust meaning that the can give good answers and are wary hard to get malware into. If the initialization and training processes are done right then the system wont need any feature engineering, this in turn can also be a problem because of the fact that the training samples need to be picked so that the algorithm can lean what is right and wrong and this training data can in many instances be optimized but one needs to remember not to be too specific because then there can be a problem of overfitting. There is also a problem if close mindedness that occurs because of the walk the method uses it will only se the information in the order it is given which might not always give a precise answer.

Edge Reconstruction Based Optimization

Within this frame of embedding there are three methods that are used “maximizing edge reconstruction, minimizing distance-based loss or minimizing margin-based loss” each of these methos try in ther way to ensure that the original graph input can be reconstructed form the embedded graph. This is done because these are relatively easy to train so that they can be installed fast, but the optimization can only use local information at each point in a graph or ranked node pairs.

Graph Kernel

The Graph kernel method is a form of work that start in a kernel and walks through a subgraph comparing each kernel that are chosen. This method is mostly used for whole graph embedding and are mostly tested with homogeneous graphs or graphs with auxiliary information. This method is good because it will only represent and compare the structures that are desired to be compared. The method can have problems with substructures that are not independent and will grow the embedding space exponentially.

Applications

This article lists a few application points embedding can be useful for.

Node Embedding

Node embedding can be used in:

- Node classification using SVM, logistic regression and/or KNN.
- Node clustering id the graph is unlabeled the embedding will reveal clusters most work uses k means for clustering
- Node recommendation/retrieval/ranking by finding association in the graph

Edge Embedding

Edge embedding can be used for:

- Triple classification a method of finding if a relation a-b goes through relation r
- Link prediction tying to predict if and what a link between two nodes could be

Hybrid and Whole-Graph Embedding

Here a user of imbedding will be able to classify graphs or subgraphs in a lower dimension than their original. Another use is for visualization because our minds can be limited its easier to visualize embedded systems.

Conclusion

They conclude that they were able to find formal definition of graph imbedding and that they from the proves proposed two taxonomies for the setting and embedding concepts. They explain some different types of embedding at where they are mostly used.