Technical Perspective – No PANE, No Gain: Scaling Attributed Network Embedding in a Single Server

Aidan Hogan
DCC, Universidad de Chile & IMFD
ahogan@dcc.uchile.cl

The machine learning community has traditionally been proactive in developing techniques for diverse types of data, such as text, audio, images, videos, time series, and, of course, matrices, tensors, etc. "*But what about graphs*?" some of us graph enthusiasts may have asked ourselves, dejectedly, before transforming our beautiful graph into a brutalistic table of numbers that bore little resemblance to its parent, nor the phenomena it represented, but could at least be shovelled into the machine learning frameworks of the time.

Thankfully those days are coming to an end.

The area of *graph representation learning* [1] has enjoyed growing momentum, spurred on not only by graph enthusiasts, but also by applications relating to transport networks, social networks, biological networks, and more recently, knowledge graphs. Within the area, we find a fork in the road. To the left, we find researchers developing novel learning frameworks over the topology of the graph itself, à la *graph neural networks*. Veering right, we find works on various forms of *graph* or *network embeddings* that transform elements of the graph into vectors, matrices, etc., that can be fed into the more traditional machine learning frameworks used for downstream tasks, but now in a principled and elegant way that is more respectful to the original graph and what it represents.

We can find two common limitations, however, when network embedding techniques are applied in real-world scenarios. First, they may struggle to cope with large graphs (let's say in the order of billions of edges). Second, in order to learn representations of more complex graph data, we often need to learn representations for features like edge direction, edge labels, node attributes, etc.

The paper "*No PANE, No Gain: Scaling Attributed Network Embedding in a Single Server*" by Yang et al. [2] addresses these two key issues of *scalability* and handling *complex features* in the context of network embeddings. The model they adopt is that of an *attributed network*, which features directed edges, where nodes can additionally be associated with weighted attributes. This approach thus narrows the gap between network embedding techniques that typically address undirected graphs and graph models popular in practice with features like directed edges and node attributes. An attributed network could be used to model, for example, a citation graph, where nodes represent edges, directed edges indicate citations, and attributes indicate (weighted) topics for each paper.

The stated goal of the paper is then to learn vector representations of nodes within an attributed network that capture "node–attribute affinities", i.e., attributes reachable from a node through zero or more hops in the graph. Both forward and backward affinities are considered. This can capture, for example, what affinities a paper (a node) has with which topics (attributes) based not only on its own topics, but also those of the papers it cites, and the paper they cite, etc., (forward affinities); as well as the topics of the papers that cite it, and the papers they are cited by, etc. (backwards affinities). These affinities are captured through forward and backward random walks (with restart) that explore neighbours in both directions with a given stop probability at each step, recording a random attribute of the node where the walk stops. Two vectors – for forwards and backwards affinity – are then learnt for each node, and one vector is learnt for each attribute, such that the dot product of a given forward-affinity node vector and a given attribute vector approximate the corresponding affinity seen through random walks; the same is applied for backwards affinity.

The pressing challenge now is to train vectors at large scale, where it is computationally challenging, for example, to jointly optimise both forwards and backwards affinities together. The paper thus proposes two novelties to address this issue. The first is to approximate both forward and backward affinities by replacing the random walks with a fixed number of iterations of a power method-style algorithm to approximate the target affinities in both directions. The second is to propose an efficient method for computing the necessary embeddings that first applies a matrix factorization of the forward and backward affinity matrices produced by the previous step in order to approximate the corresponding embeddings separately, and using these embeddings to initialise a cyclic coordinate descent method, which then jointly optimises the embeddings according to both forward and backward affinities. Finally, the authors show how the proposed algorithm, which they call PANE, can be effectively parallelised over multiple cores.

Experiments compare PANE with a wide range of baselines on a wide range of graphs, for a variety of downstream tasks, including attribute inference (predicting the attributes of a node), link prediction (predicting which node a give node links to), and node classification (predicting a label for a node). PANE is shown to largely outperform the competitors in terms not only of quality metrics (such as average precision), but also in terms of scale and performance, with only one other competitor able to cope with the largest graph considered: the Microsoft Academic Graph, with 59 million nodes, 2 thousand attributes, and close to a billion edges.

The paper thus advances the state of the art in a number of ways that should please the average graph enthusiast, showing not only how node attributes can be taken into account when learning embeddings, but also how such an approach can be made to scale by using approximations, clever initialisation techniques, and parallelism. A pressing question for the future then: how could these techniques be adapted to support attributes *and* edge labels?

REFERENCES

[1] William L. Hamilton. 2020. *Graph Representation Learning.* Morgan & Claypool Publishers. https://doi.org/10.2200/S01045ED1V01Y202009AIM046
[2] Renchi Yang, Jieming Shi, Xiaokui Xiao, Yin Yang, Juncheng Liu, and Sourav S. Bhowmick. 2020. Scaling Attributed Network Embedding to Massive Graphs. *Proc. VLDB Endow.* 14, 1 (2020), 37–49. https://doi.org/10.14778/3421424.3421430