Deep Link Prediction for Graphical Database

Aristotelis-Angelos Papadopoulos[1] **Collin Purcell**[2] **Devershi Purohit**[2] **Ishank Mishra**[2]
[1]Department of Electrical & Computer Engineering
[2]Department of Computer Science, University of Southern California
{aristotp, collinpu, dupurohi, imishra}@usc.edu

1 Introduction

In recent years, deep learning has achieved great success in the fields of vision, speech, and natural language understanding. The ability of deep learning to extract underlying patterns from complex, large-scale and high-dimensional data is well recognized. Many real-world applications are built on a graphical database, and thus utilize graphs for storing data. The graph here means a directed, attributed multi-graphs with data stored as nodes and relationships(links) between nodes stored as edges. Graphical databases are ubiquitous and often consist of hundreds of millions of nodes and relationships. There is rich information embedded in the complex topology of these graphs that can be leveraged when doing inference on the data stored in them. As a result, utilizing deep learning to extract this information has gained a lot of traction in the research community.

Our research contribution in alignment to this is two-fold. First, we propose a Graph-Based Classification Model that uses nodal features as well as the structure of a node's local sub-graph to predict links between graph nodes. Second, we propose that our model can be implemented as a generalized classification model capable of predicting links for a variety of graphical data. The basic idea behind our model is utilizing modern deep learning techniques to learn relational information from local sub-graphs and use it to predict the relationship between the target nodes.

2 Related Work

Neural networks that operate on graphs, and structure their computations accordingly, have been developed and explored extensively for more than a decade under the umbrella of "graph neural networks" [5], but have grown rapidly in scope and popularity in recent years.

Models in the graph neural network family, e.g. [9], have been explored in a diverse range of problem domains, across supervised, semi-supervised, unsupervised, and reinforcement learning settings. These models have been effective at tasks thought to have rich relational structure, such as visual scene understanding tasks [10] and few-shot learning [3]. They have also been extensively used to reason about knowledge graphs [2, 6]. For more applications of graph neural network models, see [1] and references therein.

Recently, [4] introduced the message-passing neural network (MPNN), which unified various previous graph neural network and graph convolutional network approaches by sequentially updating edge attributes, node attributes and global attributes of the graph and transmitting the information after each update. In a similar vein, [11] introduced the non-local neural network (NLNN), which unified various "self-attention"-style methods by analogy to methods from computer vision and graphical models for capturing long range dependencies in signals.

Graph neural network models usually consist of Graph Network (GN) blocks and can be divided into three main categories depending on the task that needs to be served. Node-focused and graph-focused GNs use the nodes attributes and the global attributes as outputs respectively. On the other hand, in the spirit of [8, 7], our main scope in this project is the design of an edge-focused neural network in order to predict the existence of an edge between two nodes as well as its corresponding label.

3 Problem Formulation

Our multi-graph has been instantiated with Neo4j and consists of 28 million nodes representing such things as companies, people, industry verticals, and locations and 38 million edges representing relationships such as Works-At, Located-In, and Employs. See Figure 1. Our goal is to create a generalized architecture to predict financial anomalies such as corporate bankruptcy or fraud as well as solve entity linkage problems caused by importing from disparate data sources. To do this, we frame all problems as edge classification problems. For entity linkage we will be predicting the likelihood that there exists an edge with the label "same-as" between two nodes. For anomaly detection problems we will be predicting the likelihood that there is an edge between the node being profiled and a meta-node that we create for each anomaly we are predicting with the label "is-a". For example, to predict a company's risk of bankruptcy we will predict the label of the edge between that company's node and a bankruptcy node that we have created. A single input sample for our model will be the 2D adjacency matrix of the combined local sub-graphs for that company and the bankruptcy node with the combined nodal features for each node that share an edge aligning in a third dimension. The result is a 3D tensor with dimensions $(n_1, n_2, ..., n_n) \times (n_1, n_2, ..., n_n) \times (f(n_1, n_1), f(n_1, n_2), ..., f(n_n, n_n))$ where n represents the nodes of the local sub-graph and $f(x, y)$ represents the combined features for nodes x and y or all 0's if nodes x and y do not share an edge. The output we will try to predict is the 2D adjacency matrix with the correct edge labels in each cell. In this formulation, we include the null "no edge" label in the set of edge labels for the cases that no edge exists.

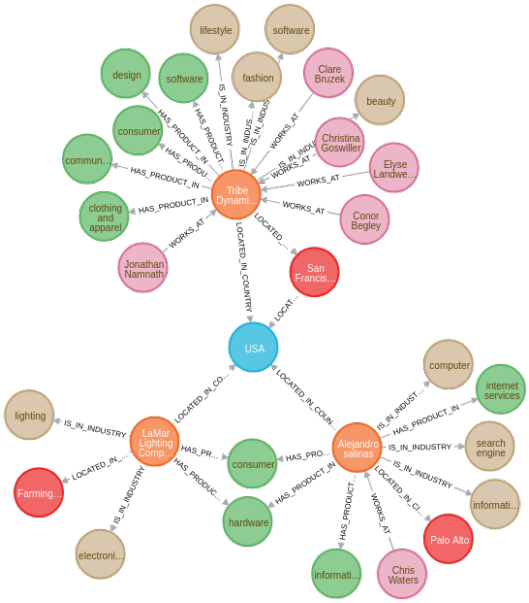

Figure 1: Example subg-raph of 3 companies who are located in the USA. A node's color represents its type and the text on an edge is that edge's label. In total there are 28 million nodes and 38 million edges.

4 Milestones

Mar 6 - Mar 20	Dataset preparation and preprocessing
Mar 21 - Apr 5	Model exploration, feature engineering
Apr 5 - Apr 15	Model refinement, parameter tuning, improving predictions
Apr 15 - Apr 23	Testing and evaluation

5 Approach

Our intuition when approaching this problem was that nodal features alone were not enough to adequately predict complex anomalies like fraud or bankruptcy and that the relationships between a node and its local sub-graph are hugely important. For example, if we are profiling a company for

potential fraud it would be critical to make our model aware of all connections and similarities to known fraudulent companies. That is why we are using the meta-node structure. Each meta-node will have all known cases of its corresponding query connected to it via an edge. This way when we create the 3D adjacency matrix for an anomaly query, all known examples of that anomaly will be included in that sample.

To implement our model we will create a prediction pipeline with a cypher phase where we query the multi-graph to get both node's local sub-graphs, an embedding phase where we transform each nodal feature into a numerical vector, an alignment phase were we merge the two local sub-graphs and form the 3D adjacency matrix, and finally the inference phase were we pass that sample through the model. For our model, we plan on using an architecture similar to those used for state of the art image segmentation since our input and output tensors will have the same shape for their 2D adjacency matrices. Unlike these models, we propose to use many $(1 \times 1 \times f(x, y))$ dimensional CNN filters for the first layer. Our intuition is that since these filters will only be convolving over the combined nodal features of two nodes that share an edge, their output would encode those two node's similarity or dissimilarity. For creating the validation and test sets we plan on removing a subset of the edges with labels we want to predict along with either of its head or tail nodes. That way when we reintroduce the edge into the graph the model hasn't already learned that there is no edge between its head and tail nodes.

Once we have optimized this base formulation of our approach there are several extensions we have in mind. One such optimization for anomaly detection is to add time stamps to each known sample and use that time stamp as the key in an attention model that will extract whatever data it determines to be pertinent from a separate vector of global economic features. For example, global factors like market liquidity, interest rates, and inflation could alone be deterministic in predicting if a company is likely to go bankrupt or not, so including them in the model would be essential. By time stamping each sample and using an attention model that has the entire history of these global factors to choose from, our model can intelligently incorporate them into our pipeline similarly to how attention is used in choosing which word in the input is most relevant to the next outputted word in machine translation.

References

[1] P. W. Battaglia et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv: 1806.01261*, 2018.

[2] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, pages 2787–2795, 2013.

[3] V. Garcia and J. Bruna. Few-shot learning with graph neural networks. In *International Conference on Learning Representations (ICLR)*, 2018.

[4] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv: 1704.01212*, 2017.

[5] M. Gori, G. Monfardini, and F. Scarcelli. A new model for learning in graph domains. In *International Joint Conference on Neural Networks*, 2005.

[6] T. Hamaguchi, H. Oiwa, M. Shimbo, and Y. Matsumoto. Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[7] J. Hamrick, K. Allen, V. Bapst, T. Zhu, K. McKee, J. Tenenbaum, and P. Battaglia. Relational inductive bias for physical construction in humans and machines. In *Proceedings of the 40th Annual Conference of the Cognitive Science Society*, 2018.

[8] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel. Neural relational inference for interacting systems. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

[9] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated graph sequence neural networks. In *International Conference on Learning Representations (ICLR)*, 2016.

[10] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battagli, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems*, 2017.

[11] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.