The Hong Kong Polytechnic University
SingularityNet

Causal analysis in Networks via Deep Neural-Symbolic Learning

Andres Suarez-Madrigal, Tesfa Yohannes, Ben Goertzel

November 15, 2019

Contents

1 Introduction 3

2 Embeddings 4

1 Introduction

The current project explores a method to predict anomalous events in the dynamics of a network, via finding their causes. The main idea in the method is to find the grammar of the network – its implicit dynamical structure– in an unsupervised manner, starting from telemetry data from its different components. If such a grammar can be obtained, then it would be possible to predict an anomaly (with a certain confidence) when observing dynamics that preceed such a state in the grammar. This would be similar to a situation in language processing where, after observing a sequence of words that form a sentence's subject (e.g. determiner-adjective-noun), we can expect a verb to follow.

This project will leverage SingularityNet's Unsupervised Language Learning (ULL) pipeline, which attempts to find the grammar implicit in a given corpus of sentences. In order to process the network data with such pipeline, the process can be divided into three stages:

1. Abstract the network dynamics by converting the state of the network at each timestep into a real-valued vector (or a set of vectors).
2. Using symbolic dynamics techniques, convert the dynamics of the network embedding vector(s) into a sequence of symbols; these sequences would be functionally equivalent to a natural language corpus in the ULL project.
3. Apply a suitable version of the ULL pipeline to the sequences obtained in step 2 (the sentences), in order to learn the network grammar.

The first step could be done in a number of ways. One popular and successful way to create distributed representations is using Neural Networks. Deep Neural Networks like graph2seq have been used to abstract graphs, with some success [2].
TESFA, YOU CAN ADD SOME STUFF HERE.

Another approach for creating embeddings is to use the features as they come from the telemetry data. In Section 2 we follow a similar processing as done by Putina et al.[1] and visualize the resulting vectors.

For step 3, we need to adapt the ULL pipeline to our purposes: instead of having a natural language sentence where every word is presented to the algorithm at the same time, we would be dealing with a continous stream of tokens representing the state of the network at different times. In particular, the parser used in the ULL pipeline needs to be replaced by one that handles a continous stream of tokens. For this purpose, we implemented a continuous version of the MST-parser proposed by Yuret [3][1]

[1]Code available at https://github.com/glicerico/stream-parser

2 Embeddings

This is a test

References

[1] A. Putina, D. Rossi, A. Bifet, S. Barth, D. Pletcher, C. Precup, and P. Nivaggioli. Telemetry-based stream-learning of BGP anomalies. In *Proceedings of the 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks - Big-DAMA '18*, pages 15–20, Budapest, Hungary, 2018. ACM Press.

[2] S. B. Venkatakrishnan, M. Alizadeh, and P. Viswanath. Graph2seq: Scalable Learning Dynamics for Graphs. *arXiv:1802.04948 [cs]*, Feb. 2018. arXiv: 1802.04948.

[3] D. Yuret. Discovery of Linguistic Relations Using Lexical Attraction. *arXiv:cmp-lg/9805009*, May 1998. arXiv: cmp-lg/9805009.