Related Works Draft 1

There are a few related works in applying machine learning to a networks problem. In Beverly [1], the author described several learning approaches for attacking various networks traffic problems with minimal dataset. In particular, learning was useful in capitalizing on under-utilized information and inferring behavior reliably. Using a HMM, Beverly was able to infer TCP congestion based on the IP address bits. Beverly’s view agrees with our work, as we both believe that we can dig deeper into the dataset. By utilizing statistical methods to aid humans in extracting the intricacies of data, we improve inference performance.

Another major related paper to this particular project is Yang's Measurement and Modeling of User Transition Among Networks [5]. This paper created one particular 3-state hidden Markov model that's evaluated by measuring the distribution of cost function of signaling a network attachment / detachment. While the paper serves as a decent exploration baseline, we explore deeper on some aspects, definitions, and assumptions. For instance, the Yang paper assumed all users form one uniform group. We expand the work by including several clustered user classes. We expand the work in both the modeling space.

Probabilistic programming is an ongoing research area. Generative learning approaches can analyze complex scenes more richly and flexibly, but have been less widely embraced due to their slow inference speed and problem specific engineering to obtain robust results [6]. One of the trending topics lately in probabilistic programming is in creating a generic inference engine that facilitates optimization of parameters. This inference engine facilitates code implementation for different models. Many current applications are under developed for probabilistic programming. Picture and Venture are two such probabilistic programming languages conceived at MIT. Both languages focus on the development of a general inference engine. With such, a user not fluent in the language of statistics can still implement probabilistic models. We refrained from using those languages as they are alpha versions and have higher learning curves. Instead, we explored PyMC3, a stable python package that’s written with a general inference engine. We chose this option because we want to explore more models without having to implement the inference methods ourselves [2, p.4].

In Tenenbaum [7], current applications and limitations for probabilistic programming are discussed, with a vision for hierarchical structural learning. That is, not only can the machine learn about the optimal parameters for the model, it can also do hill climbing in modeling space to identify the most appropriate model. With enhanced storage capacities and computational cheapness, hierarchical structural learning can facilitate data mining further by saving the scientists time to conceive the most logical model.

[1] Beverly, Robert E. Statistical learning in network architecture

[2] Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence Nature, 28 May 2015

[5] Yang, et al. Measurement and Modeling of User Transitioning Among Networks

[6] Kulkarni, et al. Picture: A Probabilistic Programming Language for Scene Perception

[7] Tenenbaum, et al. How to Grow a Mind: Statistics, Structure, and Abstraction