Using Reinforcement Learning to Guide Random Walks in Very Large Graphs

Mention these, but glance over them. They mostly use RNN and Attention Networks; seems only vaguely RL

Deep RL

1.) Graph Classification using Structural Attention. Lee, Rossi, & Kong, 2019

- Uses rewards to motivate agent to correctly classify graph with shortest random walk. Does this by representing the current node, and visited nodes with a RNN and Attention model. Once this can be done reliably, a linear layer, which the paper calls a "policy" attempts to map the vector repr. to a class the graph may be in. Seems like a stretch to me. Seems like it's just normal deep learning but whatever.

2.) Deep Learning on Graphs: A Survey. Zhang, Cui & Zhu, 2015 (Just section 6: Graph Reinforcement Learning)

- Most examples here are just applied Graph deep learning (GCNs and RNNs). Cites [1] and other GCN methods. Also cites [6] which I'll go into in-depth. Perhaps can use this paper as a transition between deep-learning in general on graphs, and Deep RL

3.) Graph-based skill acquisition for reinforcement learning. Mendonca, Matheus RF, & Ziviani, 2019 (Survey)

- Defines deep RL. Q(s,a) is now a function, mapping every <s,a> to a real vector. Because tabular methods are unreasonable for larger state spaces, use CNN or RNN to map <s,a> to vector, and do something with this vector to create a policy.

Go over these papers more in-depth.

Rules/Policy-based RL: (less black box)

3.) Graph-based skill acquisition for reinforcement learning. Mendonca, Matheus RF, & Ziviani, 2019 (Survey)

- Mentions [4,5] . Do I need to cite a survey if I'm only talking about papers it mentions?
- Actually, this paper defines graph-based RL techniques quite nicely. It defines a "centrality measure" (which is what [4,5] are) and why they are important (allows an easy reward function for graphs, and if we id particular nodes of interest via the CM, allows a pseudo-reward function)

4.) Automatic Skill Acquisition in Reinforcement Learning Agents Using Connection Bridge Centrality Moradi, Shiri, & Entezari, 2010. Communications in Computer and Information Science. 51-61

- On the right track, but too complicated. An extension on [5] but very expensive. I think [5] on its own, while simplistic and not as powerful as [4] would get the job done (if a heuristic must be specified. The "future work" section of [6] suggests that maybe adversarial techniques could be used to automatically build reward methods)

5.) Using Relative Novelty to Identify Useful Temporal Abstractions in Reinforcement Learning Simsek & Barto, 2004. ICML

- Best looking heuristic for use in something like node2vec (maybe should cite that also?). Simple, and elegant. I think it's an excellent rule to use for something on-policy because it's cheap, and works well in an episodic framework.

6.) DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning Xiong, Hoang, & Wang 2018 -- EMNLP

- I think this is the most interesting use of RL for graphs. The authors specify their own rules, which could work equally well for node2vec's paths. They use it on a KB, which can really be thought of as a highly heterogeneous D-graph. They view finding new relations between entities in it as a MC process and use an on-policy MC agent to do this. Their results look promising, and I'm curious if it would work with graphs generally, not just KBs.

Other:

7.) node2vec: Scalable Feature Learning for Networks Grover & Leskovec, 2016 -- KDD

- Need to cite this just so I can reference how I plan to use RL instead of random walk in this algorithm