SparRL: Graph Sparsification via Deep Reinforcement Learning

Ryan Wickman
rwickman@memphis.edu
University of Memphis

Figure 1: The SparRL model architecture consists of the node encoder, edge encoder, and action-value head. The input to the model includes the subgraph \( H_t \), the degrees of the nodes \( d_H \), the ratio of edges still in the graph \( r_H \), and the 1-hop neighborhood of the set of nodes in \( H_t \), \( N_t \). The node encoder uses a GAT [23] on the 1-hop neighborhood of each node embedding to create a new node embedding which is then combined with its degrees and the ratio of edges. The edge encoder combines each pair of nodes that represent an edge. The action-value function produces the q-value for each edge.

CCS CONCEPTS
- Computing methodologies → Reinforcement learning; Machine learning;
- Networks → Network reliability.

KEYWORDS
graph sparsification, deep reinforcement learning, machine learning

ACM Reference Format:
Ryan Wickman. 2022. SparRL: Graph Sparsification via Deep Reinforcement Learning. In Proceedings of the 2022 International Conference on Management of Data (SIGMOD ’22), June 12–17, 2022, Philadelphia, PA, USA. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3514221.3520254

1 PROBLEM AND MOTIVATION

Graph sparsification concerns data reduction where an edge-reduced graph of a similar structure is preferred. It has been proven to be an effective technique in a variety of application domains, such as power grid management [28, 30], integrated circuit simulation [29], and influence maximization [11, 18]. Existing sparsification methods are mostly sampling-based [6, 19, 20], which generally introduce high computational complexity and lack of flexibility for a different reduction objective. To address these challenges, we present SparRL, the first general and effective reinforcement learning-based framework for graph sparsification. SparRL can easily adapt to different reduction goals and promise graph-size-independent complexity. Extensive experiments show that SparRL outperforms all prevailing sparsification methods in producing high-quality results concerning a variety of objectives. As graph representations are very versatile, SparRL carries the potential for a broad impact.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

SIGMOD ’22, June 12–17, 2022, Philadelphia, PA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9249-5/22/06.
https://doi.org/10.1145/3514221.3520254

2 RELATED WORK

Quite a few edge-reduction-enabled learning-based algorithms [5, 24, 25, 31] have been developed for graph representation learning, which typically generate a new graph in aim of enhanced resilience or structural feature preservation. But many of them introduce new edges to the graph, which compromises the graph sparsification objective on many real-world networks where establishing new edges/connections is resource-intensive (e.g., road networks). The most relevant study to ours is NeuralSparse [31], which learns a graph representation that needs to go through graph neural networks for downstream classification. Due to this constraint, classical analytic benchmarks such as community detection or shortest path computing from the traditional graph sparsification studies are missing in [31]. In comparison, SparRL outputs a sparsified graph where existing graph analytic algorithms can be directly applied.

3 SPARRL FRAMEWORK

Framework Overview The overall goal of our work is to find an edge-sparsified graph \( G' = (V, E') \) that approximates the original graph \( G = (V, E) \) measured over some performance metric.

We treat this as an episodic task, where edges are sequentially pruned from \( G \). Each timestep \( t \), a subgraph \( H_t = (V, E_H) \) is sampled from \( E_t \) and SparRL’s action \( a_t \) consists of choosing an edge to prune from \( E_{H_t} \). This continues until \( T \) edges are pruned from the graph which will produce \( G' \). We describe this process in Algorithm 1.

During training, we exploit the simplicity of the environment by randomly pruning \( T_p \sim \mathcal{U}(0, |E| - T) \) edges from \( G \) to produce the initial sparsified graph \( G' \). This allows the SparRL agent to start in any part of the state space \( S \), without initially pruning \( T_p \) edges. Thus, requirements of efficient exploration of the state space is removed from the behavior of the target policy. When evaluating the policy, the initial state is set to the original graph \( S_1 = G \) and the number of edges to prune is set to an exact number.

Policy Learning We use Double DQN [22] to represent the SparRL sparsification policy that is parameterized by a deep neural network.
Student Abstract

Figure 2: Performance of sparsification methods over (a) PageRank, (b) community structure, and (c) shortest path distance preservation. SparRL outperforms all other methods in all experiments and metrics.

Algorithm 1: SparRL Framework

```
input : G = (V, E), T (the number of edges to prune)
output : the sparsified graph G' = (V, E')

G' ← clone G
for t = 1 to T do
    H_t ← Randomly sample |E_H| edges from E'
    d_H_t ← Degrees of nodes in H_t;
    η_t ← |E_H| / |E_t|
    N_t ← 1-hop neighborhood of nodes in H_t
    q_values ← fSparRL(H_t, d_H_t, η_t, N_t)
    a_t ← argmax_q q_values
    Prune edge a_t from G'
end for
return G'
```

The policy is trained over batches of sampled trajectories from a prioritized replay buffer [16]. The model architecture is shown in Figure 1 which composed of the node encoder, edge encoder, and action-value head. The model approximates the Q-value function:

\[ f_{SparRL}(H_t, d_{H_t}, \eta_t, N_t) = Q(a_t, a_t^1, \ldots, Q(a_t, a_{|E_H|}). \quad (1) \]

Each edge of the subgraph is independently run through the network, so the subgraph length |E_H| is not constrained by the network. Therefore, any number of edges can be considered to be pruned at each timestep during test time. During training, the pruned edges are sampled according to an ε-greedy exploration strategy, but during test time the edge with the largest q-value is pruned.

**Time Complexity.** As the subgraph length |E_H| is independent of the size of the graph, the time complexity of pruning T edges is \( O(|E_H|T) \). However, it is highly parallelizable as it can make a prediction on the entire batch of edges in the subgraph in parallel and can easily be distributed over several machines.

4 RESULTS AND CONTRIBUTION

We validate the effectiveness of SparRL over a number of classic graph analysis workloads using several real-world graph datasets from a variety of domains: Twitter [10], YouTube [12], Amazon [26], Email-Eu-Core [27], and CiteSeer [17]. We compare SparRL with a wide range of conventional sampling-based sparsification methods: Random Edge (RE), Local Degree (LD) [8], Edge Forest Fire (EFF) [8], Algebraic Distance (AD) [4], L-Spar (LS) [15], Simmelian Backbone (SB) [13], and Quadrilateral Simmelian Backbone (QSB) [14].

Table 1: SparRL compared against t-spanner for various stretch values t over CiteSeer. (x%: edge kept ratio)

| Method   | t=3     | t=4     | t=8     | t=16    | t=32    |
|----------|---------|---------|---------|---------|---------|
| SparRL   | 0.0082  | 0.0054  | 0.0405  | 0.1187  | 0.1911  |

4.1 Effectiveness of SparRL

**PageRank Preservation.** PageRank serves a critical centrality metric for many ranking-based graph applications. The results in Figure 2 (a) show SparRL is outperforms all other methods measured over the Spearman's ρ metric.

**Community Structure Preservation.** We use the Adjusted Rand Index (ARI) [9] to measure the effectiveness of SparRL on preserving the community structure of a graph by comparing non-overlapping ground truth communities to those found using the Louvian method [3] at multiple edge-kept ratios. The results in Figure 2 (b) shows SparRL outperforms all other methods at preserving the community structure.

**Shortest Path Distance Preservation.** The results in Fig. 2 (c) show that SparRL outperforms at preserving 8196 randomly sampled single-pair-shortest-path (SPSP) distances as it has the least average distance increase. When a path becomes unreachable in the sparsified graph, we set the SPSP between the two nodes to |V|. t-spanner [2, 21] provides a way to sparsify a graph while preserving the geometric distance between a pair of nodes at most t times of the original distance. We conduct an experiment study on comparing the performance of SparRL and a popular spanner algorithm given in [1] by using the NetworkX [7] implementation. The results measured over the average SPSP distance increase are given in Table 1, where we show SparRL can do better than the approximate t-spanner algorithm.

4.2 Contribution

To the best of our knowledge, SparRL is the first task-adaptive and effective reinforcement learning-based framework for graph sparsification. Through extensive experiments, we show that SparRL can extend to a variety of datasets and sparsification objectives. We have made SparRL’s code publicly available at https://github.com/rwickman/SparRL-PyTorch.
REFERENCES

[1] Surender Baswana and Sandeep Sen. 2007. A simple and linear time randomized algorithm for computing sparse spanners in weighted graphs. *Random Struct. Algorithms* 30, 4 (2007), 532–563.

[2] Joshua D. Batson, Daniel A. Spielman, Nikhil Srivastava, and Shang-Hua Teng. 2013. Spectral sparsification of graphs: theory and algorithms. *Commun. ACM* 56, 8 (2013), 87–94.

[3] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[4] Jie Chen and Ilya Safro. 2011. Algebraic distance on graphs. *SIAM Journal on Scientific Computing* 33, 6 (2011), 3465–3490.

[5] Victor-Alexandru Darvariu, Stephen Hailes, and Miro Muñozes. 2020. Improving the Robustness of Graphs through Reinforcement Learning and Graph Neural Networks. *arXiv:2001.11279* [cs.LG]

[6] Wai Shing Fung, Ramesh Harinarayan, Nicholas J. A. Harvey, and Debmalya Panigrahi. 2019. A General Framework for Graph Sparsification. *SIAM J. Comput.* 48, 4 (2019), 1196–1223.

[7] Arci A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Proceedings of the 7th Python in Science Conference*. Cael Varoquaux, Travis Vaught, and Jarrod Millman (Eds.), Pasadena, CA USA, 11 – 15.

[8] Michael Hamann, Gerd Lindner, Henning Meyerhenke, Christian I. Staudt, and Dorothea Wagner. 2016. Structure-preserving sparsification methods for social networks. *Social Network Analysis and Mining* 6, 1 (2016), 22.

[9] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of classification* 2, 1 (1985), 193–218.

[10] Jure Leskovec and Julian Mcauley. 2012. Learning to Discover Social Circles in Ego Networks. In *Advances in Neural Information Processing Systems*. F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.), Vol. 25. Curran Associates, Inc.

[11] Michael Mathioudakis, Francesco Bonchi, Carlos Castillo, Aristides Gionis, and Antti Ukkonen. 2011. Sparsification of Influence Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 529–537.

[12] Alan Mislove, Massimiliano Mocen, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and Analysis of Online Social Networks. In *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC’07)*. San Diego, CA.

[13] Bobo Nick, Conrad Lee, Pádraig Cunningham, and Ulrik Brandes. 2013. Simmelian backbones: Amplifying hidden homophily in facebook networks. In *Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining*, 525–532.

[14] Arlind Nocaj, Mark Ottmann, and Ulrik Brandes. 2014. Untangling hairballs. In *International Symposium on Graph Drawing*. Springer, 101–112.

[15] Venu Satturi, Srinivasan Parthasarathy, and Yiee Ruan. 2011. Local Graph Sparsification for Scalable Clustering (SIGMOD ’11). Association for Computing Machinery, New York, NY, USA, 721–732.

[16] Tom Schaütz, John Quan, Ioannis Antonoglou, and David Silver. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952* (2015).

[17] Prithviraj Sen, Gaileio Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI magazine* 29, 3 (2008), 93–93.

[18] Xiao Shen, Pu-Lai Chung, and Sitong Mao. 2017. Leveraging Cross-Network Information for Graph Sparsification in Influence Maximization. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 801–804.

[19] Daniel A. Spielman and Nikhil Srivastava. 2011. Graph Sparsification by Effective Resistances. *SIAM J. Comput.* 40, 6 (2011), 1913–1926.

[20] Daniel A. Spielman and Shang-Hua Teng. 2011. Spectral Sparsification of Graphs. *SIAM J. Comput.* 40, 4 (2011), 981–1025.

[21] Shang-Hua Teng. 2016. Scalable Algorithms for Data and Network Analysis. *Found. Trends Theor. Comput. Sci.* 12, 1–2 (2016), 1–274.

[22] Hadi Van Hasselt, Arthur Guez, and David Silver. 2016. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.

[23] Petar Velicković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[24] Lu Wang, Wenchao Yu, Wei Wang, Wei Cheng, Wei Zhang, Hengyuan Zha, Xiaofeng He, and Huifeng Chen. 2019. Learning robust representations with graph denoising policy network. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 1378–1383.

[25] Hang-Yang Wu and Yi-Ling Chen. 2020. Graph Sparsification with Generative Adversarial Network. *arXiv:2009.11736* [cs.SI]

[26] Jaewon Yang and Jure Leskovec. 2015. Defining and Evaluating Network Communities Based on Ground-Truth. *Knowl. Inf. Syst.* 62, 1 (Jan. 2015), 181–213.

[27] Hao Tian, Austin R. Benson, Jure Leskovec, and David F. Gleich. 2017. Local Higher-Order Graph Clustering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 555–564.

[28] Xueqian Zhao, Zhuo Feng, and Cheng Zhou. 2014. An efficient spectral graph sparsification approach to scalable reduction of large flip-chip power grids. In *The IEEE/ACM International Conference on Computer-Aided Design, ICCAD*. IEEE, 218–223.

[29] Xueqian Zhao, Junde Han, and Zhuo Feng. 2015. A Performance-Guided Graph Sparsification Approach to Scalable and Robust SPICE-Accurate Integrated Circuit Simulations. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* 34, 10 (2015), 1639–1651.

[30] Zhaoqiang Zhao and Zhuo Feng. 2017. A Spectral Graph Sparsification Approach to Scalable Vectorless Power Grid Integrity Verification. In *Proceedings of the 56th Annual Design Automation Conference, DAC*. ACM, 681–686.

[31] Cheng Zhong, Bo Zong, Wei Cheng, Dongjin Song, Jingchao Ni, Wenchao Yu, Huifeng Chen, and Wei Wang. 2020. Robust graph representation learning via neural sparsification. In *International Conference on Machine Learning*. PMLR, 11458–11468.