Improving Core Resilience of Network under Random Edge Deletion

Ricky Laishram
Syracuse University
rlaishra@syr.edu

Ahmet Erdem Sarıyüce
University at Buffalo
erdem@buffalo.edu

Tina Eliassi-Rad
Northeastern University
eliassi@neu.edu

Ali Pinar
Sandia National Laboratories
apinar@sandia.gov

Sucheta Soundarajan
Syracuse University
susounda@syr.edu

k-cores have emerged as an important concept for understanding the global structure of networks, as well as for identifying 'central' nodes within a network. The k-core [2] of a network is the maximal sub-graph in which every node has at least k neighbors, where the nodes in the higher cores are considered to be more central within the network. k-cores represent cohesive subgroups of nodes, and have been used in a broad variety of important applications, such as studying the structure of internet networks, predicting the function of proteins, or understanding the evolution of networks.

It is often valuable to understand the *resilience* of the k-cores of a network to attacks where edges are deleted (i.e., damaged communications links). *We define the n, p-core resilience of a network G, denoted by $\mathcal{R}_n^p(G)$, as the rank correlation between the ranking of the top n% nodes (ranked by core number) in the original network to that of the network after p% edges have been removed uniformly at random.* We additionally define an aggregate measure called *n-core resilience* ($\mathcal{R}_n(G)$). We define $\mathcal{R}_n(G)$ *as the mean* $\mathcal{R}_n^p(G)$ *as we vary p from* 0 *to* 100.

We examine the problem of characterizing the core resilience in terms of the network's structural features: in particular, which structural properties cause a network to have high or low core resilience? To measure this, we introduce two novel node properties - *Core Strength* and *Core Influence*. Based on these features, we propose an algorithm to determine which edges should be added to a network to improve its resilience, under the constraint that the nodes' core numbers do not change. This has important applications in complex networks (such as technological networks), where edges between nodes can drop randomly and without warning, and we wish to improve the resilience of the network while preserving its overall core structure.

1 Proposed Core Metrics

Let $\Gamma(u)$ and $K(u)$ denote the neighbors of a node u and its core number respectively. Then, we define the **Core Strength (CS)** of node u as,

$$CS(u) = |\{v : (v \in \Gamma(u)) \wedge (K(u) \leq K(v))\}| - K(u)$$

Intuitively, the CS of a node is a measure of how robust that node's core number is with respect to random edge deletions. A node with a high CS is a node with many redundant connections (i.e., many connections to other nodes with equal or higher core number), and is less likely to drop its core number if edges are deleted.

The **Core Influence (CI)** of node u measures how much that node influences the the core number of other nodes $v \in V$ where $K(u) > K(v)$. Let V_δ be the set of nodes v such that v has fewer than $K(v)$ neighbors whose core number is $K(v)$. Let $\Delta_>(v)$ be the neighbors of v that have core number greater than $K(v)$, and let $\Delta_<(v)$ be the neighbors of v with core number less than $K(v)$. Then, we define the Core Influence of u as,

$$CI(u) = \sum_{v \in V_\delta \cap \Delta_<(u)} \frac{CI(v)}{|\Delta_>(v)|}$$

We initialize each CI value to 1, and then perform updates starting from nodes with lowest core numbers to highest core number. If the nodes with high CI also have high CS, the important nodes are less likely to loose their core numbers on edge removal. We thus expect $\mathcal{R}_n^p(G)$ be be high in such case.

2 Increasing Network Resilience

It may often be the case that one is given a network (e.g., a computer or other technological network), and wishes to improve the resilience of that network against randomly deleted or dropped edges without changing the core numbers of nodes in the network (in order to, for example, preserve centrality values of nodes). Our initial results above give us insights into how we can accomplish this: if we want to improve the core robustness of a network by adding edges, we should add these edges in a way to bolster the nodes with high CI; i.e., give them higher CS.

Given a network $G = \langle V, E \rangle$ the first step is to find the edges that can be added to the network without changing the core number of any node. Let E' be the set of edges that do not exist in G. The size of E' is on the order of $|V|^2$. This is clearly too many edges to check, so we need a method to quickly filter out the edges that would change the core number if added to G. We accomplish this filtering by adapting the purecore-based method described in [1], which examines the endpoint of each potential edge (the 'purecore' of a node u is the set of nodes that have the same core number as u and could be affected by a change in the core number of u).

Once this preliminary filtering is finished, we determine which edges to add to G. To do this, we use the CI and CS measures to assign a priority to each $e \in E'$. As discussed earlier, we want to minimize the nodes with low CS and high CI. For each edge $e \in E'$, we thus assign a *priority value* $p(e)$,

$$p(u,v) = \begin{cases} \frac{CI(u)}{CS(u)} & \text{if } K(u) < K(v) \\ \frac{CI(v)}{CS(v)} & \text{if } K(u) > K(v) \\ \frac{CI(u)}{CS(u)} + \frac{CI(v)}{CS(v)} & \text{if } K(u) = K(v) \end{cases}$$

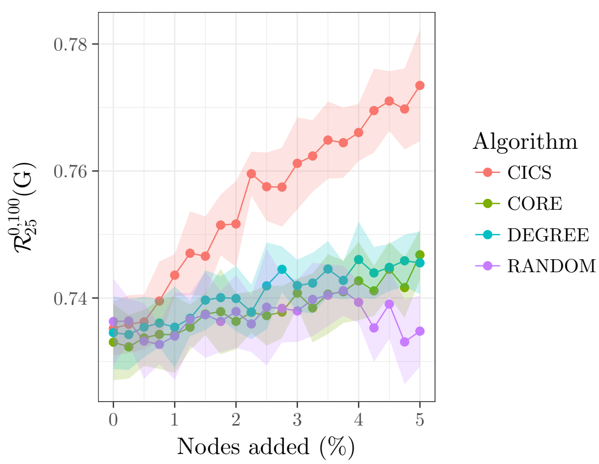

Figure 1: Comparison between average $\mathcal{R}_{25}^{0,100}(\cdot)$ vs Nodes added on the `Tech_Router` network. The red plot is the results of our algorithm (*CICS*), and the green, blue and purple plots are the results from baselines *CORE*, *DEGREE* and *RANDOM*.

To test our method, we added $0\% - 5\%$ new edges to the `Tech_Router` network[1] using our approach. For comparison, we consider three baseline methods where the edges in E' are added (1) randomly (*RANDOM*), (2) in decreasing order of the sum of the degrees of the endpoints (*DEGREE*), and (3) in decreasing order of the sum of the core numbers of the endpoints (*CORE*).

In Figure 1, we present the results of our experiments on the `Tech_Router` network. Due to space constraints, we present results on only one network, but observed similar results on other datasets. We observe that when adding edges based using our proposed method, the core resilience improves much more than with the baselines methods. In the case of *RANDOM*, the resilience may even decrease when adding edges (investigating the cause of this behavior is one of our avenues for future work).

We are currently working on several related problems, including translating our work to the case where nodes may be deleted. Additionally, we observe that in some networks the $\mathcal{R}_n^p(\cdot)$ is non-monotonic with respect to p. Why do some networks have this behavior, and which structural properties of the network can be used to predict this behavior?

References

[1] A. E. Saríyüce, B. Gedik, G. Jacques-Silva, K.-L. Wu, and Ü. V. Çatalyürek. Streaming algorithms for k-core decomposition. *Proceedings of the VLDB Endowment*, 6(6):433–444, 2013.

[2] S. B. Seidman. Network structure and minimum degree. *Social networks*, 5(3):269–287, 1983.

[1] Obtained from `www.networkrepository.com`.