Correlation of Node Importance Measures: An Empirical Study through Graph Robustness

Mirza Basim Baig
Stony Brook University
Department of Computer Science
mbaig@cs.stonybrook.edu

Leman Akoglu
Stony Brook University
Department of Computer Science
leman@cs.stonybrook.edu

ABSTRACT

Graph robustness is a measure of resilience to failures and targeted attacks. A large body of research on robustness focuses on how to attack a given network by deleting a few nodes so as to maximally disrupt its connectedness. As a result, literature contains a myriad of attack strategies that rank nodes by their relative importance for this task. How different are these strategies? Do they pick similar sets of target nodes, or do they differ significantly in their choices?

In this paper, we perform the first large scale empirical correlation analysis of attack strategies, i.e., the node importance measures that they employ, for graph robustness. We approach this task in three ways; by analyzing similarities based on (*i*) their overall ranking of the nodes, (*ii*) the characteristics of top nodes that they pick, and (*iii*) the dynamics of disruption that they cause on the network. Our study of 15 different (randomized, local, distance-based, and spectral) strategies on 68 real-world networks reveals surprisingly high correlations among node-attack strategies, consistent across all three types of analysis, and identifies groups of comparable strategies. These findings suggest that some computationally complex strategies can be closely approximated by simpler ones, and a few strategies can be used as a close proxy of the consensus among all of them.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining

Keywords

graph mining; node centrality measures; correlation analysis

1. INTRODUCTION

Large scale networks are prevalent; the Web, the power grid, social networks, etc. Studying and exploiting properties of such graphs can lead to insights with real world impact. Therefore, there has been a wide array of research on the study of real world graphs [2, 16, 21, 22, 25].

One of the most fundamental operations in network analysis is identifying the relative importance of nodes. It is well known that the nodes in scale-free real-world networks are not equally important. For example, the seminal work by Albert *et al.* found that scale-free networks, while being quite robust to random failures, are highly vulnerable to targeted attacks that select and destroy a core set of critical nodes in the network. This finding stimulated a large body of research on the response of real-world networks to various attack strategies [7, 8, 9, 23, 39].

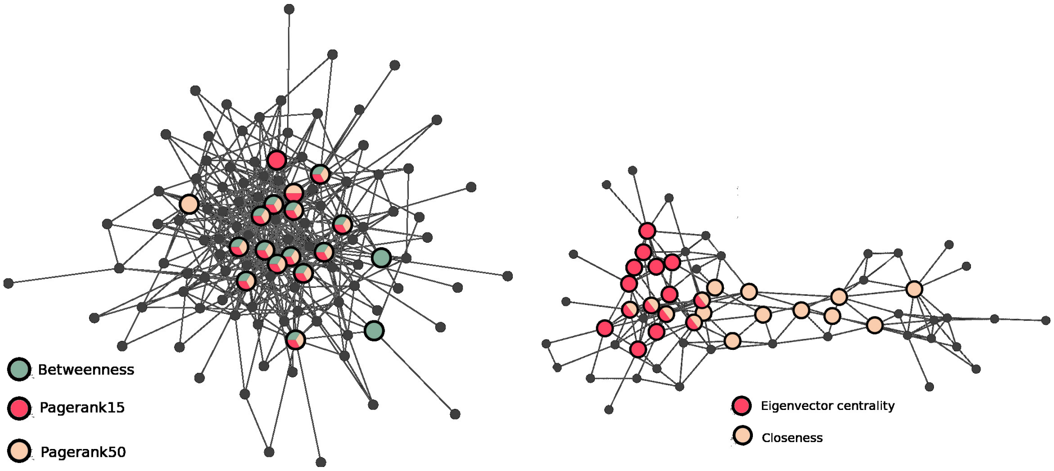
The amount of research on the attack-tolerance of real graphs is amplified due to studies across various disciplines, including physics, mathematics, computer science, and sociology. As a result, literature contains a myriad of node-based attack strategies for graph robustness. A vast majority of these strategies are heuristics which select their target nodes based on various measures of importance [2, 3, 15]. Most heuristics aim to target the most central nodes and thus employ different notions of node centrality, such as targeting nodes by highest degree or highest betweenness centrality. Different heuristics also incur varying computational cost; degree centrality can be computed in linear time while betweenness is quadratic in graph size [5].

Although a plethora of heuristic strategies have been proposed, there exists no study to date that compares and contrasts them to analyze how similar or different they are from one another. For instance, given a network and a pair of strategies, it is not well-understood whether they would pick similar sets of nodes to target or differ considerably in their choices. As an example, consider Figure 1 (a) which shows a graph in which top 15 nodes selected by three different heuristics (Betweenness and two variants of PageRank centrality) are marked. As one can notice, there is significant overlap among their node sets. On the other hand, Figure 1 (b) shows a graph in which the two sets of top 15 nodes, respectively selected by Closeness and Eigenvector centrality, share little overlap. A natural question is then: Which node-attack strategies are highly correlated to one another?

The goal of this paper is to provide the first empirical assessment of correlations between heuristic attack strategies, i.e., the node importance measures that they employ, for graph robustness. We aim to reduce the long list of existing heuristics into several groups containing the ones with correlated output. We expect that such correlations can be leveraged to approximate computationally complex heuristics with simpler correlated ones, as well as to approximately find the consensus among all the heuristics. We use three different methodologies when comparing the heuristic strategies, and aim to identify observations that hold across all three types of correlation analysis.

Our work utilizes and extends a correlation analysis framework proposed by Abrahao *et al.* to compare various graph clustering algorithms [1]. This framework was also used

WWW'15 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2743055 .



(a) Significant overlap among node sets

(b) Small overlap among node sets

Figure 1: Comparing top 15 nodes picked by heuristic strategies in two example real-world graphs (a) Word adjacencies network, (b) Dolphin social network from http://www-personal.umich.edu/ mejn/netdata/. (best viewed in color)

by Soundarajan *et al.* to analyze the correlations between graph similarity measures [29]. With the aforementioned goal in mind, we make the following contributions:

- We present the first large-scale and thorough correlation analysis of centrality-based graph attack strategies. Our study involves 15 strategies on 68 real-world graphs spread across 4 different categories (social, biological, infrastructure, and information networks).
- We measure the correlation between the strategies in three different ways: (1) by comparing their *rankings* of nodes in the networks ordered by importance for disruption, (2) by comparing the *characteristics* (or type) of the top nodes they target, and (3) the *disruption dynamics* they cause on the network, i.e., how the network disintegrates when the nodes are removed successively in their ranked order.
- Our analysis reveals the following findings: (1) the heuristic strategies, i.e., different node centrality measures employed by those, are surprisingly well correlated, (2) there exist groups of comparable strategies with strong correlation across all three measurements and the majority of the networks, and (3) a few strategies produce a ranking that is very close to the consensus ranking among all of the strategies.
- These findings offer guidelines for selecting suitable attack strategies and present approximation opportunities, where computationally expensive strategies can be closely approximated by comparable cheaper ones, and a few strategies can be used to find a close proxy of the consensus among all of them.

2. BACKGROUND AND METHODOLOGY

We compare and contrast 15 node-based graph attack strategies, with varying time complexities (Table 1).

To analyze the correlation among the strategies, we use three different analysis methodologies. First, we compare the strategies based on their overall ranking of nodes. Specifically, we consider the similarity between two strategies to be the weighted correlation between their rankings.

Second, we adopt a similar approach to Abrahao *et al.*'s [1] (that characterized graph clustering algorithms) to compare the strategies based on the characteristics of the top nodes that they target. Given the top few nodes, we leverage our earlier work [14] to characterize each node with a vector of representative graph-centric features. We quantify the similarity between two strategies through the matching and the separability between their feature vectors.

Table 1: Node-based attack strategies. k: node removal budget, n: number of nodes, m: number of edges, d: average degree, t: maximum number of iterations, C: topology dependent constant, D: depth to which ecc is computed, T: random walk length, α: damping factor, $O(\cdot)$: complexity of finding (top) k nodes.

	Id	Abbr.	Description	bigO
Random	1	r	Random node	$O(k)$
	2	rn	Random neighbor of a randomly picked node	$O(k)$
	3	rw10	Most visited node in a random walk of length $T = 10$	$O(kT)$
	4	rw50	Most visited node in a random walk of length $T = 50$	$O(kT)$
Local	5	deg	Highest degree	$O(m)$
	6	lcc	Highest local clust. co-efficient [38]	$O(nd^3)$
	7	ecc	Highest extended clustering co-efficient [12]	$O(nd^{2+D})$
Dist.	8	rad	Lowest radius [13]	$O(n^3)$
	9	cc	Highest closeness centrality [26]	$O(n^3)$
	10	betw	Highest betweenness centrality [5]	$O(nm)$
Spectral	11	eig	Highest eigen-vector centrality	$O(nC)$
	12	pr15	Highest PageRank [27] (α=0.15)	$O(mt)$
	13	pr50	Highest PageRank [27] (α=0.50)	$O(mt)$
	14	katz	Highest Katz index [17]	$O(mt)$
	15	comm	Highest self-communicability [10]	$O(n^3)$

Third, we use the ranking provided by a strategy to remove the nodes one by one from a given graph. We track the response of the graph to these removals, and then compare two strategies based on the gradual impact that they incur on the graph connectedness.

We apply each of these correlation analysis methods on 68 real-world networks from four different domains (social, biological, infrastructure, and information). We look for strong correlations among strategies which hold across all three analysis techniques and a large body of the networks.

In the following, we refer to the corresponding methods as Rank-C, Top$k$-C, and Response-C respectively (C for correlation). Since all three methods use the ranking of nodes by the strategies, we briefly discuss how we obtain these rankings. We then describe our methodologies in detail.

```
Input: a set of graphs $\mathcal{G}$, a set of attack strategies $\{s_1, \ldots, s_M\}$, number of target nodes $k$
for each graph $G \in \mathcal{G}$ do
  for each strategy $s_i$ do
    Rank all nodes in $G$ into $r_i$ using $s_i$
    Extract recursive structural features for top $k$ nodes in $r_i$ using ReFeX [14]
  end for
  Compute similarity between each $s_j$, $s_k$ pair by:
    (Rank-C) Weight-Tau ($r_j$, $r_k$)
    (Top$k$-C) Bi-Match or SVM-Sep between $2k$ ReFeX feature vectors from $s_j$ and $s_k$
  Hierarchically cluster $s_1, \ldots, s_M$ by similarity
end for
Output clusters that appear in majority of graphs in $\mathcal{G}$
```

Figure 2: Comparing & clustering attack strategies based on (Rank-C) the overall ranking of all nodes, and (Top$k$-C) the characteristics of top k nodes.

2.1 Generating ranked lists of nodes

In this work we consider both randomized and non-randomized attack strategies. Non-randomized strategies allocate a score to the network entities in a given graph. Those scores are often associated with centrality, such as eigen-vector or closeness centrality (Table 1). As such, we sort the nodes based on their scores to generate a ranking.

Randomized strategies, on the other hand, aim to speed up the selection even further; e.g., Random-Neighbor picks a random node and then a random neighbor of it, trying to approximately pick a high degree node in scale-free graphs. Such strategies do not compute scores for the network entities. To create their ranked lists, we run them on a given graph until all nodes are picked and sort them by the order they have been picked during the course of the run.

2.2 Comparing methods by Rank-C & Top$k$-C

Once we obtain the ranked lists from all strategies for a given graph, we ask two questions, respectively in Rank-C and Top$k$-C. First, how do the strategies rank the nodes? We apply the Weight-Tau technique to answer this question. Second, how does the set of top-k nodes picked by each attack strategy compare in terms of structural characteristics? We use the SVM-Sep and Bi-Match techniques for answering the second question. In both cases, we reduce the ranked lists generated previously to a single $M \times M$ similarity matrix S, where M is the total number of attack strategies. Each row and column of the similarity matrix corresponds to an attack strategy and the entry S_{ij} gives a similarity score between strategies i and j. The summary of these methods is given in Figure 2.

2.2.1 *Comparison by Weighted Tau (* Weight-Tau *)*

Rank-C uses the Weight-Tau technique of Vigna [36] to compare the overall ranked lists generated by the attack strategies. Their technique produces a similarity score $\in [-1, 1]$ between two ranked lists through a generalization of the Kendall's τ [19], where ties are carefully accounted for. Moreover, the similarity score is biased toward agreements higher in the ranked lists. That is, strategies that agree on the nodes closer to the top of their lists are assigned a higher score. This measure is particularly suitable for our setting where ties in scores are common and top ranked nodes have higher impact on robustness, i.e. matter more.

2.2.2 *Comparison by class separability (* SVM-Sep *)*

Top$k$-C compares the strategies based on the characteristics of the top-k nodes they pick. As such, it considers two strategies to be similar if they target the same *kind* of nodes. For characterization, it extracts recursive structural features [14] for each of the k nodes from a strategy. This maps each node to a vector in a feature space.

Given this transformation, we frame the problem of comparing two strategies as a class separability problem. We associate label 0 to the k feature vectors from strategy i, and label 1 to the k feature vectors from strategy j. Using this labeled data, we train a binary classifier.[1] We compute class probabilities based on 5-fold cross-validation. We then sum the probability mass of points labeled 0 being in class 0 and points labeled 1 being in class 1 and take their average as the measure of class separability. The better the classification, the higher the separability between the classes, i.e., strategies. As such, one minus separability yields a similarity score $\in [0, 1]$ between two strategies.

2.2.3 *Comparison by matching (* Bi-Match *)*

In Top$k$-C, we also compute a similarity between two strategies by finding a maximum matching between their k feature vectors. We start by creating a complete bi-partite graph in which nodes on the left represent k feature vectors from strategy i and nodes on the left depict k feature vectors from strategy j. We connect every pair of nodes in this graph with weighted edges, where weights are equal to one minus the Canberra distance [20] between the two feature vectors that map to the end points of the edge.

Next, we compute the maximum weight matching on this bi-partite graph. Briefly, a matching tries to map each left node to one and only one right node. Among all possible matchings, the one with the highest total edge weight between the matched nodes is the maximum matching. In our case, a matching with high weight implies that the entities picked by two strategies are comparable in their characteristics. By averaging the k edge weights in the matching, we compute a similarity score $\in [0, 1]$.

2.2.4 *Finding correlated attack strategies*

We employ Rank-C using Weight-Tau, and Top$k$-C using both SVM-Sep and Bi-Match to generate a similarity matrix S containing the pairwise similarities among the strategies, for each of 68 graphs in our study. In order to condense the information contained in the similarity matrices to a more manageable size, we perform complete-linkage hierarchical clustering on each similarity matrix to produce a dendogram and identify clusters that emerge in the lower (i.e., high similarity) levels of the dendogram. Each cluster in a dendogram corresponds to a group of similar strategies. We consider a cluster to be significant only if it emerges in more than 50% of the input graphs. The results for the clusterings are presented in the experiments section.

2.3 Comparing methods by Response-C

Response-C compares the attack strategies by analyzing the effects they cause on a graph when the nodes are successively removed from the graph in the rank order provided by each strategy. The conjecture is that if two strategies are similar, they would cause similar disruption on a target

[1] We train a linear SVM [35] and set the hyper-parameters by performing a grid search over 10-fold cross validation.

```
Input: a set of graphs $\mathcal{G}$, set of strategies $\{s_1, \ldots, s_M\}$
for each graph $G \in \mathcal{G}$ do
    for each strategy $s_i$ do
        Rank all nodes in $G$ using $s_i$
        Remove nodes one-by-one in rank order
        Record connectedness measure (1) fraction of GCC
        (giant connected component) size or (2) $\lambda_1(G)$
        Compute average robustness $A_i$
    end for
end for
Rank graphs in $\mathcal{G}$ into $R_i$ based on $A_i$ from each $s_i$
Compute similarity between each $s_j$, $s_k$ pair using
Weight-Tau ($R_j$, $R_k$)
Hierarchically cluster $s_1, \ldots, s_M$ by similarity
```

Figure 3: Comparing & clustering attack strategies based on (Response-C): the incurred disruption on graph connectedness.

graph. The summary of this approach is given in Figure 3.

In particular, we take the ranked list of nodes by a strategy and remove them from the target graph one by one while monitoring the value of a selected robustness measure. In this work, we consider two widely used connectedness measures; (1) fraction of the giant connected component (GCC) size and (2) the largest eigenvalue of the adjacency matrix (λ_1) of the graph. We recompute the measure every time a node is removed from the target graph and aggregate the values into a single resilience score $A = \frac{1}{N}\sum_{i=0}^{N} f(i)$, where N is the number of nodes and $f(i)$ is the value of the connectedness measure after i nodes have been removed from the graph. As such, A_s can be seen as the "average resilience" of a graph when attacked by strategy s; the lower the A_s, the less resilient the target graph.

Each strategy then ranks the given set of graphs by their resilience to the specific attack. Following on our earlier conjecture, similar attack strategies would cause similar disruption and hence provide a similar resilience ranking of the graphs. Once again, we reduce our problem of comparing attack strategies to comparing ranked lists, where we use Weight-Tau as a measure of similarity.

2.4 Finding a consensus strategy

Besides studying the similarities among attack strategies, another question we pose in this work is whether there exists an attack strategy that can be used as a proxy for a consensus among all of them. To answer this question we compute a Kemeny-Young consensus [18] of the ranked lists produced by the strategies in Rank-C and Response-C, and look for strategies that are consistently close to the consensus.

3. EXPERIMENT RESULTS

In this study we used 68 real-world graphs, spread across 4 categories (14 social, 12 biological, 36 infrastructure, and 6 information). The sizes of the graphs vary from a few thousand to a million edges. All datasets and more details can be downloaded at https://github.com/basimbaig/robust14.

3.1 Correlation analysis (Rank-C & Top$k$-C)

We start by looking at how attack strategies rank the nodes in a graph (Rank-C) and whether the structural characteristics of these nodes overlap (Top$k$-C). We set k in Top$k$-C to the number of 1% of nodes in each graph, as we have graphs of varying sizes. Table 2 presents the clusters obtained by applying our analysis framework presented in Figure 2. Note that we only show clusters that appear in at least 50% of our graphs. Even though we study a large set of strategies, we find that a majority of them are correlated to at least one other strategy. In particular, we find three clusters of highly correlated node-based strategies, namely {PageRank15, PageRank50, Betweenness}, {Katz, Eigen-vector}, and {Degree, Closeness}.

Table 2: Clusters obtained using Weight-Tau , SVM-Sep , and Bi-Match for node-based attack strategies.

Clusters (Weight-Tau)	# Graphs
1. {PageRank15, PageRank50, Betweenness}	(67/68)
2. {Katz, Eigen-vector}	(56/68)
3. {Closeness, Communicability}	(40/68)
4. {Degree, Radius}	(39/68)
Clusters (SVM-Sep)	# Graphs
1. {PageRank15, PageRank50, Betweenness}	(64/68)
2. {Katz, Eigen-vector}	(54/68)
3. {Closeness, Degree}	(35/68)
Clusters (Bi-Match)	# Graphs
1. {PageRank15, PageRank50, Betweenness}	(62/68)
2. {Katz, Eigen-vector}	(52/68)
3. {Closeness, Degree}	(44/68)

We note that the clustering results in Table 2 hold irrespective of the methodology used to compute the similarity scores. This implies that our findings most likely reflect the underlying correlations amongst the attack strategies. An exception we notice is {Degree, Radius} which appears only in our Weight-Tau results. The reason this cluster did not show up in Bi-Match and SVM-Sep is because the number of graphs where this cluster appears for those is below our threshold but is nevertheless reasonably high (29/68 for SVM-Sep , and 31/68 for Bi-Match).

Figures 4 shows how these clusters actually appear. For brevity we only show the average heatmaps in the figures but for generating the results we went through the clustering results of each graph and strategy pair.[2] Each heatmap shows the average similarity scores across all the graphs where a specific cluster appears. The heatmap marked 'All' simply shows an average of the scores across all the graphs.

We notice that the strategies in the same cluster are strongly correlated in the graphs where they appear. What is more, the clustering structure still remains visible when the similarity matrices are averaged across 'All' graphs. That is, the correlations do not "wash away" when all the graphs are considered. We also notice from the 'All' heatmaps that the attack-strategies are overall well correlated. The average pairwise similarity of the strategies is 0.88 with 0.09 standard deviation.[3]

To illustrate the difference of clustered strategies, Figure 5 shows the distribution of raw similarity scores for (i) pairs of clustered and (ii) pairs of randomly picked strategies. We see a clear difference in scores between the two, irrespective of the method used to compute the similarity scores.

[2] We present heatmaps for Bi-Match due to space limit. Those for SVM-Sep and Weight-Tau are similar.

[3] Results are similar for SVM-Sep : 0.69 (0.22), and Weight-Tau : 0.36 (0.30). Note that Weight-Tau $\in [-1, 1]$ whereas others are $\in [0, 1]$.

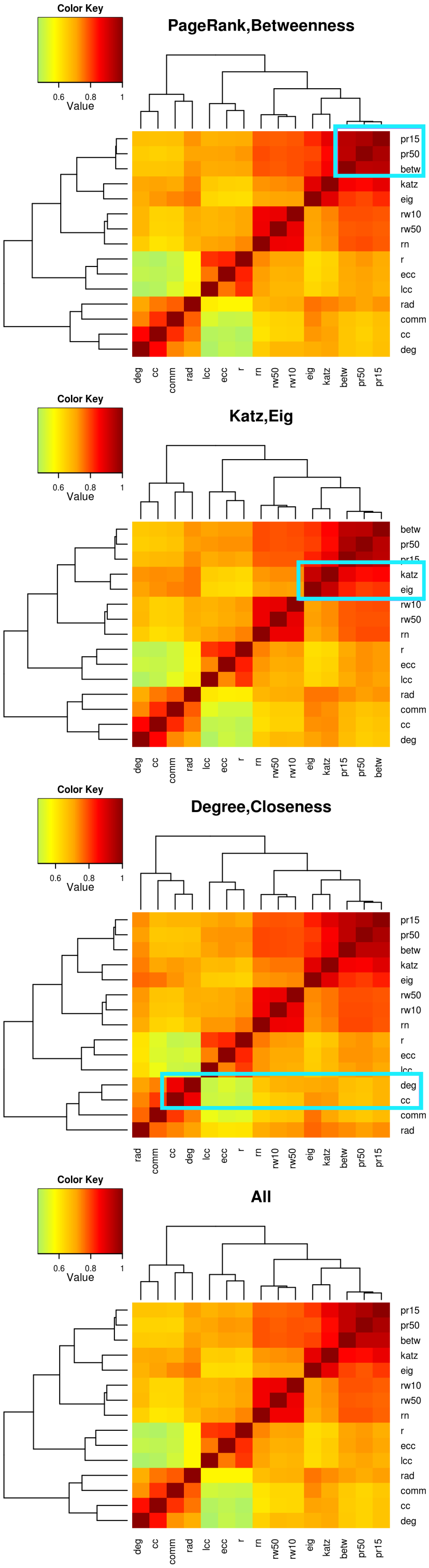


Figure 4: Node-based attack strategies clustered with Bi-Match. Heatmaps representing individual clusters from Table 2 are the average of heatmaps obtained from graphs where the cluster appears. (bottom) Average heatmap across all graphs.

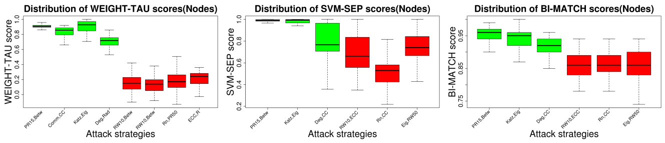
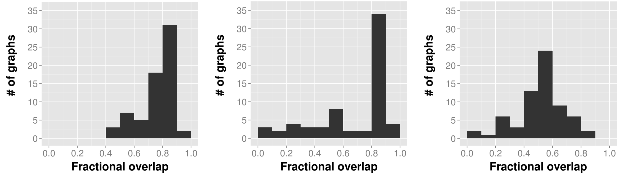


Figure 5: Box plots showing the distribution of similarity scores for all graphs for (green boxes) pairs of correlated strategies, and (red boxes) random pairs of strategies for comparison. (left) Weight-Tau, (middle) SVM-Sep, (right) Bi-Match.



(a) pr15 and betw (b) katz and eig (c) deg and cc

Figure 6: Distribution of the fractional overlap (1-1 correspondence) among top k nodes picked by correlated strategies.

Further, Figure 6 shows that the clustered strategies share significant overlap among their top k entities.

Table 1 showed that attack strategies have a wide variety of costs. Looking at our clustering results, we find attack strategies that fall in the same cluster but have differing computational costs. These include clusters {**PageRank**, Betweenness} and {**Degree**, Closeness}, where bold-faced strategies are cheaper. This result provides us with approximation opportunities; e.g., if a user is interested in picking nodes with highest betweenness (computationally expensive), s/he can employ PageRank as a proxy. This kind of approximation is tremendously helpful, especially for very large-scale real-world graphs.

Next we create consensus rankings in Rank-C and aim to identify a few (cheap) strategies close to the consensus. Table 3 lists the top 5 strategies with most similar rankings to the consensus on ten example graphs. We find that katz, eig, betw, pr15 appear in majority of the graphs, where e.g., pr15 can be used as a cheap proxy to the consensus.

Table 3: Top 5 node-based strategies closest to the Kemeny-Young consensus across 10 example graphs.

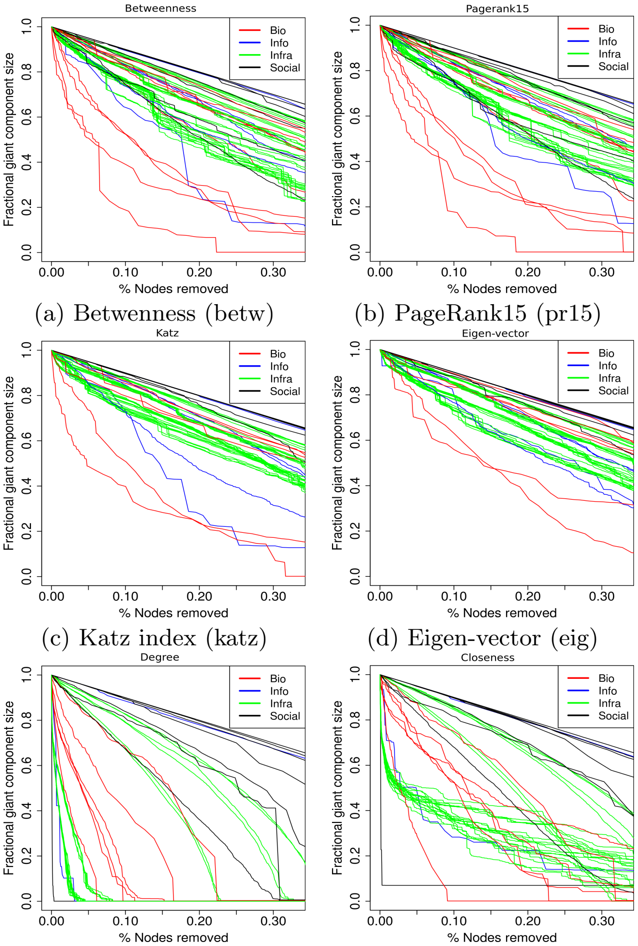
1	2	3	4	5	6	7	8	9	10
katz	katz	pr15	katz	betw	pr15	pr15	katz	katz	pr15
eig	pr15	katz	pr15	katz	katz	katz	pr15	eig	katz
pr15	pr50	pr50	pr50	pr15	pr50	comm	eig	pr15	pr50
betw	eig	betw	cc	pr50	betw	deg	pr50	pr50	eig
ecc	deg	eig	comm	ecc	eig	eig	betw	deg	betw

3.2 Correlation analysis by Response-C

Previously, we compared the nodes picked by each attack strategy directly (Weight-Tau) or by mapping them to a feature space (Bi-Match , SVM-Sep). Next, we actually simulate attacks on our graphs. That is, we remove nodes from each graph in order of the ranked list produced by each attack strategy. We use this attack-driven study as another way to validate our clustering results in §3.1.

Figure 7 shows how the robustness changes as more and more nodes are removed from the graphs. We notice that the graphs respond to correlated attack strategies similarly. For example, highly correlated (a) Betweenness and (b) PageRank cause similar disruption on a given graph.

The similarity between strategies based on how they rank the graphs by their resilience, as shown in Figure 8 (for both GCC fraction and λ_1), provides results in agreement with



(a) Betweenness (betw)

(b) PageRank15 (pr15)

(c) Katz index (katz)

(d) Eigen-vector (eig)

(e) Degree (deg)

(f) Closeness (cc)

Figure 7: Disruption dynamics of all graphs when attacked by various strategies. Each line corresponds to a graph (colored by type). Notice that the clustered, i.e., correlated strategies (i.e., a&b, c&d, e&f) cause similar disruption on graphs.

earlier clusters. The figure also shows the similarity to the Kemeny-Young consensus, where pr15 and betw are once again among the top 3 closest strategies to the consensus. As PageRank is simpler than Betweenness, pr15 can be used as a proxy for the consensus among these node-based strategies.

Overall, the findings using Response-C corroborate our results using Rank-C and Top$k$-C. This suggests that our findings are significant and not a byproduct of specific comparison methodology employed.

4. RELATED WORK

Manipulating Graph Robustness. Given a graph, manipulating its structure in order to improve its robustness, decrease its vulnerability, or to simulate attack scenarios has been a research area of wide interest. Several works have focused on increasing robustness by adding new edges [3, 6, 31, 37], or rewiring existing edges [3, 24, 30]. Others have developed algorithms to degrade robustness, such as the spectral radius (or λ_1), so as to decrease the vulnerability of a graph to propagation of viruses, diseases, rumors, etc. [31, 32, 34]. In addition, several works have developed heuristic algorithms to simulate attacks to real-world networks and studied their response and tolerance to targeted intentional attacks [2, 7, 8, 9, 23, 39]. Example attack strategies include removal of most connected (i.e., highest degree) nodes [2, 3], or highest betweenness centrality [15]. This work focuses on a large body of such heuristic strategies study their correlations and find out their similarities and distinctions.

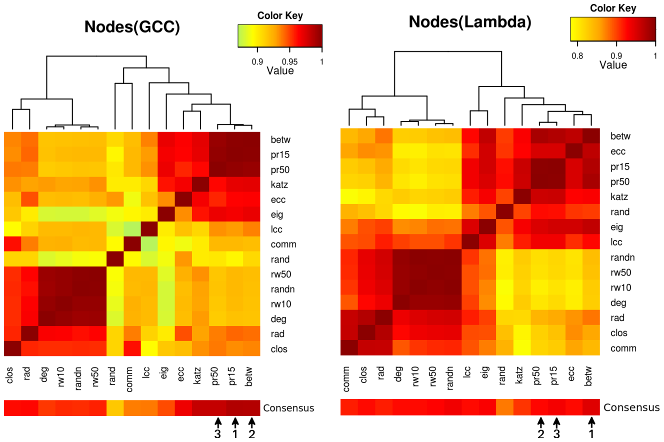


Figure 8: Similarity of strategies based on how they rank graphs by resilience in Response-C. Also shown is similarity of strategies to Kemeny-Young consensus, top 3 marked with arrows.

Correlation Analysis of Algorithms and Measures. Prior work that investigated correlation among node centrality measures [4, 11, 28, 33] also found high correlations and results implying clusters. Those, however, studied very small-scale networks that are only from one domain (often social), and considered fewer measures than our work. Recently, Vigna studied the ranking correlations of five centrality measures on larger graphs using their proposed measure [36].

The seminal work by Abrahao *et al.* proposed a framework for comparing different algorithms by their type of output [1]. Their work compared graph clustering methods and studied the type of clusters that they produce (w.r.t. e.g., density, size, cut-size, etc.). Similarly, Soundarajan *et al.* [29] analyzed correlations of graph similarity measures and proposed guidelines for selecting an appropriate measure.

5. CONCLUSION

Following the finding that scale-free networks are vulnerable to targeted attacks [2], a myriad of attack strategies has been developed to target nodes whose removal effectively degrade graph connectedness. However, how these strategies correlate with one another is not well understood. In this work, we present the first large-scale correlation analysis of attack strategies for graph robustness. We study 15 strategies on 68 real-world graphs, and utilize 3 different methodologies for correlation analysis. Our analyses show that (1) node importance measures employed by the strategies are well correlated, (2) several groups exhibit strong correlation across all the three methodologies, and (3) a few strategies are consistently close to a consensus among all of them. These findings improve our understanding of the strategies in the literature and present us with approximation opportunities, where computationally expensive measures can be approximated by comparable cheaper ones, and a few strategies can be used as a proxy to an overall consensus.

Acknowledgments Research is supported by the ARO Young Investigator Program Contract No. W911NF-14-1-0029, NSF CAREER 1452425, and NSF IIS 1017181. Any conclusions expressed in this material are of the authors' and do not necessarily reflect the views of ARO or NSF.

6. REFERENCES

[1] B. D. Abrahao, S. Soundarajan, J. E. Hopcroft, and R. Kleinberg. On the separability of structural classes of communities. In *KDD*, pages 624–632, 2012.

[2] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406(6794), 2000.

[3] A. Beygelzimer, G. Grinstein, R. Linsker, and I. Rish. Improving network robustness by edge modification. *Physica A: Stat. Mech. and its Appl.*, 357(3-4):593–612, 2005.

[4] J. M. Bolland. Sorting out centrality: An analysis of the performance of four centrality models in real and simulated networks. *Social Net.*, 10(3):233–253, 1988.

[5] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[6] H. Chan, L. Akoglu, and H. Tong. Make it or break it: Manipulating robustness in large networks. In *SDM*, 2014.

[7] R. Cohen, K. Erez, D. B. Avraham, and S. Havlin. Breakdown of the Internet under Intentional Attack. *Physical Review Letters*, 86(16):3682–3685, 2001.

[8] P. Crucitti, V. Latora, M. Marchiori, and A. Rapisarda. Error and attack tolerance of complex networks. *Physica A: Stat. Mech. and its Appl.*, (1-3):388–394, 2004.

[9] E. Estrada. Network robustness to targeted attacks. the interplay of expansibility and degree distribution. *Phys. J. B - Complex Systems*, 52(4):563–574, 2006.

[10] E. Estrada, N. Hatano, and M. Benzi. The physics of communicability in complex networks. *Physics Reports*, 514(3):89–119, 2012.

[11] K. Faust. Centrality in affiliation networks. *Social Networks*, 19(2):157–191, 1997.

[12] A. Fronczak, J. A. Hołyst, M. Jedynak, and J. Sienkiewicz. Higher order clustering coefficients in Barabási–Albert networks. *Physica A: Statistical Mechanics and its Applications*, 316(1):688–694, 2002.

[13] P. Hage and F. Harary. Eccentricity and centrality in networks. *Social networks*, 17(1):57–63, 1995.

[14] K. Henderson, B. Gallagher, L. Li, L. Akoglu, T. Eliassi-Rad, H. Tong, and C. Faloutsos. It's who you know: Graph mining using recursive structural features. In *KDD*, 2011.

[15] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han. Attack vulnerability of complex networks. *Phy. Rev. E*, 65(5), 2002.

[16] U. Kang, M. McGlohon, L. Akoglu, and C. Faloutsos. Patterns on the connected components of terabyte-scale graphs. In *ICDM*, pages 875–880. IEEE Computer Society, 2010.

[17] L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

[18] J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[19] M. G. Kendall. A new measure of rank correlation. *Biometrika*, pages 81–93, 1938.

[20] G. N. Lance and W. T. Williams. A general theory of classificatory sorting strategies ii. clustering systems. *The computer journal*, 10(3):271–277, 1967.

[21] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *KDD*, 2005.

[22] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Int. Math.*, 6(1):29–123, 2009.

[23] R.-H. Li, J. X. Yu, X. Huang, H. Cheng, and Z. Shang. Measuring robustness of complex networks under mvc attack. In *CIKM*, 2012.

[24] V. H. Louzada, F. Daolio, H. J. Herrmann, and M. Tomassini. Smart rewiring for network robustness. *Journal of Complex Networks*, 1(2):150–159, 2013.

[25] M. McGlohon, L. Akoglu, and C. Faloutsos. Weighted graphs and disconnected components: patterns and a generator. In *KDD*, pages 524–532. ACM, 2008.

[26] T. Opsahl, F. Agneessens, and J. Skvoretz. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Net.*, 32(3):245–251, 2010.

[27] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *Stanford InfoLab*, 1999.

[28] R. B. Rothenberg, J. J. Potterat, D. E. Woodhouse, W. W. Darrow, S. Q. Muth, and A. S. Klovdahl. Choosing a centrality measure: Epidemiologic correlates in the Colorado Springs study of social networks. *Social Networks*, 17(3-4):273–297, 1995.

[29] S. Soundarajan, T. Eliassi-Rad, and B. Gallagher. A guide to selecting a network similarity method. In *SDM*, 2014.

[30] A. Sydney, C. M. Scoglio, and D. Gruenbacher. Optimizing algebraic connectivity by edge rewiring. *Applied Mathematics and Computation*, 219(10):5465–5479, 2013.

[31] H. Tong, B. A. Prakash, T. Eliassi-Rad, M. Faloutsos, and C. Faloutsos. Gelling, and melting, large graphs by edge manipulation. In *CIKM*, pages 245–254, 2012.

[32] H. Tong, B. A. Prakash, C. E. Tsourakakis, T. Eliassi-Rad, C. Faloutsos, and D. H. Chau. On the vulnerability of large graphs. In *ICDM*, 2010.

[33] T. W. Valente, K. Coronges, C. Lakon, and E. Costenbader. How Correlated Are Network Centrality Measures? *Connections (Toronto, Ont.)*, 28(1):16–26, Jan. 2008.

[34] P. Van Mieghem, D. Stevanović, F. Kuipers, C. Li, R. Van De Bovenkamp, D. Liu, and H. Wang. Decreasing the spectral radius of a graph by link removals. *Phy. Rev. E*, 2011.

[35] V. Vapnik. *Statistical Learning Theory*. Wiley NY, 1998.

[36] S. Vigna. A weighted correlation index for rankings with ties. *CoRR*, abs/1404.3325, 2014.

[37] H. Wang and P. Van Mieghem. Algebraic connectivity optimization via link addition. In *Bionetics*, 2008.

[38] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[39] J. Wu, H. Z. Deng, Y. J. Tan, and D. Z. Zhu. Vulnerability of complex networks under intentional attack with incomplete information. *Journal of Physics A*, 40(11), 2007.