A Sample-Based Algorithm for Approximately Testing r-Robustness of a Digraph

Yuhao Yi, Yuan Wang, KTH

Xingkang He, Notre Dame

Stacy Patterson, RPI

Karl H. Johansson, KTH

CDC 2022

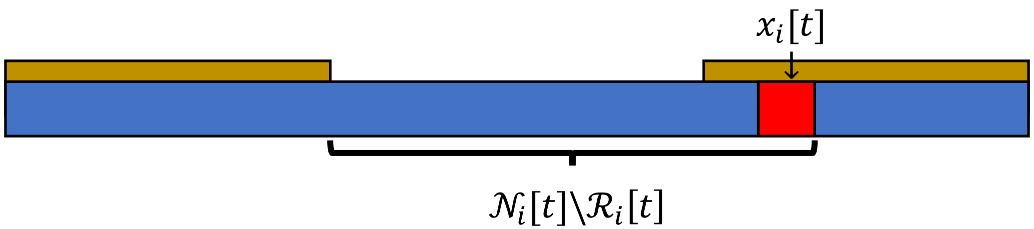
Cancún, Mexico

Background: Resilient Consensus

- Resilient consensus
  - Reach agreement in the presence of adversaries
- Types of adversary
  - Byzantine
  - Malicious
- F-local and F-total sets
  - F-local: each honest node has at most F adversarial neighbors
  - F-total: at most F adversarial nodes in the network
- Weighted-Mean-Subsequence-Reduced algorithm

W-MSR Algorithm

- At time step t, a node i obtains the values of its in-neighbors and itself
- It sorts the values and removes values of a node set $\mathcal{R}_i[t]$
  - $\mathcal{R}_i[t]$ is constructed from the highest and lowest F values in the list, and modified according to $x_i[t]$



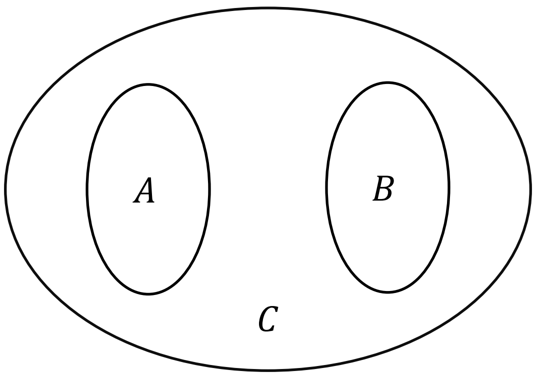
  - $x_i[t+1] = \sum_{j \in \mathcal{N}_i[t]\backslash\mathcal{R}_i[t]} w_{ij}\, x_j^i[t]$
- $(2F+1)$-robustness is sufficient for the W-MSR algorithm to tolerate Byzantine/Malicious F-local/total adversaries

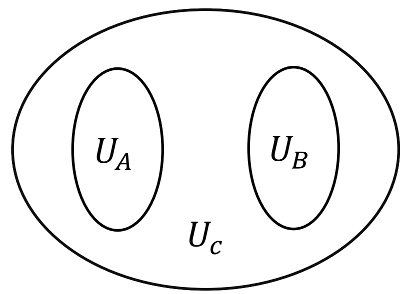
r-Robustness

- $\boldsymbol{r}$**-Reachability**: A set S is r-reachable if these exists a node $v \in S$, v has at least r in-neighbors from $V \backslash S$.
- $\boldsymbol{r}$**-Robustness**: A graph G is r-robust if for every pair of non-empty, disjoint sets $A, B \subseteq V$, at least one of A and B is r-reachable.
- The decision problem of r-robustness is coNP-complete
  - No polynomial time algorithm to solve the problem exactly unless P=NP
- A naïve algorithm enumerates all subset pairs
- We propose an additive approximation algorithm in dense digraphs.

Sampling Vertices

- Randomly sample a node set U; enumerate partitions of U





- For $v \in V \backslash U$, approximate numbers of neighbors in A, B, and C using neighbors in U_A, U_B, and U_c.
- For $v_1, v_2 \in V \backslash U$, the estimated numbers of neighbors in A, B, and C are NOT independent

Construct the “Counterexample”

- Suppose there is a partition that refutes r-robustness
- For a node $v \notin U$, there are 4 cases
- Case 1:
- $\frac{|\mathcal{N}_v \cap (U_A \cup U_C)|}{|U|} \geq \frac{r}{n} + \epsilon$ and $\frac{|\mathcal{N}_v \cap (U_B \cup U_C)|}{|U|} \geq \frac{r}{n} + \epsilon$,
- with high probability, $|\mathcal{N}_v \cap (A \cup C)| \geq r$, and $|\mathcal{N}_v \cap (B \cup C)| \geq r$
- v is not in either A or B, w.h.p.
- We assign v to C

Construct the "Counterexample" (cont'd)

- Case 2:
- $\frac{|\mathcal{N}_v \cap (U_A \cup U_C)|}{|U|} \geq \frac{r}{n} + \epsilon$, and $\frac{|\mathcal{N}_v \cap (U_B \cup U_C)|}{|U|} < \frac{r}{n} + \epsilon$,
- $|\mathcal{N}_v \cap (A \cup C)| \geq r$, $|\mathcal{N}_v \cap (B \cup C)| < r + 2\epsilon n$, w.h.p.
- Not in B w.h.p.
- Could be in A or C
- Never a bad idea to assign it to A because moving nodes from C to A
  - Never increases $|\mathcal{N}_v \cap (A \cup C)|$ or $|\mathcal{N}_v \cap (B \cup C)|$ for any v
- Assign v to A

Construct the "Counterexample" (cont'd)

- Case 3:
  - $\frac{|\mathcal{N}_v \cap (U_A \cup U_C)|}{|U|} < \frac{r}{n} + \epsilon$, and $\frac{|\mathcal{N}_v \cap (U_B \cup U_C)|}{|U|} \geq \frac{r}{n} + \epsilon$,
  - Assign v to B, for similar reasons
- Case 4 (without assuming minimum degree):
  - $\frac{|\mathcal{N}_v \cap (U_A \cup U_C)|}{|U|} < \frac{r}{n} + \epsilon$, and $\frac{|\mathcal{N}_v \cap (U_B \cup U_C)|}{|U|} < \frac{r}{n} + \epsilon$
  - w.h.p. , $|\mathcal{N}_v \cap (A \cup C)| < r + 2\epsilon n$, and $|\mathcal{N}_v \cap (B \cup C)| < r + 2\epsilon n$
  - $A = \{v\}, B = V \backslash \{v\}, C = \emptyset$ refutes $(2r + 4\epsilon n + 1)$-robustness

Algorithm Outline

- Sample a set U,
- For each partition $\pi(U) = (U_A, U_B, U_C)$,
  1. assign nodes in $V \backslash U$ to A, B, C using $\pi(U)$.
  2. Make a pass to reassign misclassified nodes due to large estimation errors
     - There are only a small number of such nodes
- Approximately constructed the "counterexample"

Result

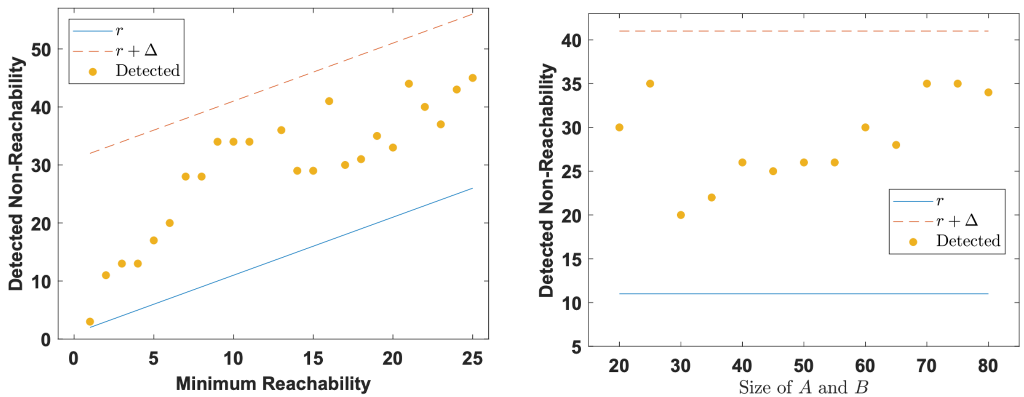
Theorem: Given a graph G, two numbers $r > 0$, $\epsilon \stackrel{\text{def}}{=} \Delta/n$ ($\epsilon \in (0,1]$), if the minimum in-degree is at least $2r + \Delta$, there is an algorithm which

- certifies r-robustness if G is $(r + \Delta)$-robust;
- refutes $(r + \Delta)$- robustness, with probability at least $(1 - \delta)$, if G is NOT r-robust;
- runs in $\exp\left(O\left(\frac{\log 1/(\epsilon\delta)}{\epsilon^2}\right)\right) \cdot m$ time, which is linear in m if ϵ is a given constant

- The algorithm can still be used as a heuristic even if the number of samples is not large enough.

Numerical Examples

- Synthesized networks with 200 nodes, permuted



- r-robust but not $(r+1)$-robust
- Most of the counterexamples are found in 30s on a laptop

Conclusion and Future Work

- Additive approximation algorithm for testing r-robustness

- Future work
  - Improve dependency on ϵ
  - Impact of graph regularity
  - Local Search Methods
  - More experiments and comparisons

Thank you!