*THIS DOCUMENT EXPLAINS THE* scale-free-protest\_superlinearScaling *REPOSITORY ON GITHUB AS WELL AS NEXT STEPS TO TAKE.*

The goal of this project is to understand if superlinear scaling can be explained via social network processes. Superlinear scaling refers to the fact that protests in large cities sometimes have more protesters per capita than smaller cities. For example, if City A has three times as many inhabitants as City B, it is likely to have more than three time as many protesters. This pattern does not arise in every protest wave, but it is consistent enough that it is worth investigating. The emergence of superlinear scaling will be investigated by modeling protest participation on networks as the networks vary in size (the number of nodes).

Before starting coding, please read (Steinert-Threlkeld and Steinert-Threlkeld 2021). It uses a series of simulations to investigate the emergence of the repression-dissent puzzle as a function of network structure. Though it does not focus on scaling, its code is useful as a starting point for building models for this paper. In particular, pay attention to the Holme-Kim models as those are the ones this project will use since they are most similar to real world networks.

Motivation/OLD\_Protest\_SuperlinearScaling.pdf summarizes the state of this project as of November 1, 2018. This paper documents superlinear scaling in actual protest data and shows initial modeling that failed to generate superlinear scaling. The models used for those results are earlier, less sophisticated versions of what was used in the PLOS ONE paper, so inability to generate scaling should not be overinterpreted. In addition to providing motivation, the key use of this paper is to show the analysis to perform for the simulation results. The x-axis will be network size, the y-axis is size of a simulation's protest, and the line of best fit is the resulting scaling relationship.

Scripts/simulations/simulations.py is what you will use to develop your code. In simulations.py, Holme-Kim models are the POWERLAW\_CLUSTER graph type. Each experiment generates a dataframe based on the simulation, and these dataframes are what you will use to investigate superlinear scaling. See Scripts/analysis/ for code that analyzes the simulation output, especially 01\_ProcessSimulationData\_v3.R.

For all of the following experiments, *let the clustering parameter of the Holme-Kim network vary by the same number of increments as in Steinert-Threlkeld and Steinert-Threlkeld 2021.* This parameter is m\_t, and the code calls it the scaling parameter.

Once you are ready to investigate superlinear scaling, see if protest scaling emerges simply by changing the size of the network. Exactly what the maximum size is will depend on your computer’s power, but hopefully it can be at least 10,000 nodes. The minimum should be 100, and to start there should be at least 10, but hopefully more, size steps between the minimum and maximum network sizes. For each network size, run 100 trials; if you find the simulations take more than two days. For now, use uniform thresholds and node removal repression. Call this investigation **experiment1.**

If uniform thresholds fail to generate superlinear scaling, replace them with narrower ones. That is, the model currently allows the thresholds to range from [0,1], but make the maximum smaller. Try .5 and see if superlinear scaling emerges; if not, try .25. Call these investigations **experiment2a** and **experiment2b** respectively.

If narrower uniform thresholds fail to generate superlinear scaling, replace them with normal thresholds. Simulations.py currently uses np.random.normal(0.25, 0.122) as the distribution. If that fails to generate superlinear scaling, try np.random.normal(0.125, 0.061). Call these investigations **experiment3a** and **experiment3b** respectively.

As you try these experiments, make sure to save the results as different files, i.e. do not overwrite your files.

For the first analysis of results, do not filter results based on m\_t. Once this analysis is done, repeat the analysis but using only the simulations where m\_t=.315; that value produces levels of local clustering most similar to observed social networks. If neither of these analyzes reveal superlinear scaling, repeat them but for every unique value of m\_t. The main result from this analysis will be a graph where the x-axis is m\_t and the y-axis is the slope of the fit line for the results for each value of m\_t.

# REFERENCES

Steinert-Threlkeld, Shane, and Zachary Steinert-Threlkeld. 2021. “How Social Networks Affect the Repression-Dissent Puzzle.” *PLoS ONE* 16(5 May): 1–23. http://dx.doi.org/10.1371/journal.pone.0250784.