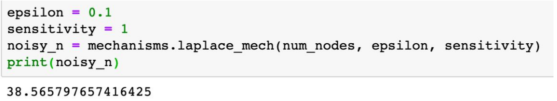
Releasing Differentially-Private Edge Counts of Networks

Sam Durst | Dave Landay

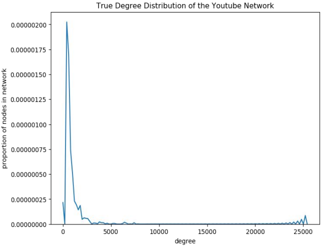
Paradigm Shift: Node-DP and Edge-DP

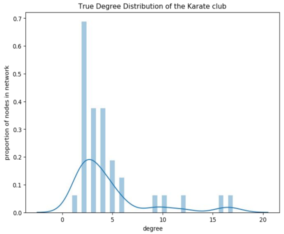
- **Node-DP:** Two graphs G and G' are neighbors if one can be obtained from the other by deleting a node and its adjacent edges. (i.e: differ by one node and sum total of edges connecting that node to other nodes in the network)
- **Edge-DP:** Two Graphs G and G' are neighbors if they differ by one edge



```
epsilon = 0.1
sensitivity = 1
noisy_n = mechanisms.laplace_mech(num_nodes, epsilon, sensitivity)
print(noisy_n)

38.565797657416425
```

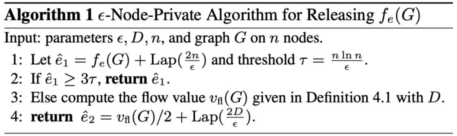




For Small Graphs, Global Sensitivity Won't Cut It

```
Noisy edge count that satisfies node-dp: 73.27273276031647
Noisy edge count that satisfies node-dp: 44.85308910552921
Noisy edge count that satisfies node-dp: 49.30224525236329
Noisy edge count that satisfies node-dp: 73.71352063225345
Noisy edge count that satisfies node-dp: 334.90051585381184
Noisy edge count that satisfies node-dp: -18.196147788006655
Noisy edge count that satisfies node-dp: 217.04731214107503
Noisy edge count that satisfies node-dp: -124.98265241723666
Noisy edge count that satisfies node-dp: 200.38078652266293
Noisy edge count that satisfies node-dp: -324.52905888166185
```

How do we achieve privacy with small graphs?



```
Algorithm 1 $\epsilon$-Node-Private Algorithm for Releasing $f_e(G)$
Input: parameters $\epsilon, D, n$, and graph $G$ on $n$ nodes.
  Let $\hat{e}_1 = f_e(G) + \mathrm{Lap}(\frac{2n}{\epsilon})$ and threshold $\tau = \frac{n \ln n}{\epsilon}$.
  If $\hat{e}_1 \geq 3\tau$, return $\hat{e}_1$.
  Else compute the flow value $v_{\mathrm{fl}}(G)$ given in Definition 4.1 with $D$.
  return $\hat{e}_2 = v_{\mathrm{fl}}(G)/2 + \mathrm{Lap}(\frac{2D}{\epsilon})$.
```

First we test to see if our noisy edge count, scaled by double the amount of nodes, is larger or smaller than the constant threshold we introduce, tao. If our noisy result is large enough, then we know that there are enough edges to ensure that we can release this result without compromising the privacy.

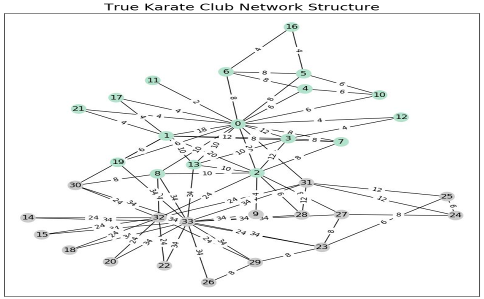
If the threshold is not met however, this could imply that releasing the query would possibly reveal information about the original data. Therefore, extra steps must be taken to ensure privacy.

Capacity

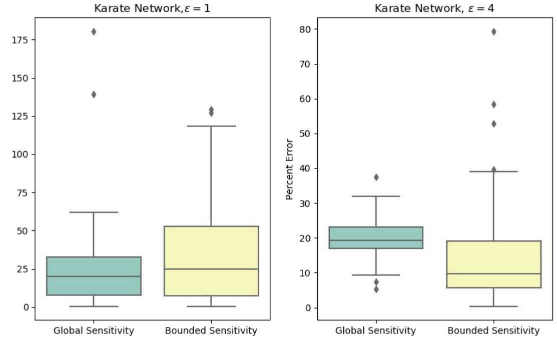
The D in Algorithm 1 represents capacity. The capacity is the upper bound for information to spread over the network. It must be chosen by an analyst, however there is no simple way for the analyst to choose a good capacity without trial and error. One alternative is to use the sparse vector technique with the above threshold method to determine a good capacity.

With the capacity chosen, the next step in the algorithm is to compute the maximum flow value. Flow is a measure of node sensitivity; i.e if we remove a node how much damage can it do? We can then return laplace noise over the flow scaled to two times the capacity because by adding or removing one node from a graph, the total number of edges can change by at most twice the chosen capacity.

This result is then returned to represent the node differentially private edge count.



Bounded Sensitivity Vs Global Sensitivity (D=15)



Future Work

- Sparse Vector Technique to pick a good capacity D