Fraud Detection by Dense Subgraph Detection

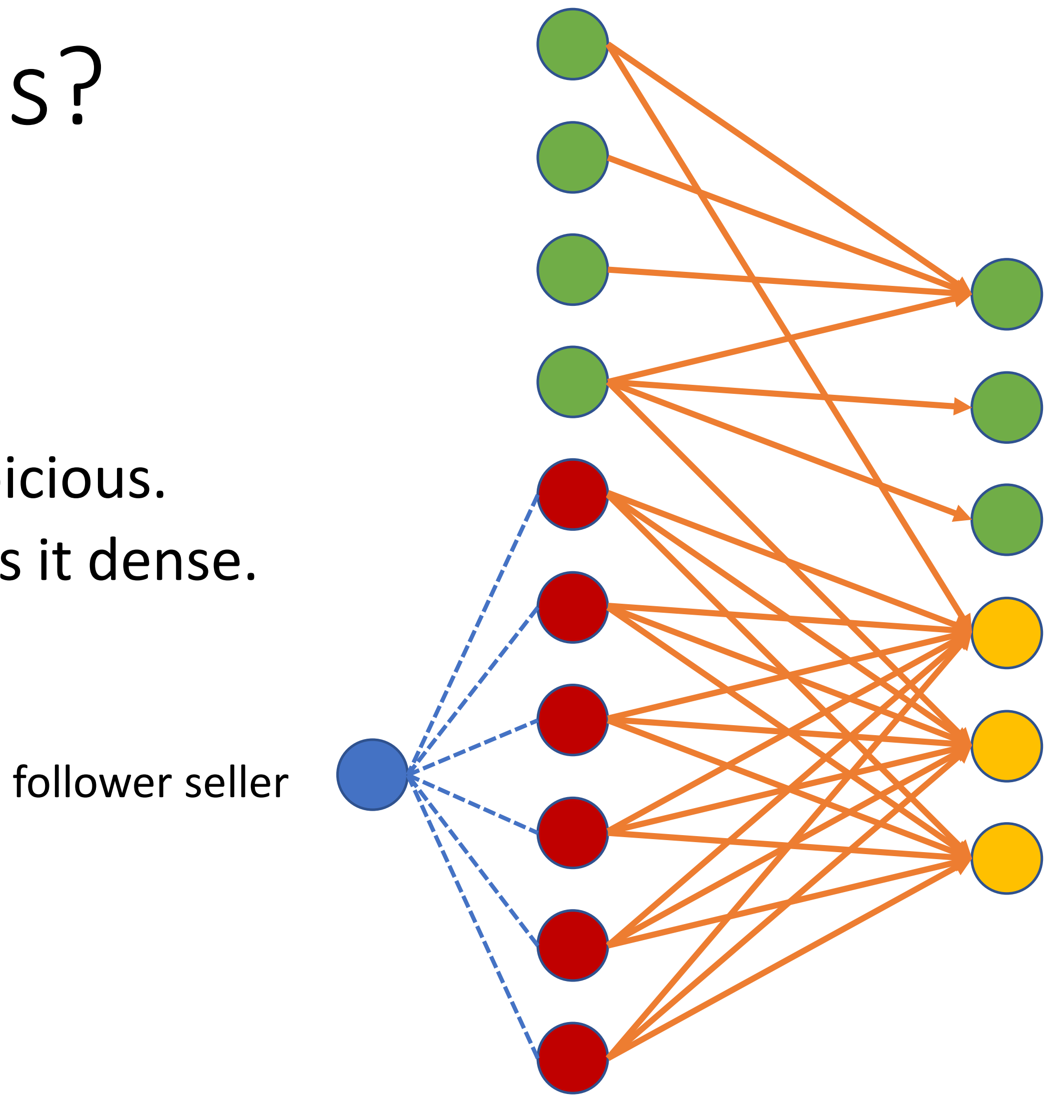
Tong Zhao

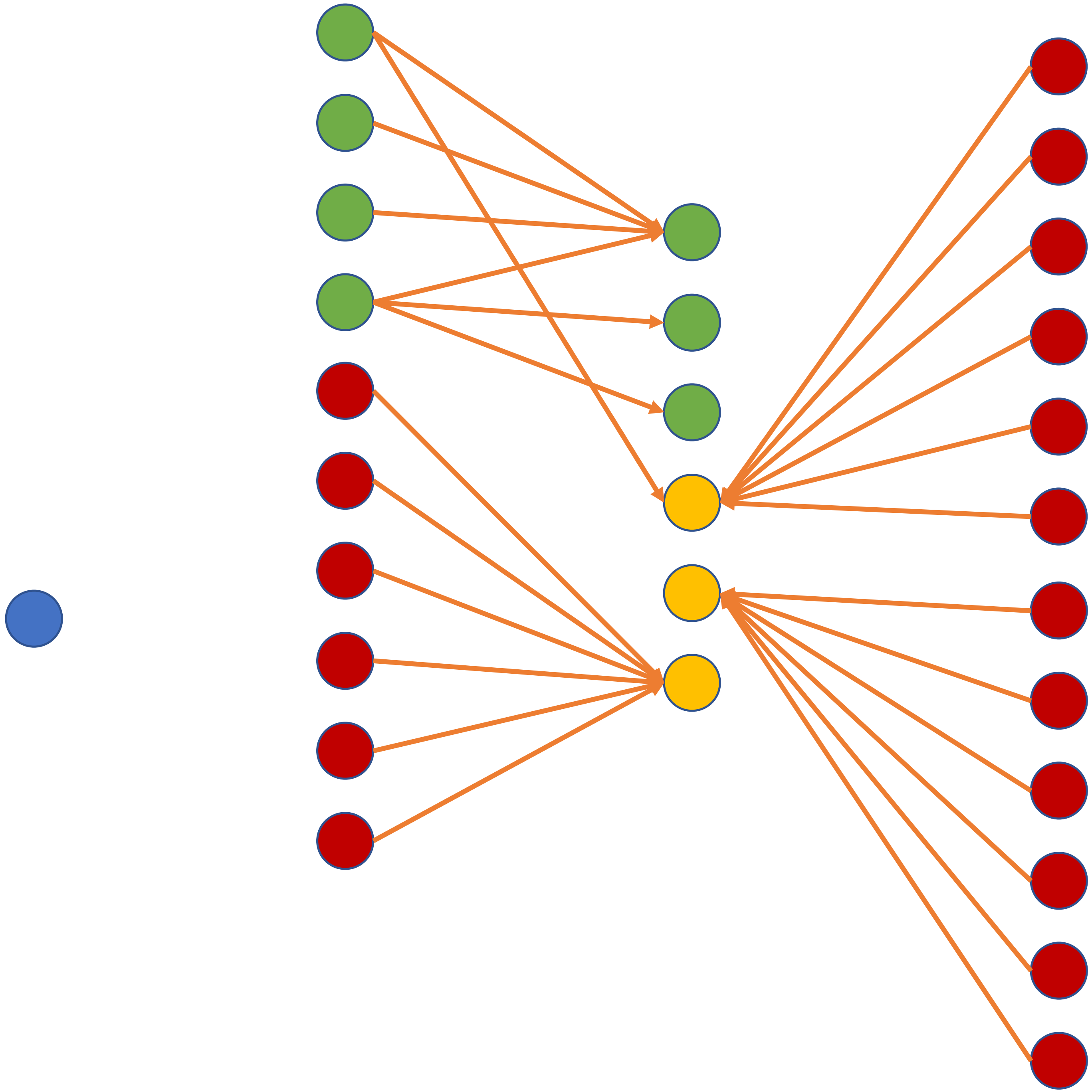
Why dense subgraphs?

- Graph-based fraud detection
  - Unsupervised learning.
  - Unexpecting high density is suspicious.

Why dense subgraphs?

- Graph-based fraud detection
  - Unsupervised learning.
  - Unexpecting high density is suspicious.
  - Fraudsters’ avoiding effort makes it dense.





Hardworking
follower seller

Dense Subgraph Detection

- Given a graph $G = (V, E)$ with vertices V and edges $E \subseteq V \times V$.
- Find a subgraph S such that $d(S)$ is maximized.
- Edge density (average degree): $d(S) = \frac{|E(S)|}{|S|}$
  - The larger, the better.
  - The denser, the better.

Charikar’s greedy algorithm (2000) [1]

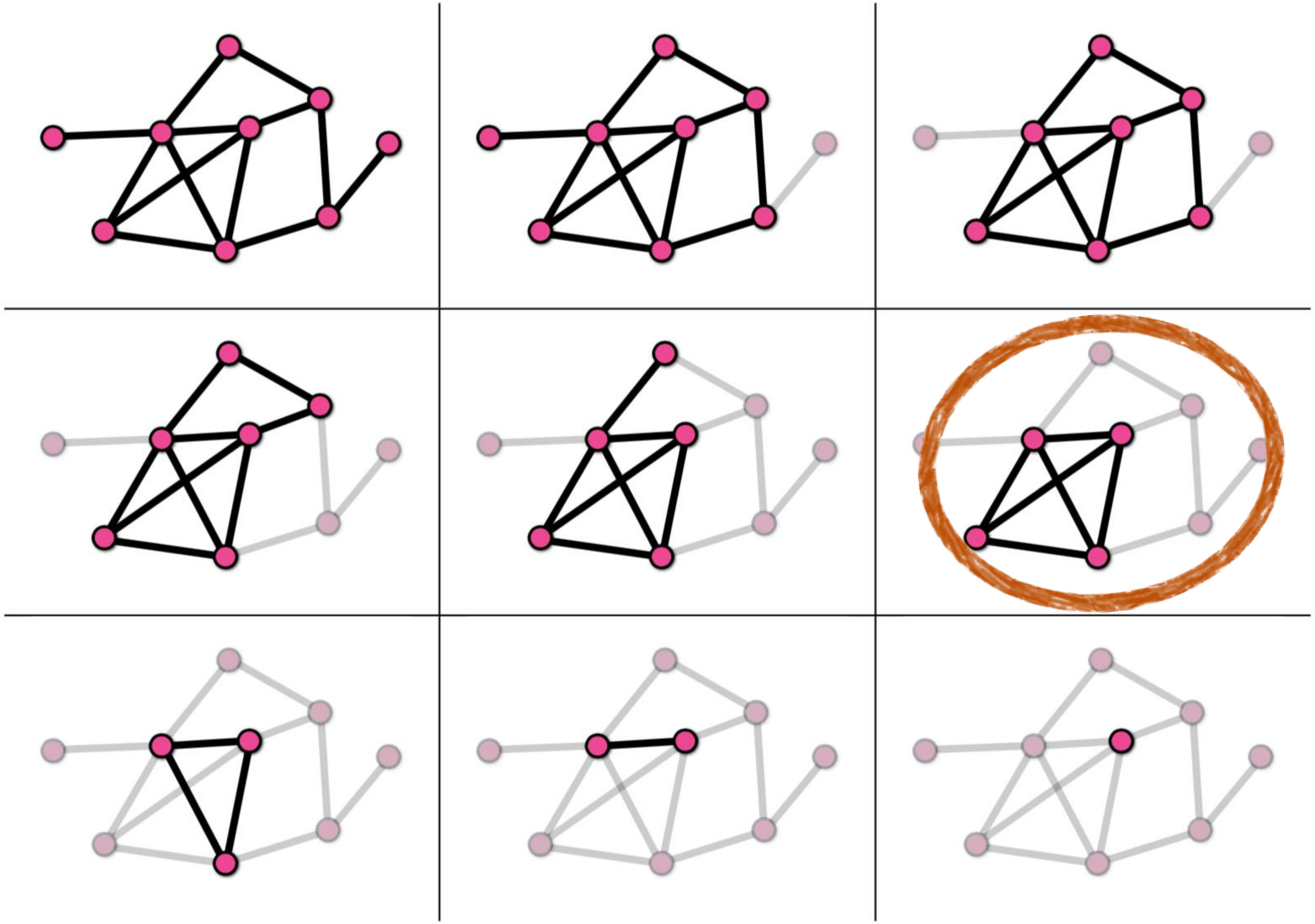


Figure from [3].

Fraudar [2] (Based on Charikar’s algorithm)

Require: Bipartite $G = (\mathcal{U} \cup \mathcal{W}, \mathcal{E})$; density metric g of the form in (1)

procedure FRAUDAR (G, g)
  Construct priority tree T from $\mathcal{U} \cup \mathcal{W}$
  $\mathcal{X}_0 \leftarrow \mathcal{U} \cup \mathcal{W}$
  for $t = 1, \ldots, (m+n)$ **do**
    $i^* \leftarrow \arg\max_{i \in \mathcal{X}_i} g(\mathcal{X}_i \setminus \{i\})$
    Update priorities in T for all neighbors of i^*
    $\mathcal{X}_t \leftarrow \mathcal{X}_{t-1} \setminus \{i^*\}$
  end for
  return $\arg\max_{\mathcal{X}_i \in \{\mathcal{X}_0, \ldots, \mathcal{X}_{m+n}\}} g(\mathcal{X}_i)$
end procedure

Total runtime: $O((|V| + |E|)\log|V|)$

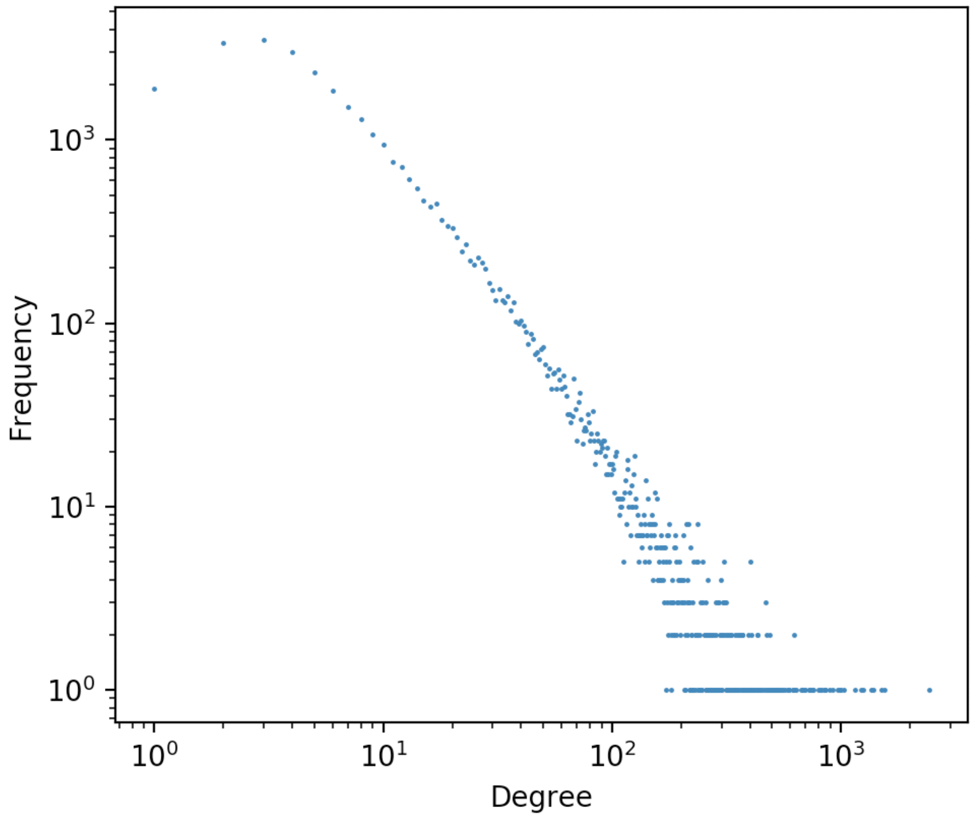
[2] c

Implementation

- Fraudar's source code.
- Written in Python.
- About 300 lines.
- Graphs stored in sparse matrix by SciPy.

Dataset

- Graphs generated by the provided graph generator. [4]
  - Fixed average degree as 20.
  - Changed # of vertices.
- Twitter dataset with 41.7 million users and 1.47 billion follows.
  - Failed.



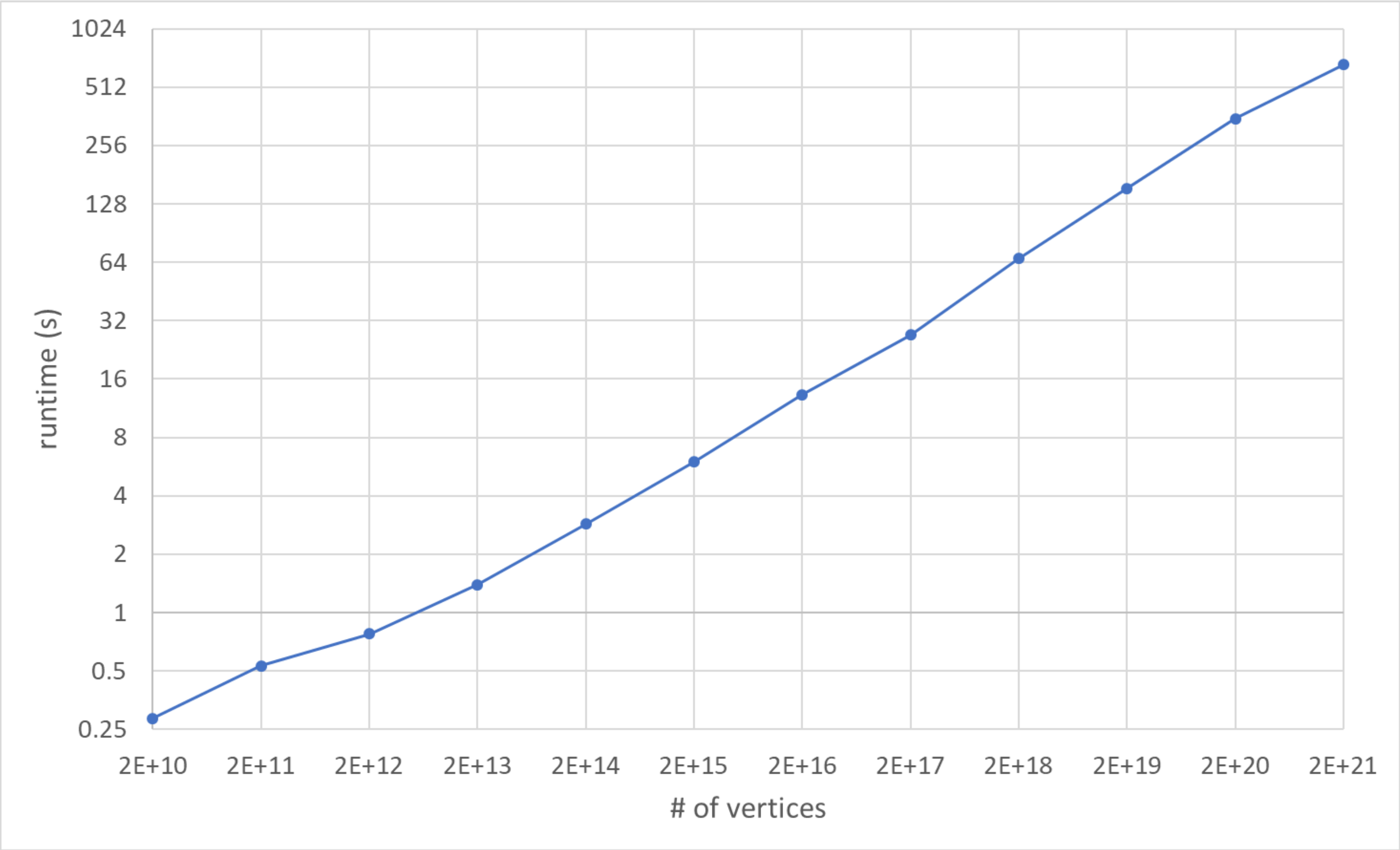
Performance

- Density of the result is theoretically guaranteed.
  - Charikar's algorithm is a provable 2-approximation algorithm.

$$d(S') \geq \frac{1}{2}\, d(S_{opt})$$

  - S' denotes the result subgraph by Charikar's algorithm.
  - S_{opt} denotes the optimal solution.

Performance



Future plan

- Apply Charikar’s algorithm on larger graphs.
- Dense subgraph detection for dynamic graphs.

References

- [1] Charikar, Moses. "Greedy approximation algorithms for finding dense components in a graph." *International Workshop on Approximation Algorithms for Combinatorial Optimization*. Springer, Berlin, Heidelberg, 2000.
- [2] Hooi, Bryan, et al. "Fraudar: Bounding graph fraud in the face of camouflage." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.
- [3] Gionis, Aristides, and Charalampos E. Tsourakakis. "Dense subgraph discovery: Kdd 2015 tutorial." *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015.
- [4] https://github.com/cooperative-computing-lab/graph-benchmark