

Summary of “Demystifying Graph Sparsification Algorithms in Graph Properties Preservation” by Y. Chen et al.

Summary by Vasilis Papastergios (ID: 3651)

Big Data Algorithms
B.Sc. School of Informatics, AUTH
Spring Semester 2023-2024

Table of contents

01

Introduction

04

Experimental Study

02

Graph Sparsification

05

**Results & Take-home
message**

03

Graph Metrics




01

Introduction




Introduction

In this presentation we provide a summary of the research paper “Demystifying Graph Sparsification Algorithms in Graph Properties Preservation” authored by Y. Chen et al., 2023. The paper is accepted to be presented in the Proceedings of 50th International Conference on Very Large Databases (VLDB 2024). The presentation is held for the literature assignment in the course of Big Data Algorithms, Spring Semester 2023-2024.





Objectives

- Provide a concise summary of the paper.
 - Explore several classical and SoA sparsification algorithms.
 - Present and explain the paper results.
 - Elaborate on the appropriateness of sparsification algorithms, depending on the downstream task.
- 



02

Graph Sparsification

Graph Sparsification

Graph sparsification is a technique that approximates an arbitrary graph with a sparse graph. The sparse graph is, essentially, a subset of the initial graph edges and/or vertices. The most common case in sparsification algorithms is that the produced sparse graph consists of all the vertices and only a subset of edges of the original graph.



Why Graph Sparsification ?

The 3 V's of Big Data

Volume

A real-world graph can easily get out of hand in terms of the memory required to store and process it efficiently.

Velocity

Real-world applications usually operate in an online mode, i.e. the vertices and/or edges dynamically arrive as a stream.

Variety

How to do Graph Sparsification ?

Random

randomly samples a number of edges to preserve, based on the given prune rate

K-Neighbors

selects k edges for each vertex or all edges if the vertex degree is less than k .

Rank Degree

starts with random vertices (seeds) and preserves edges to the neighbors with higher degree. The newly introduced vertices are the new seeds. The process is repeated until prune rate is achieved.

Local Degree

similar to Rank Degree but deterministic. For each vertex selects edges to the $\deg(u)^a$ top-ranked neighbors, where $a \in [0, 1]$ is a hyper-parameter that controls the prune rate.

Spanning Forest

constructs a spanning forest of the given graph. The algorithm is randomized and does not provide control over the prune rate.

t-Spanner

constructs a spanning tree in which all pairwise distances of vertices are at most t times larger than the original.

How to do Graph Sparsification ?

Forest Fire

constructs a graph by adding one vertex at a time and forming edges to certain subsets of the existing vertices. It also simulates the “burning” of some edges under some specified probability.

Local Similarity

similar to L-spar, but using a normalized, ranked similarity measure.

G-spar

sorts all edges based on the Jaccard similarity of the incident vertices (the higher, the better). Preserves top-k edges, where k is defined based on the desired prune rate.

SCAN Similarity

similar to G-spar, but using structural similarity (SCANSimilarity) instead of Jaccard.

L-spar

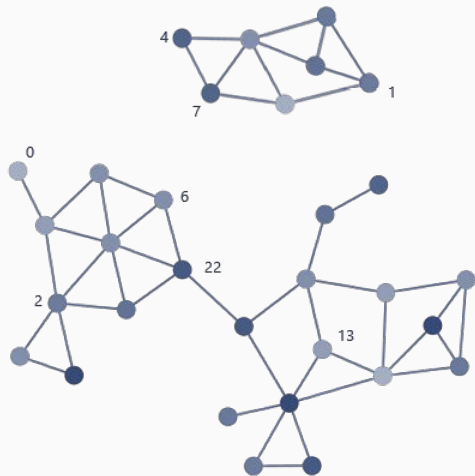
is the local variation of G-spar, where d^c edges are kept from each vertex locally, based on their Jaccard similarity score.

Effective Resistance

calculates the effective resistance for all edges. Afterwards, edges are selected with a probability proportional to their effective resistance.

03

Graph metrics



Graph Metrics

Basic

- Degree Distribution
- Laplacian Quadratic Form

Distance

- All pairs shortest path
- Diameter
- Vertex Eccentricity

Centrality

- Betweenness
- Closeness
- Eigenvector
- Katz

Clustering

- # of communities
- Local Clustering Coefficient
- Global Clustering Coefficient
- Clustering F1 score


Application-level

- PageRank
- Min cut / Max flow
- Graph Neural Networks (GNNs)



Sparsification and Metrics

An **ideal** sparsification algorithm needs to achieve a high prune rate while keeping the behavior of the downstream task as close to that of the original full graph. However, despite the large number of sparsification algorithms and graph metrics, there was **not much research work** conducted focusing on the connection between the use of specific sparsification algorithms and the presented graph metrics.





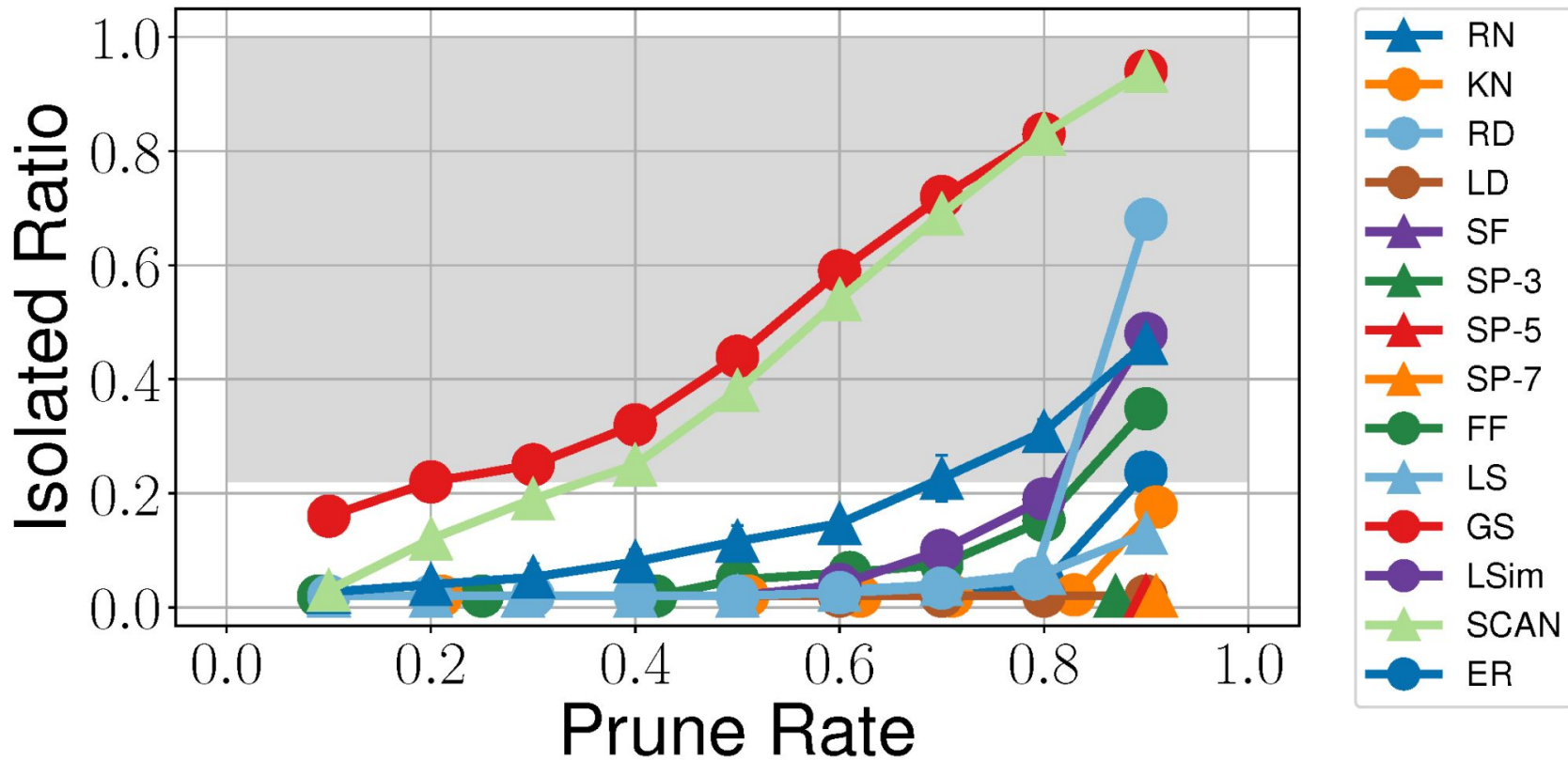
04

Experimental study

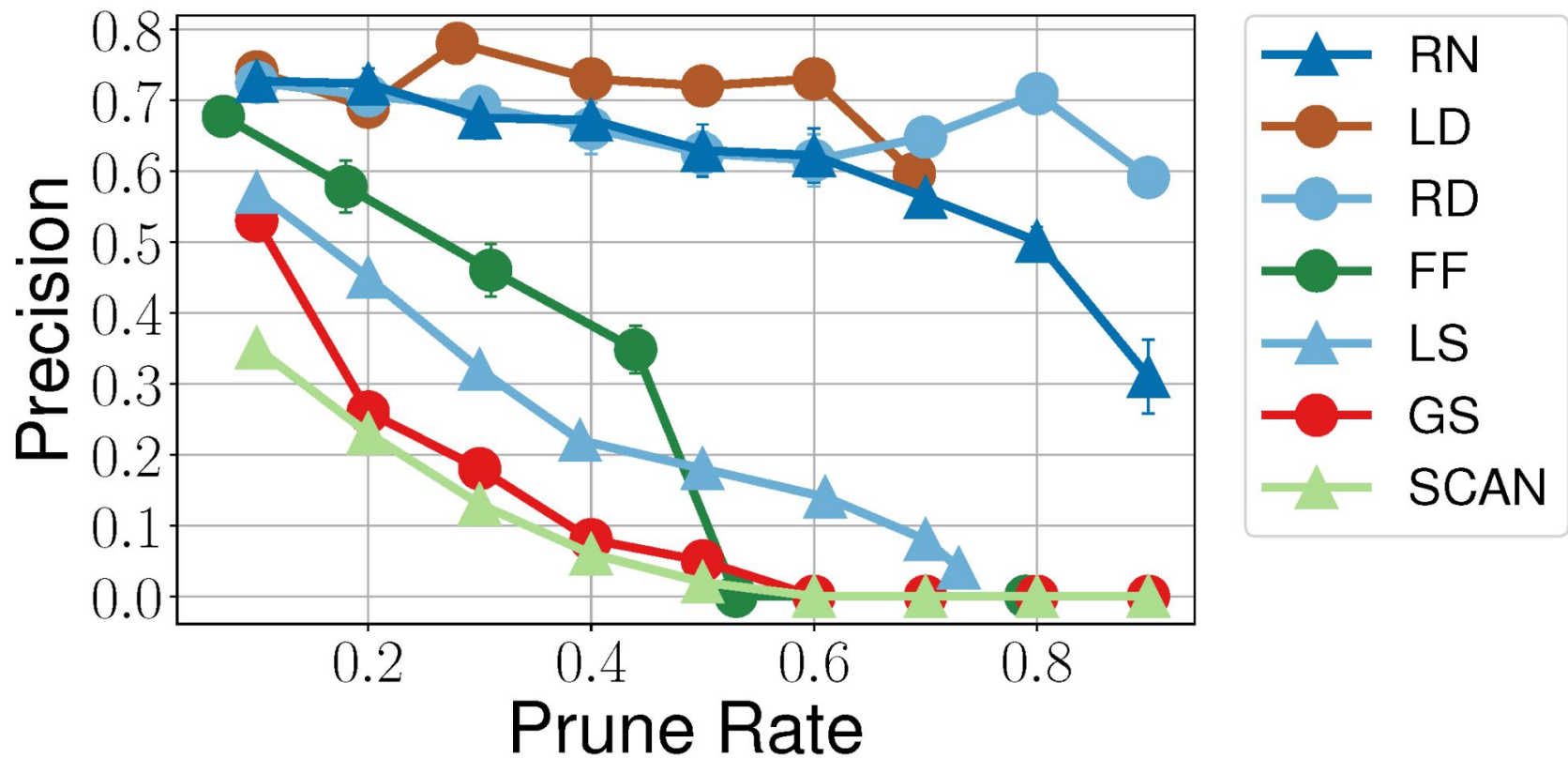
Demystifying Sparsifiers



Results (a glimpse)



Results (a glimpse)



Results


(final insights)

- **Random:** preserves relative (distribution-based or ranking-based) properties, for example, degree distribution, and top centrality rankings. It struggles to preserve absolute (valued-based) properties, for example, number of communities, clustering coefficient, and min-cut/max-flow.
- **K-Neighbors, Spanning Forest, t-Spanners:** preserves graph connectivity; keeps pair unreachable ratio and vertex isolated ratio low.
- **Rank Degree, Local Degree:** preserves graph connectivity and edges to high-degree vertices (hub vertices). Perform well on distance metrics (APSP, eccentricity, diameter) and centrality metrics.
- **Forest Fire:** Empirically it does not excel at any metrics evaluated.
- **G-Spar, SCAN:** Empirically perform well in preserving ClusterGCN accuracy.
- **L-Spar, Local Similarity:** preserves the edge to similar vertices, thus preserves clustering similarity.
- **ER:** preserves the spectral properties of the graph, specifically the quadratic form of the graph Laplacian. It perform well in preserving min-cut/max-flow results.



Take-home message

The authors' findings indicate that there is no single sparsification algorithm that excels in preserving all graph properties. The authors highlight the importance of **selecting** appropriate sparsification algorithms **based on the downstream task**.





Thank you!

Do you have any questions?



papastva@csd.auth.gr



github.com/Bilpapster



linkedin.com/in/bilpapster