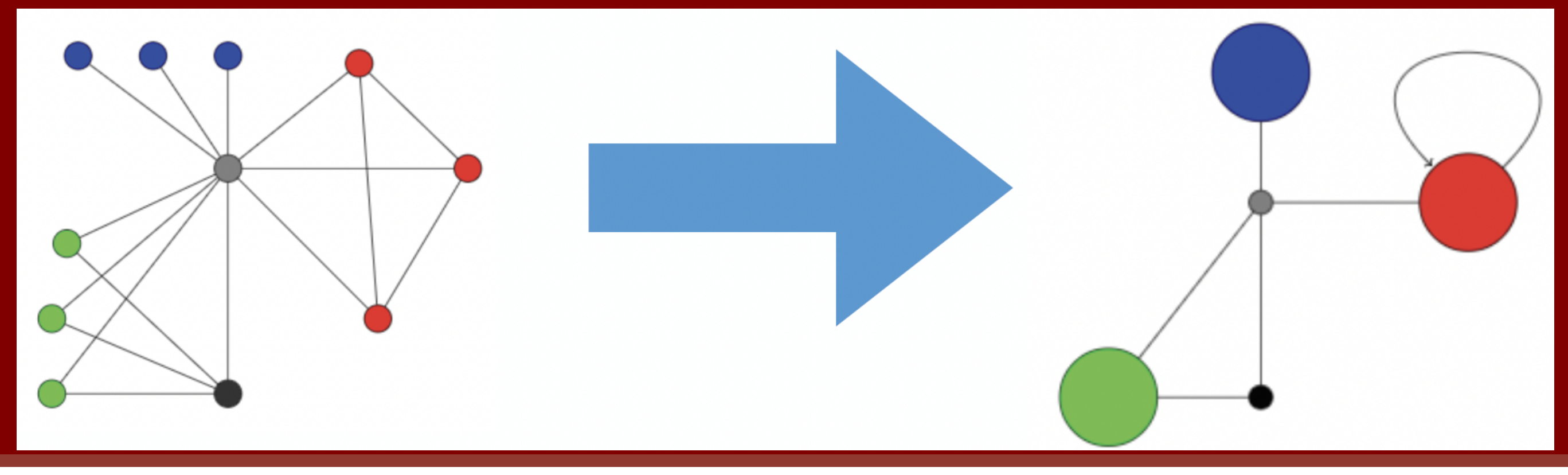


Scalable Algorithm for Graph Summarization

Hajiabadi, Mahdi   Thomo, Alex   Srinivasan, Venkatesh

University of Victoria

Background

Why graphs are popular?

Graphs are the most natural representation of real world data as set of nodes and set of edges:

- Protein-Protein interactions
- Social networks, Web graphs, Collaboration networks
- Transportation networks

Challenges:

Graphs are increasing exponentially:

- 3.5 billion web pages connected by 129 billion hyperlinks
- Online social networks with 300 billion connections
- Storing, mining and visualization are the main challenges.

Graph Summarization is used for:

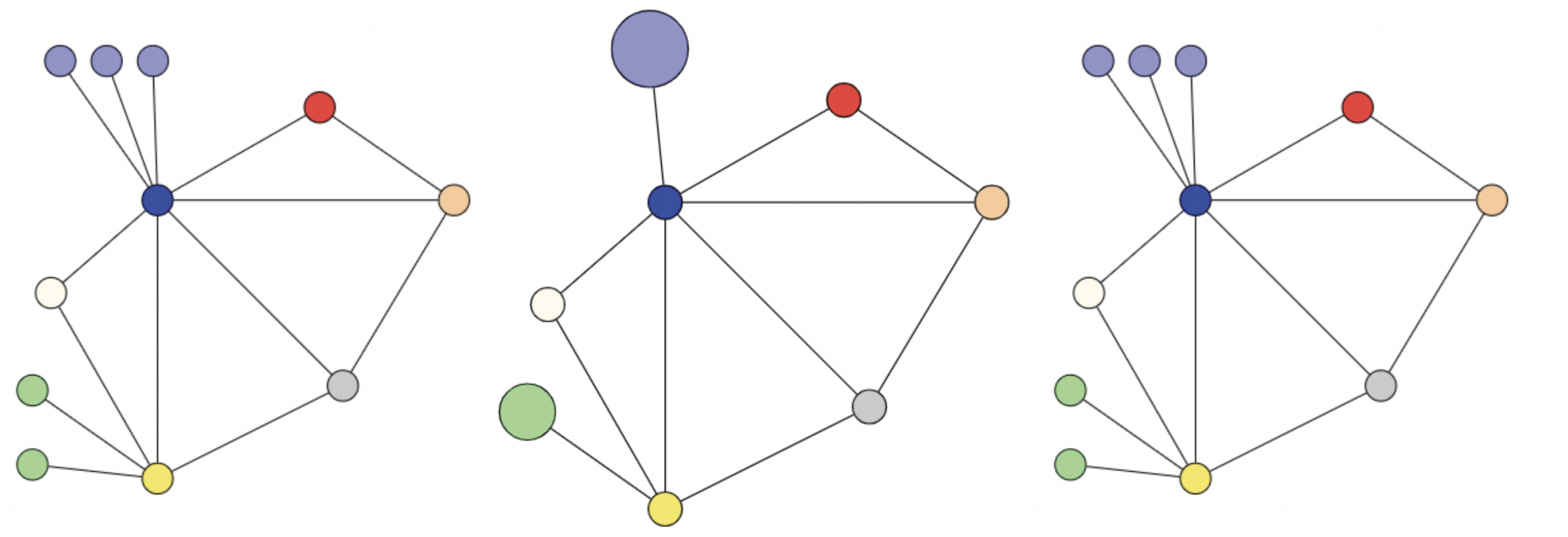
- Better Visualization
- Effective query answering
- Decreasing the footprint of graph

Definition of Graph Summarization:

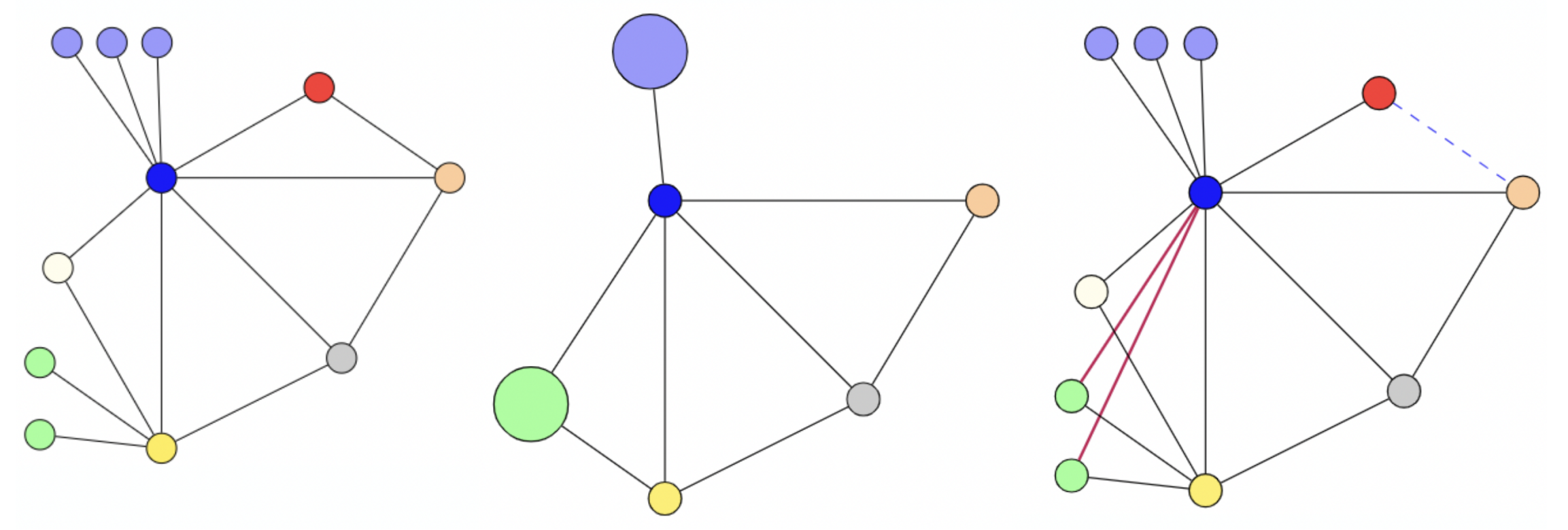
Find compact representation of the original graph called summary.

Graph Summarization can be either:

- Lossless: Summarizing graph without loosing any information:



- Lossy: Loosing some information from the original graph in order to gain more compression.



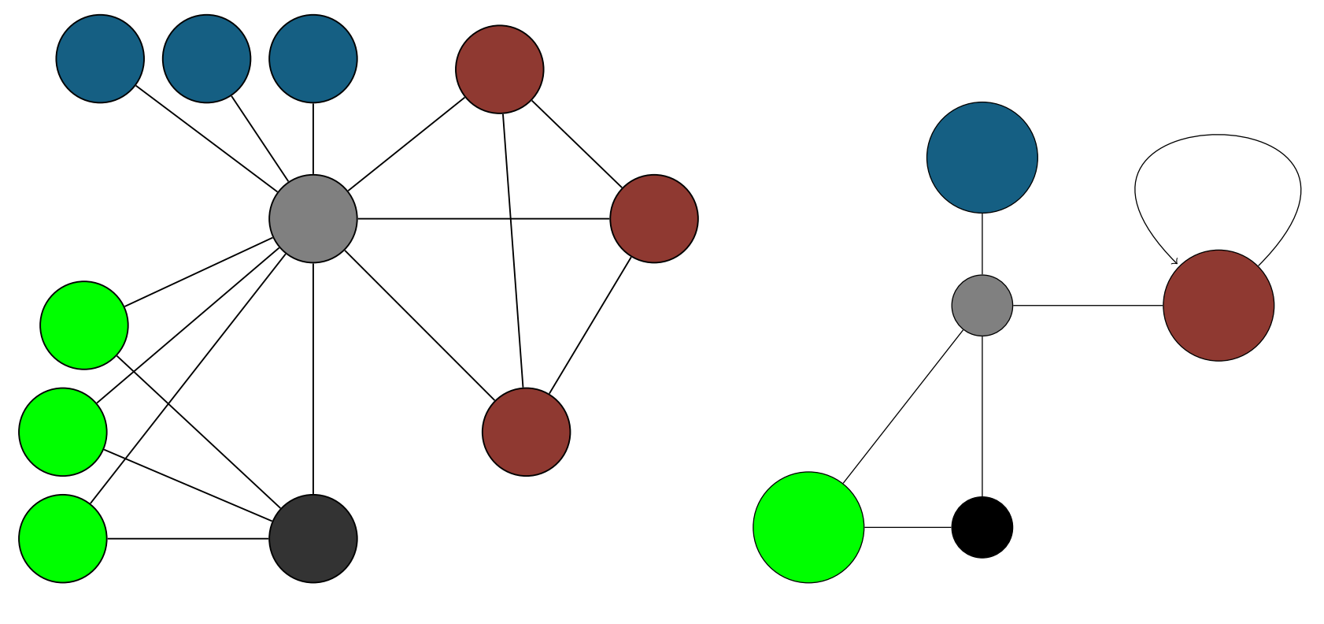
Contribution:

- We present a super fast lossless algorithm, G-SCIS
- Using G-SCIS summary for query answering

Intuition

In a lossless summary each node can be either

- In a supernode with size 1 (grey and black nodes in the following Figure)
- Inside a supernode representing a clique (Red nodes)
- Inside a supernode representing an independent set (Green nodes)



(a) Original Graph   (b) Summary Graph

G-SCIS:

A Naive Approach:

The task is finding a set of nodes which share above features, and merge them together in the same supernode

A naive approach is comparing the neighbors of each node with the neighbors of all other nodes. This approach is not scalable on large graphs ($O(VE)$). (It takes around one and a half year for a large graph with 39 million nodes and 1.5 billion edges).

Proposed Method (G-SCIS):

Alternatively we use the hash function which is highly applicable in data clustering and cryptography. Hashing is a probabilistic algorithm which does not have any **False Negative** errors but it may have **False Positive** error.

Steps:

- Using a hash fucntion to bucketize all the nodes in the graph according to its neighborhood
- Filter buckets in order to make them **false positive free**
- Now each filtered bucket is a supernode
- Draw superedge between supernodes

Dataset Description

We used seven web and social graphs from (`http://law.di.unimi.it/datasets.php`) varying from moderate size to very large. The following Table describes the data in terms of number of nodes, number of edges, and abbreviation.

Graph	Abbr	Nodes	Edges
cnr-2000	CN	325,557	5,565,380
hollywood-2009	H1	1,139,905	113,891,327
hollywood-2011	H2	2,180,759	228,985,632
indochina-2004	IC	7,414,866	304,472,122
uk-2002	U1	18,520,486	529,444,615
arabic-2005	AR	22,744,080	1,116,651,935
uk-2005	U2	39,459,925	1,581,073,454

Table 1. Summary of datasets

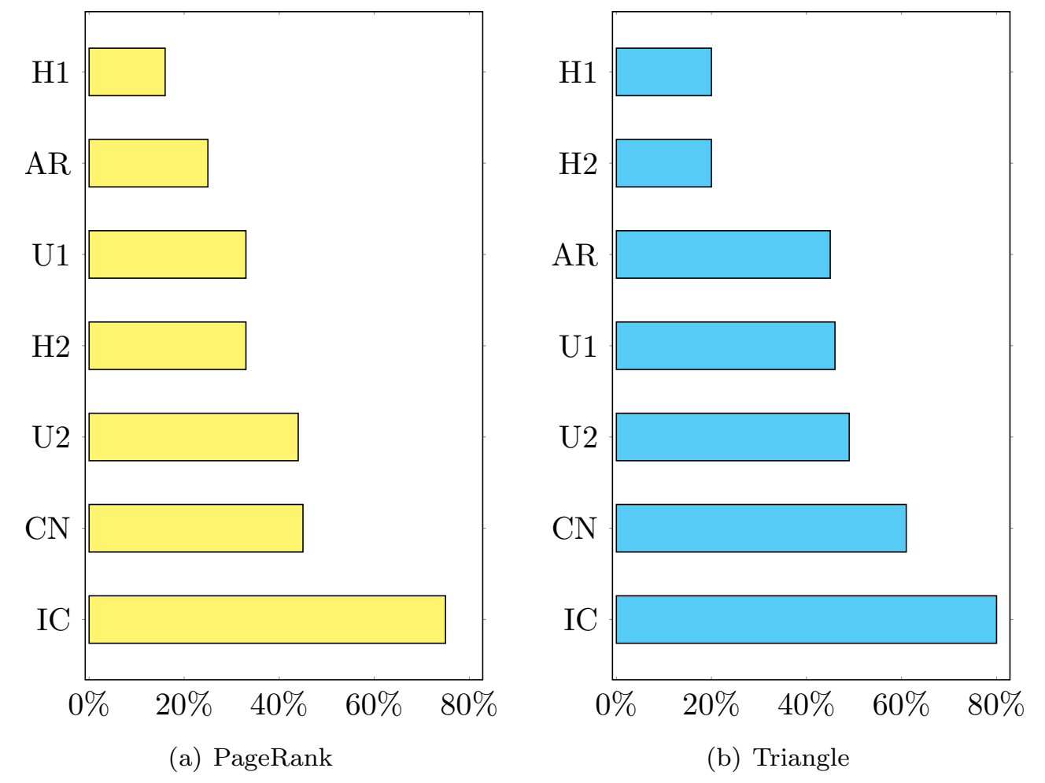
Experiments

RN value and running time

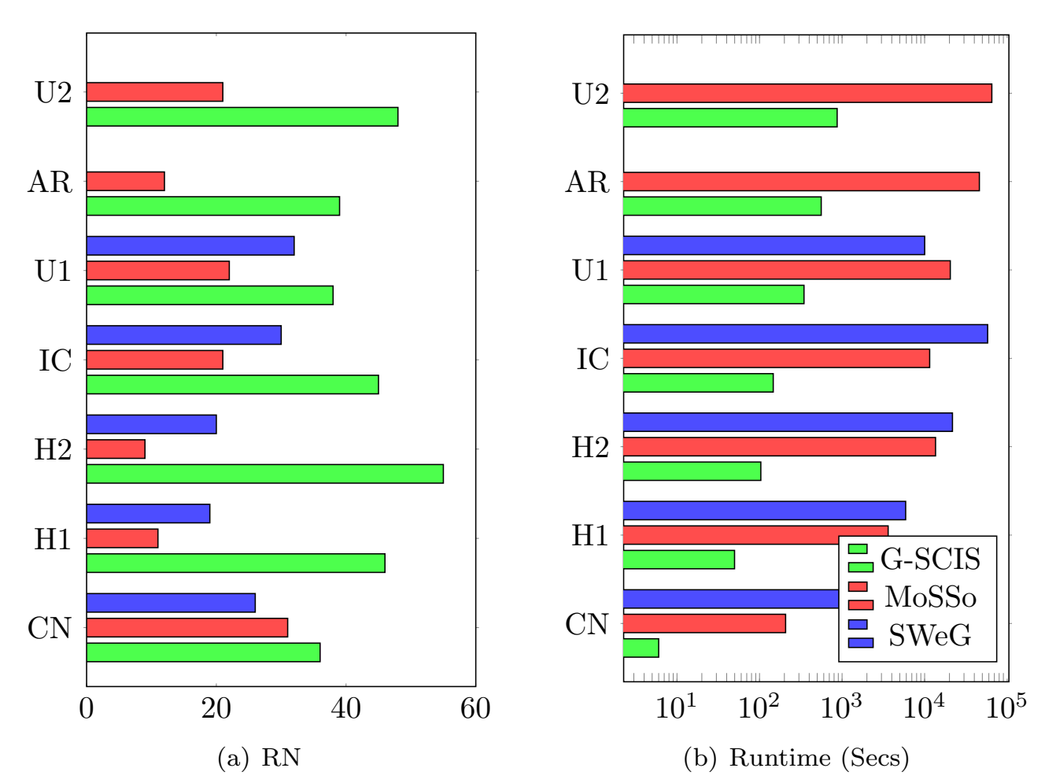
- The proposed method, G-SCIS (Graph Summarization based on Clique and Independent Set), is compared with MoSSo [1] and SWeG [2] in terms of compression (RN) and running time.
- Figure 1 shows the comparisons in terms of RN and running time in log-scale
- G-SCIS is up to 1000 time faster and it achieves 2.5x more compression compared with others
- It takes just 15-20 minutes to summarize a huge graph with 39 million nodes and 1.5 billion edges (U2) while the other one (MoSSO) takes a day to get the job done.

Query answering using summary graph

- We use G-SCIS summary graph as-is in order to answer
- We compare the running time of running query on the original graph vs running time of G-SCIS + running query on G-SCIS graph.
- We chose two different queries (PageRank, Triangle counting).
- Figure shows the relative improvement of using G-SCIS for answering queries.
- It is up to 5x faster if we use G-SCIS to answer queries



(c)



(d)

Conclusion

- We presented a fast algorithm which is up to 1000 time faster and 2x more compression
- We showed using the summary graph can speed up the query answering

This was a part of our recent publication in Knowledge Discovery and Data Mining Conference (SIGKDD 2021). You can have access the paper here

Future Work

- Graphs are dynamic and huge in nature and we have to summarize them

References

[1] Jihoon Ko, Yunbum Kook, and Kijung Shin.
Incremental lossless graph summarization.
In *KDD*, 2020.

[2] Kijung Shin, Amol Ghoting, Myunghwan Kim, and Hema Raghavan.
Sweg: Lossless and lossy summarization of web-scale graphs.
In *WWW*, 2019.