Reproducibility Report for ACM SIGMOD 2020 Paper: “Efficient Algorithms for Densest Subgraph Discovery on Large Directed Graphs”

ANDREAS KIPF, Massachusetts Institute of Technology, USA

The authors provided all code, datasets, and commands necessary to reproduce the results in the paper. We were able to reproduce most of the results.

1 INTRODUCTION

The paper [1] studies the *densest subgraph discovery problem* on large directed graphs. The authors propose exact and approximate algorithms to solve this problem efficiently.

The authors are:

- Chenhao Ma, The University of Hong Kong
- Yixiang Fang, University of New South Wales
- Reynold Cheng, The University of Hong Kong
- Laks V.S. Lakshmanan, The University of British Columbia
- Wenjie Zhang, University of New South Wales
- Xuemin Lin, University of New South Wales

We were able to run the experiments on our own hardware and could reproduce most results.

2 SUBMISSION

The reproducibility submission contained a link to a GitHub repository, a description of the software requirements, and a link to the datasets. Further, it listed the individual commands to generate the results and described how to interpret the output to reproduce the figures and tables in the paper. While we were able to run all commands, we found the documentation insufficient to interpret the results. Going forward, we recommend that the authors add a single script that generates all of the commands and processes the output into a more readable format.

Submission contents:

- GitHub repository with code and README: https://github.com/chenhao-ma/SIGMOD2020DDS
- Datasets: https://drive.google.com/file/d/184NwGPLhWjozCNKMM_OcNGGEk50iQalr/view

3 HARDWARE AND SOFTWARE ENVIRONMENT

Table 1 shows the environments used in the paper and in the reproducibility review.

Table 1. Hardware & Software environment

	Paper	Repro Review
CPU	Intel Xeon Silver 4110	Intel Xeon Gold 6230
cores	16	40
GHz	2.10	2.10
RAM	256GB	256GB
Storage	6001GB TOSHIBA MG04ACA6 HDD	3.5TB Seagate ST4000NM0115 HDD
Compiler	gcc 7.5.0	gcc 10.2.0
Packages	Boost 1.66.0, CMake 2.8	Boost 1.72.0, CMake 3.19.1

4 REPRODUCIBILITY EVALUATION

4.1 Process

Reproducing the results involved compiling the code with the provided CMake file and running a single binary with different commandline parameters. It was straightforward to generate the runtime results, even though some algorithm and dataset combinations took up to 400 hours to complete. Reproducing statistics shown in various tables in the paper involved parsing and combining the output, which took some manual effort. We thank the authors for helping us to resolve these difficulties.

In the process, we found that the original CMake file did not enable CPU-specific optimizations (*march=native*). We discussed this issue with the authors and ran the experiments with native optimizations.

4.2 Results

We were able to reproduce the main findings of the paper. We can confirm that the proposed exact and approximate algorithms outperform the state-of-the-art baselines. However, we could not reproduce the stated relative runtime differences between the algorithms. Specifically, we found Core-Exact to be between 1× and 8× faster than the state-of-the-art exact algorithm (instead of 2× to 100× as stated in the paper). Similarly, we could not verify that the approximate algorithm KS-Approx is the most efficient one on most datasets. We found Core-Approx to be more efficient. The reason is likely that we compiled the code with CPU-specific optimizations (*march=native*) enabled.

REFERENCES

[1] Chenhao Ma, Yixiang Fang, Reynold Cheng, Laks V. S. Lakshmanan, Wenjie Zhang, and Xuemin Lin. 2020. Efficient Algorithms for Densest Subgraph Discovery on Large Directed Graphs. In *SIGMOD Conference*. ACM, 1051–1066.