Reproducibility Report for ACM SIGMOD 2020 Paper: “Factorized Graph Representations for Semi-Supervised Learning from Sparse Data”

KYRIAKOS PSARAKIS & ASTERIOS KATSIFODIMOS, TU Delft, Netherlands

In general, the inspected work is reproducible with instructions on running parts of the work. The process did run into some issues regarding some outdated scripts. However, the authors were very eager to help and addressed all of them. Thus, we conclude that this work is fully reproducible.

1 INTRODUCTION

The paper we reproduced is Factorized Graph Representations for Semi-Supervised Learning from Sparse Data by Krishna Kumar P. (IIT Madras), Paul Langton (Northeastern University), and Wolfgang Gatterbauer (Northeastern University) [1]. The presented work’s experiments are split into two parts. In the first, the authors perform tests with generated data using commodity hardware (Figures 5 and 6). All the scripts are provided, and we could reproduce the figures with the provided data and generate the data themselves with the provided code. In the second part, the authors used an HPC cluster on real-world data (Figures 7 and 8) but, some issues appeared regarding the code parallelization and some outdated code. However, after contacting the authors, they gave us an updated code that would scale up as expected and addressed all our comments as fast as they could. The overall experience interacting with the authors as well as their artifacts was quite positive.

2 SUBMISSION

The authors’ submission contains a Github repository with instructions, the scripts for the first part of the experiment (Figures 5 and 6), and the second part’s non-parallelized scripts (Figures 7 and 8). In all of the scripts, the parameters used in the paper were provided. Furthermore, Jupyter notebooks were employed to visualize the entire experimental pipeline better. Finally, all the data used were given in the form of pre-generated CSV files, and the code that generated them for the synthetic datasets and all the real-world data were uploaded in cloud storage. The links for the resources mentioned above are given below.

- GitHub repository with code and scripts at:
  https://github.com/northeastern-datalab/factorized-graphs
- Detailed readme file at:
  https://github.com/northeastern-datalab/factorized-graphs/blob/master/Readme.md
- Reproducibility instructions at:
  https://github.com/northeastern-datalab/factorized-graphs/blob/master/reproducibility.md
- Data generators at:
  https://github.com/northeastern-datalab/factorized-graphs/blob/master/sslh/graphGenerator.py
- Real-world data at:
  https://drive.google.com/drive/folders/1fqTgfW8f-PUwnAj432YgsFVjgbUdOHuu

3 HARDWARE AND SOFTWARE ENVIRONMENT

In the paper, the authors used two different setups. Table 1 displays the hardware used with the synthetic (Hardware1), real-world datasets (Hardware2), and the hardware used in our reproducibility.

The hardware that we used is a commodity laptop with comparable hardware for the first part and the SurfSara HPC cluster [1] for the second. A minor issue that we found is that even though the authors state that the runtime environment is python3, we tried with python3.8, and some errors came up during the setup. We tested with python3.6, and that eliminated the error. However, the authors had noted in a different place that the project is built in python 3.6, and that is how we figured it out.

Table 1. Hardware & Software environment

	Paper Hardware1	Paper Hardware2	Repro Review HW1	Repro Review HW2
CPU	Intel Core i5	Intel E5-2680 v4	AMD Ryzen 7 4800H	SurfSara HPC
cores	2	14 per CPU	8	60
GHz	2.5	2.4	2.9	2.1
RAM	16GB	256GB	16GB	64GB
Storage	SSD	SSD	SSD	HDD

4 REPRODUCIBILITY EVALUATION

4.1 Process

Other than the minor issue expressed in Section 3, the runtime environment's setup with all the required dependencies went as expected. Following the reproducibility instructions, it was easy to reproduce all the paper figures (5, 6, and 7) using the authors-provided output. The next thing that we tried is to reproduce the data used by the figure generation process and not use the author-provided ones. Again this is simple with the provided scripts but, we run into an issue regarding the real-world data. The scripts that the authors provided at first had a problem with outdated code and a bug in the parallelization part, so we came in contact with the authors detailing our issues and findings up to that point. Then the authors said that they would get the code that they ran in their cluster and match it with what they had in the provided repository. This fixed the error regarding the outdated code, and the parallelization one remained. We came in contact with them about this again, and they said that it did not appear in the test, and to our understanding, it is related to the fact that our cluster's configuration is different. The authors then worked on a fix so the code will work on both our clusters and upload it. Consequently, after these two fixes by the authors and many repetitive runs on our side, figuring out all the issues, we can safely conclude that the provided artifact as a whole is reproducible.

4.2 Results

In general, after our comments, all of the paper's findings were reproducible. Thus, we conclude that the work is fully reproducible.

REFERENCES

[1] Krishna Kumar P., Paul Langton, and Wolfgang Gatterbauer. 2020. Factorized Graph Representations for Semi-Supervised Learning from Sparse Data. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (Portland, OR, USA) *(SIGMOD '20)*. Association for Computing Machinery, New York, NY, USA, 1383–1398. https://doi.org/10.1145/3318464.3380577

[1] https://userinfo.surfsara.nl/systems/hpc-cloud