Università
degli Studi
di Padova

Dipartimento
di Ingegneria
dell'Informazione

INFORMATION ENGINEERING DEPARTMENT

MASTER'S DEGREE IN COMPUTER ENGINEERING

APEROL from Networks: Analyzing Pipeline and Embedding Representations for Optimized Learning (from Networks)

Professor
Fabio Vandin

Students
Marco Annunziata, 2160851
Silvia Mondin, 2141201
Sveva Turola, 2160852

ACADEMIC YEAR 2025-2026

Contents

1 Datasets

Before proceeding to a discussion of the preprocessing of the datasets, it is necessary to inform readers that changes have been made to the Pennsylvania dataset in the proposal. The dataset from [1] has been used instead. The decision to change the dataset was made to have a directed graph, instead of an undirected one, so to have more uniformity among road networks.

Due to concerns about the large number of edges of the Twitch dataset during the first experiments, it has been replaced with a smaller Deezer dataset from [2].

We also want to provide more information about the two "Mus Musculus" and "Saccharomyces cerevisiae" datasets. Both of them have been downloaded from the STRING Database [3], filtered with the following options: > 0.4 confidence score for Mus Musculus, > 0.7 for Saccharomyces cerevisiae; only AB pairs (undirected).

What follows is the complete updated table (changes highlighted):

Network	$\lvert V \rvert$	$\lvert E \rvert$	**Type**
Pennsylvania [4]	1,088,092	3,083,796	Directed
Padua (province) [5]	122,680	304,184	Directed
Hong Kong (city) [6]	43,620	91,542	Directed
Italian Covid-19 Retweet Network [7]	221,574	800,000	Directed
Deezer [2]	143,884	846,915	Undirected
GitHub Developers [8]	37,700	289,003	Undirected
Mus Musculus Protein Interactions [3]	20,969	800,000	Undirected
Saccharomyces cerevisiae Protein Interactions [3]	5,786	100,000	Undirected
Bio-grid-fission-yeast [9]	2,000	25,300	Undirected

Table 1: Datasets used in the project

A preprocessing pipeline is employed where all datasets are converted into CSV files and only useful features are kept, if present. Undirected edges (u, v) are represented using a pair of directed edges (u, v) and (v, u). Unweighted datasets are represented as having a uniform weight of 1.0 for all edges. Finally, if the node IDs are not numeric, they are mapped to numeric consecutive IDs starting from 0. The Python script that performs this preprocessing step is available in the project's Github repository.

2 Experiments

We provide additional details on the train-validation-test split in our pipeline. Consider graph $G = (V, E)$, where V denotes the set of vertices and E is the set of edges. A first split of the set of edges E is needed for the embedding algorithms, creating E_{embed} and E_{class}. E_{embed} is used for the subgraph $G_{\text{embed}} = (V, E_{\text{embed}})$ given as input to the embedding methods: the shallow embedding methods will use it to learn a lookup table Z of node embeddings, whereas the deep embedding methods aim to learn a function f that can generalize well. The remaining edges in E_{class} are instead reserved for the classification task, where the second split occurs: from E_{class} the sets E_{train}, E_{val}, E_{test} are created for the training, validation and test of the classification model.

To train and evaluate the classification models, labeled *negative* edges are added to each of the three splits. A negative edge in this setting is defined to be an edge $(a, b) \notin E$ with $a, b \in V$. Furthermore, in order to avoid data leakage, the sampled negative edges must be distinct from those used internally by the embedding algorithms during training.

For each edge (u, v) belonging to any of these splits of E_{class}, its corresponding embedded representation is constructed by combining the embeddings z_u, z_v of its incident nodes, using an operator such as the average, the dot product, or concatenation. These edge embeddings are the ones used as input to the classification models.

A feasible split for a large graph could be: (from E) 80% for E_{embed} and 20% for E_{class}; (from E_{class}) 60% for E_{train}, 20% for E_{val} and 20% for E_{test}.

We revised the implementations that will be adopted for our experiments: [10] for Node2Vec, [11] for LINE, [12] for DVNE, [13] for GraphSage. Depending on our specific needs, some of these could be modified or used as an inspiration for our own implementations. In particular, DVNE was not evaluated on directed graphs in the original work [14], meaning it may be necessary to explore adaptations for this setting. If no satisfactory solution is found, DVNE will be restriced only for undirected graphs to mantain a fair comparison.

Following some initial experiments, we decided to use the DEI cluster *"Blade"* as hardware for the computation of the larger graphs. The specifications for the nodes of the cluster are available in the open documentation [15]. More information about the project's memory usage and runtime will be available in the final report.

References

[1] Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford Large Network Dataset Collection.* `http://snap.stanford.edu/data`. June 2014.

[2] Benedek Rozemberczki et al. "GEMSEC: Graph Embedding with Self Clustering". In: *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2019.* ACM. 2019, pp. 65–72. URL: `https://snap.stanford.edu/data/gemsec-Deezer.html`.

[3] Damian Szklarczyk et al. "The STRING database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest". In: *Nucleic Acids Research* 51.D1 (2023), pp. D638–D646. DOI: `10.1093/nar/gkac1000`. URL: `https://string-db.org/cgi/download`.

[4] A. Dasgupta J. Leskovec K. Lang and M. Mahoney. "Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters". In: *Internet Mathematics* 6.1 (2009), pp. 29–123. URL: `https://snap.stanford.edu/data/roadNet-PA.html`.

[5] Marco Annunziata. *Padua_Network_dataset_2025.* 2025. URL: `https://github.com/Remdox/Padua_Network_dataset_2025`.

[6] yzengal. *RoadNetwork-China-City.* 2021. URL: `https://github.com/yzengal/RoadNetwork-China-City/blob/main/Hongkong.road-d.tar.gz`.

[7] V. Mesina et al. *Italian Covid-19 Retweet Network (2020-2022) [Data set].* 13th International Conference on Complex Networks and Their Applications (CNA2024), Istanbul, Turkey. Zenodo. 2024. URL: `https://doi.org/10.5281/zenodo.13909011`.

[8] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. *Multi-scale Attributed Node Embedding.* 2019. arXiv: `1909.13021 [cs.LG]`. URL: `http://snap.stanford.edu/data/github-social.html`.

[9] Ryan A. Rossi and Nesreen K. Ahmed. "The Network Data Repository with Interactive Graph Analytics and Visualization". In: *AAAI.* 2015. URL: `https://networkrepository.com/bio-grid-fission-yeast.php`.

[10] palash1992. 2022. URL: `https://github.com/palash1992/GEM`.

[11] tangjianpku et al. 2019. URL: `https://github.com/tangjianpku/LINE`.

[12] Lakshya-99. 2020. URL: `https://github.com/Lakshya-99/Deep_Variational_Network_Embedding/tree/master`.

[13] William L. Hamilton et al. 2018. URL: `https://github.com/williamleif/GraphSAGE`.

[14] Dingyuan Zhu et al. "Deep Variational Network Embedding in Wasserstein Space". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '18. London, United Kingdom: Association for Computing Machinery, 2018, pp. 2827–2836. ISBN: 9781450355520. DOI: `10.1145/3219819.3220052`. URL: `https://doi.org/10.1145/3219819.3220052`.

[15] Università degli Studi di Padova. *Cluster "Blade"*. URL: `https://docs.dei.unipd.it/en/CLUSTER/Overview`.

Contribution of Authors and AI Usage

The contributions for this project are:

- Datasets preprocessing: Silvia Mondin (implementation, writing);

The following AI tools were used:

- Copilot was used to suggest code and report snippets.
- DeepL was used to correct the grammar and syntax of the text.