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 Motivation

Graphs have emerged as a natural model for the representation and analysis of data and complex systems in various domains. For instance, link prediction in social networks can help to understand the spreading process of rumors or epidemics [6], and in biological networks, it is commonly used for predicting novel interactions between proteins [2].

However, most traditional machine learning algorithms are not designed to work directly with graph-structured data, as they require numeric vectors or matrices as inputs. To address this challenge, graph embedding techniques have been developed to learn low-dimensional vector representations of nodes, edges, or entire graphs while preserving their topological properties.

Consequently, in order to design an effective graph embedding technique, it is necessary to consider several criteria [1]. Two of these are adaptability and scalability. An embedding is considered adaptable if it can be utilized for various data and tasks without the need for retraining. The scalability of an embedding is determined by its capacity to process large-scale networks within a reasonable amount of time.

The objective of this study is to evaluate known embeddings from the perspective of these two criteria.

From the perspective of adaptability, the objective is to determine the potential impact of the graph domain on the precision of an embedding. In other words, the objective is to ascertain whether there exist approaches that are better suited for a particular field, whether there is one embedding that consistently outperforms the others, or if the embeddings are comparable. Eventually, we would like to assess the adaptability of these embeddings when they are applied to different tasks.

In the context of scalability, the objective is to identify the embeddings that are better suited to process large graphs. The objective of this study is to evaluate the tradeoff between the accuracy of the predictions and the time required for training and inference.

2 Datasets

The datasets that will be used in this project are nine and they are divided into three categories: Road Networks, Social Networks, and Biological Networks. For each of these categories three datasets of different size were chosen: one small (~40k nodes), one medium (~100k nodes), and one large (~1M nodes). The datasets, with their main properties (number of nodes, number of edges, and type of graph) are reported in the table 1.

Network	$\|\boldsymbol{V}\|$	$\|\boldsymbol{E}\|$	**Type**
Pennsylvania [7]	1.088.092	1.541.898	Undirected
Padua (province) [4]	122.680	164.737	Directed
Hong Kong (city) [10]	43.620	91.542	Directed
Italian Covid-19 Retweet Network [5]	0	800.000	Directed
Twitch Gamers [9]	168.114	6.797.557	Undirected
GitHub Developers [8]	37.700	289.003	Undirected
Mus Musculus Protein Interactions	0	800.000	Undirected
Saccharomyces cerevisiae Protein Interactions	0	100.000	Undirected
Bio-grid-fission-yeast [7]	2000	25.300	Undirected

Table 1: Datasets used in the project

3 Methods

Node embedding is the task of mapping nodes into an embedding space, where each node v is represented by a vector $\mathbf{z_v} \in \mathbb{R}^d$ using the mapping function $ENC : v \to \mathbf{z}_v$. The encoding of nodes into embeddings has to be such that similarity between each pair of nodes is kept in the correspondent embeddings. In order to achieve this, a similarity score $S(u, v)$ is computed between pairwise nodes and a decoder function $DEC : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is used for computing a similarity score for the embeddings. The discrepancy between the two scores is then compared and minimized by means of a loss function, such as the empirical loss:

$$\mathcal{L} = \sum_{i,j} \ell(DEC(\mathbf{z}_u, \mathbf{z}_v), S(u, v))$$

Based on this context, the specifics of which encoder and decoder is used is what really differentiates the node embedding methods. In this project, four node embedding methods will be employed:

- Node2Vec, a shallow encoding which is a tunable version of the DeepWalk algorithm, which is often used in the scientific literature as the baseline for comparing node embedding methods.
- LINE, a shallow encoding which is am highly-scalable approximation algorithm.
- Deep Variational Network Embedding in Wasserstein Space, a deep encoding method using a Variational Autoencoder. This metod has a particular approach of encoding each node to a probability distribution, so in this case the similarity score between embeddings is a distance between probability distributions, which can be measured using the Wasserstein distance.
- GraphSage, a deep encoding method making use of a GNN that aggregates the features of each node's neighborhood and learns to generate node embeddings from that.

Given the graph embeddings, we use them as input to ML models to make predictions, depending on the task to solve. In the case of this project and the datasets used, the main task is link prediction. Link prediction is the problem of predicting a link between two nodes in a network. The models that will be used for solving the task are:

- Support Vector Machines
- Random Forest
- Multilayer Perceptron

4 Experiments

To evaluate the methods used in this project, two different evaluation measures will be used: the Area Under the Receiver Operating Characteristic curve (AUROC), and the Area Under the Precision-Recall curve (AUPR). AUROC is historically considered the primary performance metric used to evaluate the performances of Link Prediction methods [3]. However, AUROC favors accurate classification of positive examples, at the cost of misclassifying the negative ones. In a scenario like Link Prediction problem, which is inherently skewed towards the negative class, it may not be suitable and can overestimate the performance of the methods. For this reason, the AUPR curve was also selected, as it can provide better evaluation of Link Prediction in the presence of class imbalance.

5 References

[1] Anthony Baptista et al. "Zoo guide to network embedding". In: *Journal of Physics: Complexity* 4.4 (2023), p. 042001.

[2] Björn H Junker and Falk Schreiber. *Analysis of biological networks*. Vol. 2. Wiley Online Library, 2008.

[3] I. Bhargavi Kalyani, A. Rama Prasad Mathi, and Niladri Sett. "Evaluating link prediction: new perspectives and recommendations". In: *International Journal of Data Science and Analytics* 20.7 (2025), pp. 6855–6886. ISSN: 2364-4168. DOI: 10.1007/s41060-025-00858-0. URL: https://doi.org/10.1007/s41060-025-00858-0.

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

[5] 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.

[6] Romualdo Pastor-Satorras et al. "Epidemic processes in complex networks". In: *Reviews of modern physics* 87.3 (2015), pp. 925–979.

[7] Ryan A. Rossi and Nesreen K. Ahmed. "The Network Data Repository with Interactive Graph Analytics and Visualization". In: *AAAI*. 2015. URL: https://networkrepository.com.

[8] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. *Multi-scale Attributed Node Embedding*. 2019. arXiv: 1909.13021 [cs.LG].

[9] Benedek Rozemberczki and Rik Sarkar. *Twitch Gamers: a Dataset for Evaluating Proximity Preserving and Structural Role-based Node Embeddings*. 2021. arXiv: 2101.03091 [cs.SI].

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

6 Contribution of Authors and AI Usage