Are Graph Embeddings the Panacea?

– an Empirical Survey from the Data Fitness Perspective

Qiang Sun
Du Q. Huynh Mark Reynolds Wei Liu

pascal.sun@research.uwa.edu.au
{du.huynh, mark.reynolds, wei.liu}@uwa.edu.au

Computer Science and Software Engineering
The University of Western Australia

November 22, 2024

THE UNIVERSITY OF WESTERN AUSTRALIA

Contents

Problem Statement

THE UNIVERSITY OF WESTERN AUSTRALIA

Graph Structure and Network Datasets

Co-existence Networks:

Actors, Amazon Computer, Photos and Ratings
Coauthor CS and Physics, Tolokers, Flickr, Questions

Citation Networks:

CiteSeer, Cora, DBLP, PubMed
WebKG (extended version, nodes hyperlinked)

Social Networks:

BlogCatalog, Github, Twitch (friends or followers)

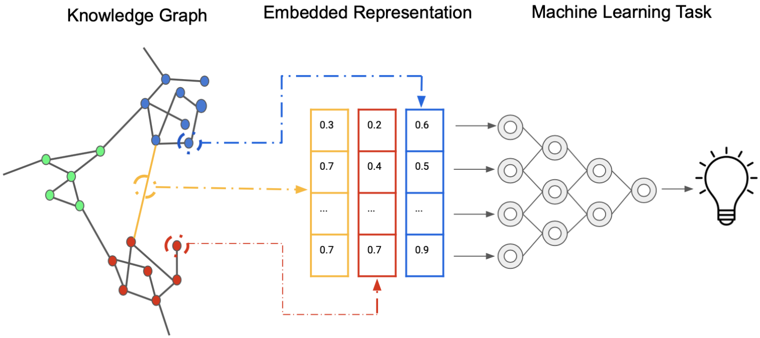
Grid Network:

Minesweeper (adjacent nodes)

Knowledge Graphs:

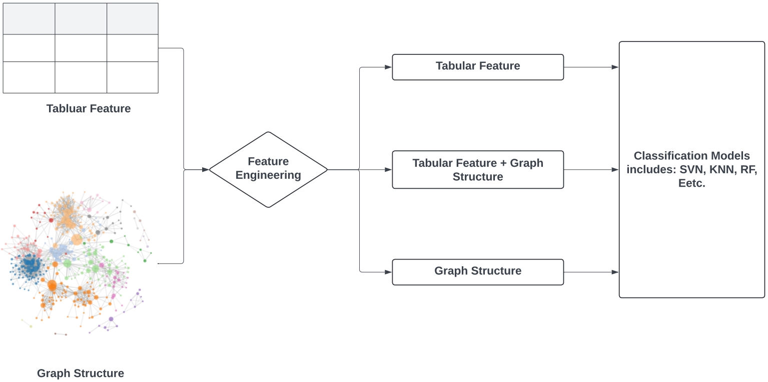
Roman Empire (mini knowledge graph)
Wiki (abstracted into a graph)

Graph Representation Learning for Machine Learning Tasks



Workflow for graph data

Feature Engineering



Feature engineering workflow

Research Questions

For node classification problems

Q1. Is there a potential benefit of applying graph representation learning?

Q2. Is structural information alone sufficient?

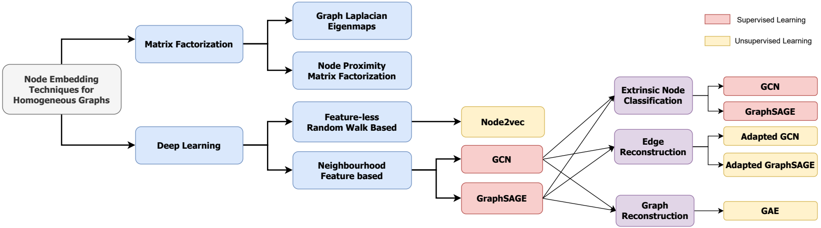
Q3. Which embedding technique would best suit my dataset?

Literature Review

Previous Surveys: Graph embeddings and KG embeddings

- 2017-Now, 6 surveys[1] about graph embeddings and knowledge graph ebemddings
- Objective of the surveys are focusing on graph embedding algorithms and applications.
- There is a lack of research effort on quantifying **the intricate relationship between specific network structure features and the performance of different graph embedding techniques.**

[1] Cai et al., Chen et al., Goyal and Ferrara, Makarov et al., Xu



Node Embedding Techniques for Homogeneous Graphs.[2]

[2]Adapted from Cai et al.

Experiment Setup

Representative Network Characteristics

$\|V\|$	Total number of nodes	$\|E\|$	Total number of edges
d	Dimension of node features	K	Number of classes
$\bar{k}$	Average degree of nodes ($= 2\|E\|/\|V\|$)	k_{var}	Second moment of degree distribution
k_{min}	Minimum node degree	k_{max}	Maximum node degree
L	Average shortest path between all node pairs	D	Diameter (The maximum distance between all possible pairs of nodes)
T	Transitivity (measuring likelihood of triangle formation)	C	Average clustering (quantifying the tendency of nodes to cluster together)
γ	Degree exponent of the power-law degree distribution, $P(k) \sim k^{-\gamma}$		

Network characteristics: notations and definitions

Dataset Categories and Network Types

Network Type	Dataset topic	Description
Co-existence Network (7)	Actor (1)	Wikipedia co-occurrence of actors; Classify into categories.
	Amazon (3)	Product co-purchases, bag-of-words from reviews; Product/review categories.
	Coauthor (CS, Physics) (2)	Authorship network, paper keywords; Study fields classification.
	Tolokers (1)	Toloka worker data, shared tasks; Banned worker prediction.
Citation Network (15)	Cora, etc. (12)	Publications, citations, bag-of-words; Publication classes.
	WebKG (3)	University web pages, hyperlinks, bag-of-words; Page categories.
Social Network (8)	Twitch (6)	Streamers, mutual follows, game embeddings; Language use classification.
	BlogCatalog (1)	Blog platform users and friendships, TF-IDF from blogs; User categories.
	Github (1)	Developer relationships, locations, repos; Web/ML developer classification.
Knowledge Graph (2)	Wiki (1)	Wikidata graph, item relations, one-hot vectors; Item categories.
	Roman Empire (1)	Wikipedia articles, word connections, embeddings; Syntactic roles classification.
Social Knowledge Network (2)	Questions (1)	Yandex Q dataset, answered questions, user profile embeddings; Active user prediction.
	Flickr (1)	Users, images, metadata, text annotations; Tag-based classification.
Grid (1)	Minesweeper (1)	Grid cells, adjacent mines; Mine presence prediction.

Dataset Context, where each number inside the parentheses denotes the number of datasets for that given network type or dataset topic.

Network Characterisation of the Datasets

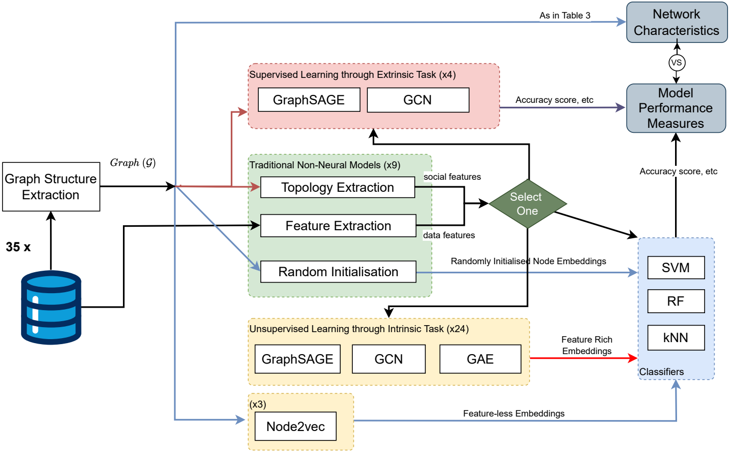
Dataset	$\|V\|$	$\|E\|$	d	K	$\bar{k}$	k_{var}	$k_{\min}$	$k_{\max}$	L	C	T	D	γ
Actor	7,600	30,019	932	5	7.90	400.56	1	1,304	4.11	0.05	0.04	12	2.81
AZ_COMPUTERS	13,752	245,861	767	10	35.76	6,221.40	**0**	2,992	3.38	0.34	0.10	10	2.83
AZ_PHOTO	7,650	119,081	745	8	31.13	3,204.10	**0**	1,434	4.05	**0.40**	0.17	11	2.92
AGD_BlogCatalog	5,196	171,743	8,189	6	66.11	7,376.53	**5**	769	2.51	0.12	0.08	**4**	4.06
AGD_CiteSeer	3,312	4,715	3,703	6	2.85	19.81	1	100	**1.08**	0.07	0.03	28	3.31
AGD_Cora	2,708	5,429	1,433	7	4.00	44.23	1	169	1.17	0.13	0.02	19	3.05
AGD_Flickr	7,575	239,738	**12,047**	9	63.30	21,303.96	1	1,881	2.41	0.33	0.10	4	2.76
AGD_Pubmed	19,717	44,338	500	3	4.50	75.51	1	171	6.34	0.03	0.01	18	4.20
AGD_Wiki	2,405	16,523	4,973	17	13.74	499.01	1	281	3.65	0.32	**0.44**	9	3.82
CF_CiteSeer	4,230	5,337	602	6	**2.52**	20.43	1	85	1.35	0.11	0.08	26	2.83
CF_Cora	19,793	63,421	8,710	**70**	6.41	118.33	1	297	1.16	0.26	0.13	23	3.38
CF_Cora_ML	19,793	63,421	8,710	**70**	6.41	118.33	1	297	1.16	0.26	0.13	23	3.38
CF_DBLP	17,716	52,867	1,639	4	5.96	123.03	1	339	1.16	0.13	0.10	34	3.23
CF_PubMed	19,717	44,324	500	3	4.49	75.43	1	171	6.34	0.06	0.05	18	4.17
CiteSeer	3,327	4,552	3,703	6	2.73	18.92	**0**	99	1.08	0.14	0.13	28	2.63
Coauthor_CS	18,333	81,894	6,805	15	8.93	162.75	1	136	5.42	0.34	0.18	24	5.18
Coauthor_Physics	34,493	247,962	8,415	5	14.38	449.23	1	382	5.16	0.37	0.18	17	4.88
Cora	2,708	5,278	1,433	7	3.89	42.53	1	168	1.17	0.24	0.09	19	2.98
Cora_Full	19,793	63,421	8,710	**70**	6.41	118.33	1	297	1.16	0.26	0.13	23	3.38
GitHub	37,700	289,003	128	**2**	15.33	6,761.61	1	**9,458**	3.25	0.17	0.01	11	2.54
HGD_AZ_Ratings	24,492	93,050	300	5	7.60	93.39	**5**	132	16.24	0.29	0.15	46	3.60
HGD_Minesweeper	10,000	39,402	**7**	**2**	7.88	62.44	3	**8**	46.67	0.22	0.33	99	2.79
HGD_Questions	**48,921**	153,540	301	**2**	6.28	774.50	1	1539	4.29	**0.02**	0.01	16	**1.83**
HGD_Roman_empire	22,662	32,927	300	18	2.91	**9.50**	2	14	**2331.56**	0.19	0.28	**6,824**	**6.34**
HGD_Tolokers	11,758	**519,000**	10	**2**	**88.28**	**33,978.74**	1	2,138	2.78	0.27	0.11	11	3.08
PubMed	19,717	44,324	500	3	4.50	75.43	1	171	6.34	0.06	0.05	18	4.18
TWITCH_DE	9,498	162,636	128	**2**	34.25	8,363.44	3	4,261	2.72	0.20	0.05	7	2.58
TWITCH_EN	7,126	42,450	128	**2**	11.91	634.28	3	722	3.68	0.13	0.04	10	2.79
TWITCH_ES	4,648	64,030	128	**2**	27.55	3,198.91	3	1,024	2.88	0.22	0.08	9	2.58
TWITCH_FR	6,551	119,217	128	**2**	36.40	7,328.28	2	2,042	2.68	0.22	0.05	7	2.61
TWITCH_PT	1,912	33,211	128	**2**	34.74	4,324.69	3	769	2.53	0.32	0.13	7	2.53
TWITCH_RU	4,385	41,689	128	**2**	19.01	2,098.57	3	1,231	3.02	0.17	0.05	9	2.58
WEBKB_Cornell	**183**	**298**	1,703	5	3.26	60.52	1	94	3.20	0.10	0.01	8	3.09
WEBKB_Texas	**183**	325	1,703	5	3.55	75.22	1	104	3.03	0.11	**0.01**	8	2.62
WEBKB_Wisconsin	251	515	1,703	5	4.10	81.85	1	122	3.26	0.11	0.01	8	2.50

Graph Representation Learning

- Node2vec[3]
- GraphSAGE (SAmple and aggreGatE)[4]
- GCN (Graph Convolutional Network)[5]
- GAE (Graph Auto Encoder)[6]

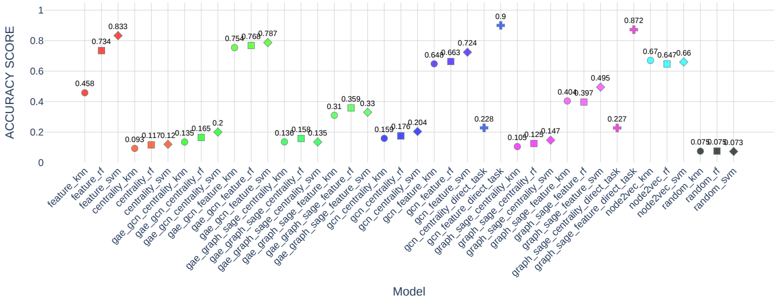
[3] Grover and Leskovec
[4] Hamilton et al.
[5] Pei et al.
[6] Goyal and Ferrara

Survey Experiment Design



Results and Discussion

Results: Dataset Coauthor CS accuracy score



Dataset Coauthor CS accuracy score for models[7]

[7] Code and results available: https://github.com/PascalSun/PAKDD-2024

Q1: Do all datasets benefit from graph structure representation learning?

Evaluation metric: Feature Only models with those integrating node features and graph structure

Results:

- 21 of 35 datasets showed improved performance with graph structures (e.g., Minesweeper, Roman Empire, Twitch).
- Datasets like PubMed, Wiki, and Flickr performed better with Feature Only models.

Highlight: The PubMed dataset has the longest average shortest path among the citation networks, potentially explaining its exceptional performance.

Conclusion: Graph structures enhance performance in many cases but are not always superior.

Q2: Is structural information alone sufficient?

Evaluation Metric: *Structure Only* models with those integrating node features and graph structure.

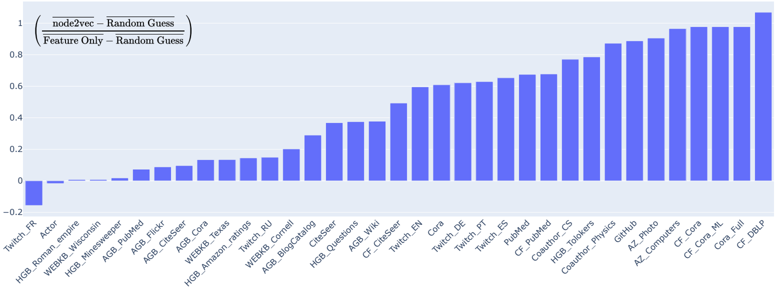
Findings:

- Combined models outperform *Structure Only* models.
- *Node2vec* is on par with *Feature Only* for dense networks.

Highlight: Short-diameter networks favor *Structure Only* models.

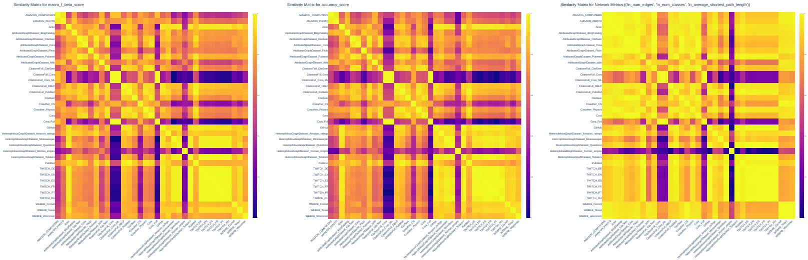
Conclusion: Structural data alone falls short; *Node2vec* excels in specific dense networks.

Q2: Is structural information alone sufficient?



Comparison of classification accuracies of the **node2vec** models versus the **Feature Only** models for all the datasets (the bar symbol represents the average accuracy of the model over the SVM, RF, and KNN classifiers).

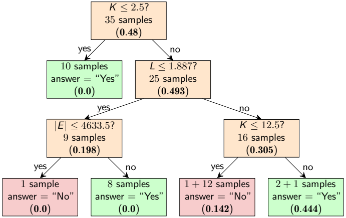
Q3: Which graph embedding representation model(s) suit my dataset?



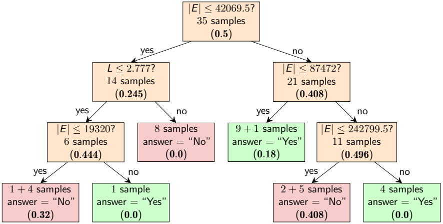
Similarity matrices for: F1 score (left), Accuracy score (middle), Network Parameters ($|E|$, K, L) (right).

Q3: Which graph embedding representation model(s) suit my dataset?

Q1: Do datasets benefit from graph embedding learning?



Q2: Is structural information alone sufficient?



Decision Trees for Q1 and Q2 based on $\mathbf{v}'_{\text{net}} = (|E|, K, L)$. The boldface numbers inside parentheses denote the Gini indices. Each leaf node is coloured in green (or pink) for the "Yes" (or "No") answer to the question.

THE UNIVERSITY OF WESTERN AUSTRALIA

Conclusion and Future Work

Graph Network Type and Model Performance

Sparse Networks: These networks have long average shortest path lengths, a large number of classes, and limited edges, making them generally unsuitable for classification tasks via graph representation learning.

Dense Networks: Characterized by their well-connected nature and rich edge information, these networks show excellent performance with random walk-based models such as node2vec, especially compared to attribute-only models.

Best Performers: The most effective models are those that integrate neighbor-attribute-based supervised learning, consistently outperforming others by effectively leveraging both structural and attribute information.

Future work

Enhance Dataset Diversity: Address the dominance of citation networks by expanding to more varied domains, which is crucial for improving dataset diversity.

Advance Feature Representation: Move beyond bag-of-words and one-hot encoding for textual attributes. Implement semantic-enriched content embeddings using Large Language Models (LLM) to leverage recent advancements in this field.

Explore Deeper Networks: While node2vec shows impressive performance, there is a need to develop and study deeper networks inspired by random walks, as our current non-random walk-based models are limited to two layers.

Q & A

THANK YOU

Q & A

References

[1] H. Cai, V. W. Zheng, and K. C.-C. Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. **IEEE Transactions on Knowledge and Data Engineering**, 30 (9):1616–1637, 2018. ISSN 1041-4347, 1558-2191, 2326-3865. doi: 10.1109/TKDE.2018.2807452.

[2] S. Chen, S. Huang, D. Yuan, and X. Zhao. A Survey of Algorithms and Applications Related with Graph Embedding. In **Proceedings of the 2020 International Conference on Cyberspace Innovation of Advanced Technologies**, pages 181–185, Guangzhou China, Dec. 2020. ACM. ISBN 978-1-4503-8782-8. doi: 10.1145/3444370.3444568.

[3] P. Goyal and E. Ferrara. Graph embedding techniques, applications, and performance: A survey. **Knowledge-Based Systems**, 151:78–94, 2018. ISSN 09507051. doi: 10.1016/j.knosys.2018.03.022.

[4] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, pages 855–864. ACM, 2016. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939754.