On the Interpretability and Evaluation of Graph Representation Learning

Antonia Gogoglou
Capital One
McLean, VA 22102
antonia.gogoglou@capitalone.com

C. Bayan Bruss
Capital One
McLean, VA 22102
bayan.bruss@capitalone.com

Keegan E. Hines
Capital One
McLean, VA 22102
keegan.hines@capitalone.com

Abstract

With rising interest in graph representation learning, a variety of approaches have been proposed to effectively capture a graph's properties. While these approaches have improved performance in graph machine learning tasks compared to traditional graph mining techniques, they are still perceived as *black-box* techniques with limited insights into the information encoded in these representations. In this work, we explore methods to interpret node embeddings and propose the creation of a robust evaluation framework for comparing graph representation learning algorithms and hyperparameters. We test our methods on graphs with different properties and investigate the relationship between embedding training parameters and the ability of the produced embedding to recover the structure of the original graph in a downstream task.

1 Introduction

Graphs play a key role in many machine learning tasks providing the structured information needed to learn meaningful patterns and generate predictive models. However, it is challenging to represent complex structures like graphs in an expressive and efficient way that they can be fed into machine learning applications. Advances in the field of Graph Representation Learning [8, 5] appear to provide a mapping that embeds nodes, or entire graphs, as dense low dimensional vectors. Recently proposed approaches such as DeepWalk [13], LINE [16], node2vec [6], GCNs [10] and GraphSage [7] treat this mapping as a machine learning task itself and aim to optimize it so that relationships in the embedding space accurately reflect the topology of the original graph.

A common categorization distinguishes between *shallow* and *deep* node embeddings. Shallow embeddings rely on first- or higher-order proximity derived from the original graph, often via random walks, to provide the context of a node and inform its representation. Deep learning approaches include Graph Convolutional Networks (GCNs) and Message Passing Neural Networks (MPNNs) which extend the concept of convolution to describe a node as a function of its neighborhood. Regarding the objective to optimize during training of the embeddings, unsupervised approaches optimize for link reconstruction, supervised approaches for an externally assigned node label and semi-supervised operate on a subset of labeled nodes.

Given that different embedding approaches optimize for different objectives and operate on different input, it is expected that there is not a single "one-fits-all" node embedding technique. Recent work

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

has focused on evaluating graph representation learning techniques with regards to their ability to distinguish graph properties [4, 18]. In this direction, we investigate the interpretability of node embeddings and propose an evaluation framework that answers the following questions:

- What information do node embeddings express and can we derive metrics to quantify their properties?
- How can we evaluate node embeddings with or without external labels and is there a single approach that maximizes performance across all tasks?
- Can complicated structures of the original graph be captured in embeddings trained on the local context around a node?

The rest of the paper is organized as follows: Section 1.1 describes our proposed methodology, while Section 2 shows the results of our experiments and concludes the article.

1.1 Methods

1.1.1 Interpretability

In graph representation learning, nodes are typically embedded into a fixed D dimensional vector space (where D is a hyperparameter) Theoretically, the space is as condensed of a representation as we can get, without loss of information. This indicates that an *interpretable* embedding dimension would be highly associated with a particular feature of the original graph, a so-called disentangled representation [9, 2, 11]. In NLP these features are often expressed in the form of semantic categories of words [15, 12]. In the case of graphs such categories can be derived from extrinsic or intrinsic sources, with the former being categories or labels assigned externally to nodes while the latter refers to groups found in the decomposition of the original graph (e.g. communities or partitions).

We define an *Interpretability Score* adapted from [15] for each dimension and each group of nodes:

$$IS_{top(d,l)} = \frac{|C_l \cap top_k(E_d)|}{|C_l|} \times 100 \quad IS_{bottom(d,l)} = \frac{|C_l \cap bottom_k(E_d)|}{|C_l|} \times 100 \tag{1}$$

where C_l is the l_{th} group of nodes and E_d is the d_{th} embedding dimension, while k is a hyperparameter set equal to the cardinality of C_l for our experiments. Interpretability scores are produced for both the top and bottom nodes at each embedding dimension and they can be aggregated by taking the maximum or average. Thereafter, scores are aggregated either across multiple groups to get the score for a single embedding dimension or across embedding dimensions to obtain per group scores.

$$IS_d = \underset{l=0\ to\ L}{agg_1}(agg_2(IS_{top(d,l)}, IS_{bottom(d,l)})) \quad IS_l = \underset{d=0\ to\ D}{agg_1}(agg_2(IS_{top(d,l)}, IS_{bottom(d,l)})) \tag{2}$$

If the top nodes in the positive or negative direction of an embedding dimension are highly associated with a particular node category and at the same time have lower overlap with the rest of the categories, then the interpretability of this dimension is strong.

1.1.2 Embedding approaches and Datasets

In random-walk based embedding models, there are two general components, a system for generating long random walks (with some variants depending on the model), and a shallow one layer neural network skip-gram model. Each of the components contains a set of hyperparameters out of which the most commonly reported one is embedding dimensionality.

To investigate the proposed evaluation methods we use three datasets: one coming from the financial sector (*Brand Level Merchants - BLM*) [3] and two from the social networks sphere (*BlogCatalog*) and (*Flickr*) [17]. The BlogCatalog dataset contains friendship connections between bloggers. Additionally, it contains labels for each node referring to 39 categories the bloggers could be affiliated with. Similarly, Flickr data contains links between users of the Flickr board and 195 categories users can be associated with. The Brand Level Merchant dataset is constructed from credit card transaction logs. By taking any two transactions that share an account within a specified time window, a set

Table 1: Dataset Statistics

	Number of nodes	Number of edges	Density	Number of communities
Brand Level Merchants	over 100,000	over 8×10^6	1.2×10^{-3}	400 (80 for 95% of nodes)
BlogCatalog	10,312	333,983	6.3×10^{-3}	6
Flickr	80,513	5,899,882	1.18×10^{-3}	17

of merchant pairs, meaning walk lengths equal to 2, are generated. For all datasets we generate embeddings using the GENSIM implementation of word2vec [14] with the same hyperparameters proposed in [3] and [13].

1.1.3 External and Internal Evaluation

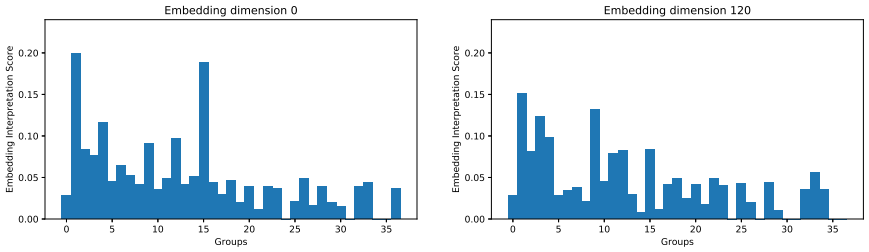
In this work, we focus on evaluating embeddings both internally, meaning their ability to capture graph structure and externally, meaning their distinguishing power against node labels. In the embedding space, similar nodes are expected to be placed closer together, but the notion of similarity can be arbitrarily defined based on node features, neighborhoods or connectivity patterns. Communities are a broadly used way of graph partitioning and can capture complex structural similarity. Consequently, they make a good test case for evaluating how graph structural properties are represented in the embedding space. Two learning problems are generated from this: pairwise community detection, which is a binary classification task of whether a pair of nodes belong in the same community and node level community prediction, which we treat as a multi-class classification problem of predicting the community a node belongs in given its embedding representation. For graphs that contain node labels, like BlogCatalog and Flickr, we treat them the same way. The goal in both tasks is to test the embeddings' efficiency to separate nodes. For community detection we use Louvain's algorithm for optimizing modularity [1].

2 Results

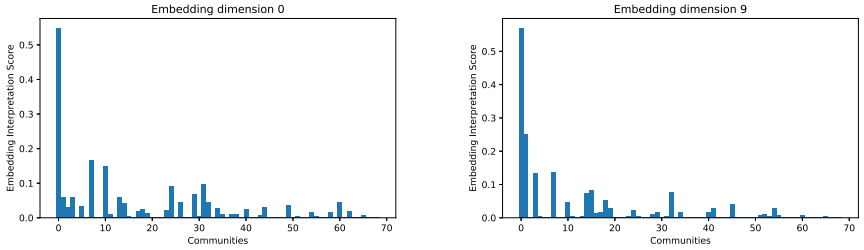
For our experiments we produced embeddings for all datasets with embedding dimensionality of 10, 64 and 128. First, we examine Interpretability Scores (*IS*) aggregated over different axes to explore the association of the embedding space with both external and internal node categorization. Figure 1 shows the distribution of *IS* values over node communities for Brand Level Merchants and over node groups for the BlogCatalog and Flickr graphs. We observe that some node categories are highly associated with multiple dimensions of the embedding space (e.g. community 0 in Brand Level Merchants). These are the most highly populated categories and contain a larger variety of patterns expressed in multiple dimensions. Each embedding dimension appears to also be individually correlated with a particular subset of node categories. For instance the 0^{th} dimension for BlogCatalog is mostly correlated with groups 1 and 15, while the 120^{th} dimension carries information for groups 1, 14 and 3.

Next, we report in Table 2 the performance of different dimensionality embeddings on a set of prediction tasks described in Section 1.1.3. We observe that, by increasing the number of embedding dimensions, the ability to predict community membership does not improve, with an edge given to denser representations in BlogCatalog and Brand Level Merchants. Interestingly, performance in all node classification tasks we undertook is highly linked with Interpretability Scores distribution (see Figure 1), with the highest values being achieved for community prediction over node classification. Performance in external node classification increases with the number of dimensions for the BlogCatalog data, while for Flickr data medium sized embeddings outperform the rest in this task. We can conclude that hyperparameter tuning can be based on two axes: graph properties of the dataset and the structures of the original graph we need the embeddings to capture.

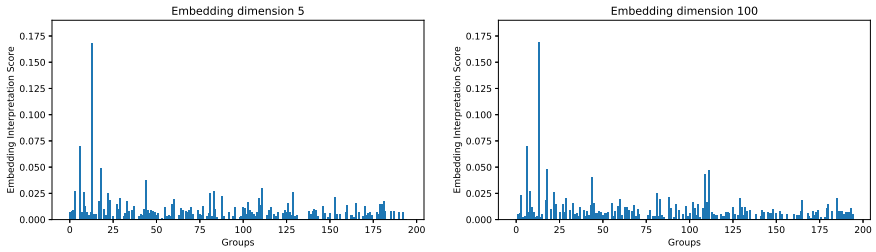
Link prediction accuracy, for which random-walk based approaches optimize, appears to be correlated with external node classification. This is not always the case with community prediction, which favors smaller sized embeddings in BlogCatalog and Brand Level Merchants while link prediction improves with higher number of dimensions in the same datasets. In the Flickr graph different dimensionality embeddings achieve almost identical link prediction AUC scores, but show big



(a) *IS* for the 0^{th} and 120^{th} embedding dimensions across node groups for BlogCatalog data (D = 128)



(b) *IS* for the 0^{th} and 9^{th} embedding dimensions across communities for Brand Level Merchants (D = 10)



(c) *IS* for the 5^{th} and 100^{th} embedding dimensions across communities for Flickr (D = 128)

Figure 1: Interpretability scores over node categorizations for selected embedding dimensions

Table 2: Performance for different classification tasks with various embedding dimensionality values. In binary classification values are *F1-scores*, in multi-class *micro-averaged F1* and LPAUC is Link Prediction *AUC*.

D	Brand Level Merchants			BlogCatalog				
	Community			Group		Community		
	Binary	Multi-class	LP AUC	Binary	Multi-label	Binary	Multi-class	LP AUC
10	**0.78**	0.84	**0.98**	0.55	0.35	0.71	**0.86**	0.87
64	0.71	**0.86**	0.95	0.75	**0.42**	0.68	0.80	0.90
128	0.71	0.85	0.94	**0.78**	0.40	**0.72**	0.83	**0.93**
D				Flickr				
				Binary	Multi-label	Binary	Multi-class	LP AUC
10	–	–	–	**0.70**	0.37	**0.80**	0.85	0.95
64	–	–	–	**0.70**	**0.40**	0.70	0.88	**0.96**
128	–	–	–	0.67	**0.40**	0.77	**0.94**	**0.96**

deviations in performance in community prediction. Our findings imply that optimizing for link occurrence or external labels alone is not always sufficient to evaluate the embedding space as a whole and graph structure based tasks can shed light into the quality of latent representations. This is only the first the step in an effort to design a generalizable evaluation framework for different graph representation approaches across graphs with varying properties.

References

[1] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[2] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[3] C Bayan Bruss, Anish Khazane, Jonathan Rider, Richard Serpe, Antonia Gogoglou, and Keegan E Hines. Deeptrax: Embedding graphs of financial transactions. *arXiv preprint arXiv:1907.07225*, 2019.

[4] Ayushi Dalmia, Manish Gupta, et al. Towards interpretation of node embeddings. In *Companion Proceedings of the The Web Conference 2018*, pages 945–952. International World Wide Web Conferences Steering Committee, 2018.

[5] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.

[6] Aditya Grover and Jure 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.

[7] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[8] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[9] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017.

[10] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[11] Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.

[12] Sungjoon Park, JinYeong Bak, and Alice Oh. Rotated word vector representations and their interpretability. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 401–411, 2017.

[13] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[14] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[15] Lütfi Kerem Şenel, Ihsan Utlu, Veysel Yücesoy, Aykut Koc, and Tolga Cukur. Semantic structure and interpretability of word embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(10):1769–1779, 2018.

[16] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[17] Lei Tang and Huan Liu. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 817–826. ACM, 2009.

[18] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.