TUD Datasets: A collection of benchmark datasets for learning with graphs

Anonymous Authors[1]

Abstract

Recently, there has been an increasing interest in learning with graph data, especially using graph neural networks. However, the development of meaningful benchmark datasets and standardized evaluation procedures is lagging behind. That is, most paper papers evaluate their methods on small-scale datasets leading to high standard deviations and hard to interpret results, consequently hindering advancements in this area. To address this, we introduce the TUD DATASET for graph classification and regression. The dataset consists of over 150 datasets from a wide range of applications and varying sizes. We provide Python-based data loaders, baseline implementations, and evaluation tools. Here, we give an overview of the datasets, evaluation tools, and provide baseline experiments.

1. Introduction

Graph-structured data is ubiquitous across application domains ranging from chemo- and bioinformatics to image and social network analysis. To develop successful machine learning models in these domains, we need techniques that can exploit the rich information inherent in the graph structure, as well as the feature information contained within nodes and edges. In recent years, numerous approaches have been proposed for machine learning with graphs—most notably, approaches based on graph *kernels* (Kriege et al., 2019) or using *graph neural networks* (GNNs) (Gilmer et al., 2017). However, most papers, even recent ones, evaluate newly proposed architectures or methods on a fixed set of small-scale benchmark datasets leading to high standard deviations and hard to interpret results.

Here, we give an overview of TUD DATASETS. The benchmark collection consists of over 150 datasets from a wide range of domains for supervised learning with graphs, i.e., classificiation and regression. All datasets are provided in a common dataset format at `graphlearning.io`, and easily be accessed from popular graph learning frameworks such as *Pytorch Geometric* (Fey & Lenssen, 2019) and *DGL* (Wang et al., 2019).

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Related work. There exists two approaches to supervised learing with graphs, graph kernels and graph neural networks (GNNs). Graph kernels have been studied extensively in the past 15 years, see (Kriege et al., 2019) for a thorough overview. Important approaches include random-walk and shortest paths based kernels (Gärtner et al., 2003; Sugiyama & Borgwardt, 2015; Borgwardt & Kriegel, 2005; Kriege et al., 2017), as well as the Weisfeiler-Lehman subtree kernel (Shervashidze et al., 2011a; Morris et al., 2017). Further recent works focus on assignment-based approaches (Kriege et al., 2016; Nikolentzos et al., 2017), spectral approaches (Kondor & Pan, 2016), and graph decomposition approaches (Nikolentzos et al., 2018).

Recently, GNNs (Gilmer et al., 2017) emerged as a alternative to graph kernels. Notable instances of this model include (Duvenaud et al., 2015), (Li et al., 2016), (Hamilton et al., 2017) and the spectral approaches proposed in (Bruna et al., 2014; Defferrard et al., 2016; Kipf & Welling, 2017)—all of which descend from early work in (Kireev, 1995; Merkwirth & Lengauer, 2005; Scarselli et al., 2009). A unifying message passing architecture can be found in (Gilmer et al., 2017). Two recent surveys (Wu et al., 2019; Zhou et al., 2018) provide a thorough overview of graph neural networks

The papers (Fey & Lenssen, 2019; Errica et al., 2019; Dwivedi et al., 2020) evalute GNNs using a unified evaluation procedure, however, both only use small scale datasets. Recently, `ogb.stanford.edu` launchend, however the provided datasets for graph classification focus on chemistry applications, and the number is quite limited at this point.

Contributions We give an overview of TUD DATASET, its unified evalution procudures, and baseline methods. Moreover, we report results on a experimental study comparing graph kernels and GNNs on a subset of the TUD DATASET.

2. Overview of the datasets

The TUD DATASET contain over 150 datasets provided at `graphlearning.io`. The data loader, baseline methods, experimental evaluation tool can be installed by running `pip install tud-datasets`, see XXX for further details.

2.1. Baselines

In order to provide meaningful baselines, we provide implementation of common graph kernels was well as GNNs baselines. We have implemented the *Weisfeiler-Lehman Subtree* (Shervashidze et al., 2011b), *Shortest-path* (Borgwardt & Kriegel, 2005), *Graphket* (Shervashidze et al., 2009), *Weisfeiler-Lehman Optimal Assigment* (Kriege et al., 2016) kernel, as well as the higher-order WL kernels (Morris & Mutzel, 2019), in C++ and made them accessable through the Python interface of TUD DATASET, see XXX for further details. Moreover, all GNN architectures implemented in PyTorch Geometric can be used as baseline as well.

Vertex kernel and edge kernel?

2.2. Evaluation tools

In order to insure a fair and meaningful comparision between methods, we propose the following evaluation proceudres, and software tools for kernels and GNN approaches. All proposed methods can be conviently accessed from the Python interface, see XXX for further details. First, for kernels we propose the established C-SVM implementation LIBSVM (Chang & Lin, 2011) for kernels that compute a Gram matrix, and the linear C-SVM implementation LIBLINEAR (Fan et al., 2008) for kernels that can be computed based on explicit feature maps. We optimize GNNs end-to-end using ADAM (). Too compute accuracies and other metrics for evaluation, we propose to use 10-fold cross validation, where we select a validation set uniformly at random from each training fold (10% of the training fold), to select hyperparamets, e.g., number of iterations, C parameter, number of layers, learning rate, feature dimension, etc.

Repeat ten times?

3. Experimental evaluation

Our intention here is to provide baseline experiments, and compare graph kernels and GNNs. More precisely, we address the following questions:

Q1 Are GNNs superior to graph kernels? Is there a single method that dominates?
Q2 ?
Q3 ?

3.1. Experimental protocol

We used the following datasets, graph kernels, and GNN baselines.

Datasets. We used the DEEZER_EGO_NETS, GITHUB_STARGAZERS, ENYMES, IMDB-BINARY, IMDB-MULTI, MCF-7, MOLT-4, NCI1, PROTEINS, REDDIT-BINARY, REDDIT_THREADS, TWITCH_EGOS, UACC257. See the appendix for dataset statistics.

Graph kernels. As kernel baselines we used the *Weisfeiler-Lehman Subtree* (Shervashidze et al., 2011b), *Shortest-path* (Borgwardt & Kriegel, 2005), *Graphket* (Shervashidze et al., 2009), *Weisfeiler-Lehman Optimal Assigment* WL-OA (Kriege et al., 2016), and δ-2-LWL$^+$ kernel (Morris & Mutzel, 2019) included in the TUD DATASET package. The C-parameter was selected from $\{10^{-3}, 10^{-2}, \dots, 10^{2}, 10^{3}\}$ from the validation set. For the larger datasets, we computed sparse feature vectors for each graph and used the linear C-SVM implementation of Liblinear (Fan et al., 2008). The number of iterations of the 1-WL, WL-OA, and the δ-2-LWL$^+$ were selected from $\{0, \dots, 5\}$.[1]

GNNs. We used the GNN architectures GCN (Kipf & Welling, 2017), SAGE (Hamilton et al., 2017), GIN and GIN-ε (Xu et al., 2019) as neural baselines. For all experiments, we used the global mean operator to obtain graph-level outputs. For each dataset, we optimize the number of hidden units from $\{16, 32, 64, 128\}$, the number of layers from $\{2, 3, 4, 5\}$ and the learning rate from $\{0.0001, 0.001, 0.01\}$ with respect to the validation set. The batch size was fixed to 128.

For both methods, we used the evaluation procecude described in Section 2.2 to optimize hyperparameters and compute accuracies. All experiments can be fulled reprodcued from the scripts provided at XXX.

3.2. Results and discussion

See Table 1.

4. Conclusion

We gave an overview of the TUD DATASET, and reported on the results of an experimental study comparing graph kernels and GNNs on a subset of the data. We believe that our dataset collection will spark further progress in the area of graph represention learning. We are looking forward to adding more dataset to our collection, and are excited about contributions from the community, and reseachers and practitionors from other areas.

[1] As already shown in (Shervashidze et al., 2011a), choosing the number of iterations too large will lead to overfitting.

Table 1. Classification accuracies in percent and standard deviations, OOT— Computation did not finish within one day, OOM— Out of memory.

	Graph Kernel	Dataset					
		DATA0	DATA1	DATA1	DATA1	DATA1	DATA1
Kernel	1-WL	±	±	±	±	±	±
	1-WL	±	±	±	±	±	±
	1-WL	±	±	±	±	±	±
	1-WL	±	±	±	±	±	±
GNN	1-WL	±	±	±	±	±	±
	1-WL	±	±	±	±	±	±
	1-WL	±	±	±	±	±	±
	1-WL	±	±	±	±	±	±

5. Acknowledgement

We thank everybody who provided datasets for TUD DATASET.

References

Borgwardt, K. M. and Kriegel, H.-P. Shortest-path kernels on graphs. In *IEEE International Conference on Data Mining*, pp. 74–81, 2005.

Bruna, J., Zaremba, W., Szlam, A., and LeCun, Y. Spectral networks and deep locally connected networks on graphs. In *International Conference on Learning Representation*, 2014.

Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. ACM.

Defferrard, M., X., B., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, pp. 3844–3852, 2016.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems*, pp. 2224–2232, 2015.

Dwivedi, V. P., Joshi, C. K., Laurent, T., Bengio, Y., and Bresson, X. Benchmarking graph neural networks. *CoRR*, abs/2003.00982, 2020.

Errica, F., Podda, M., Bacciu, D., and Micheli, A. A fair comparison of graph neural networks for graph classification. *CoRR*, abs/1912.09893, 2019.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Fey, M. and Lenssen, J. E. Fast graph representation learning with pytorch geometric. *CoRR*, abs/1903.02428, 2019.

Gärtner, T., Flach, P., and Wrobel, S. On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*, volume 2777 of *Lecture Notes in Computer Science*, pp. 129–143. 2003.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, 2017.

Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pp. 1025–1035, 2017.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Kireev, D. B. Chemnet: A novel neural network based method for graph/property mapping. *Journal of Chemical Information and Computer Sciences*, 35(2):175–180, 1995. ACS.

Kondor, R. and Pan, H. The multiscale laplacian graph kernel. In *Advances in Neural Information Processing Systems*, pp. 2982–2990. NIPS, 2016.

Kriege, N. M., Giscard, P.-L., and Wilson, R. C. On valid optimal assignment kernels and applications to graph classification. In *Advances in Neural Information Processing Systems*, pp. 1615–1623. NIPS, 2016.

Kriege, N. M., Neumann, M., Morris, C., Kersting, K., and Mutzel, P. A unifying view of explicit and implicit feature maps for structured data: Systematic studies of graph kernels. *CoRR*, abs/1703.00676, 2017. Accepted for publication in *Data Mining and Knowledge Discovery*.

Kriege, N. M., Johansson, F. D., and Morris, C. A survey on graph kernels. *CoRR*, abs/1903.11835, 2019. Accepted for publication in *Applied Network Science*.

Li, W., Saidi, H., Sanchez, H., Schäf, M., and Schweitzer, P. Detecting similar programs via the Weisfeiler-Leman graph kernel. In *International Conference on Software Reuse*, pp. 315–330, 2016.

Merkwirth, C. and Lengauer, T. Automatic generation of complementary descriptors with molecular graph networks. *Journal of Chemical Information and Modeling*, 45(5):1159–1168, 2005.

Morris, C. and Mutzel, P. Towards a practical k-dimensional weisfeiler-leman algorithm. *CoRR*, abs/1904.01543, 2019.

Morris, C., Kersting, K., and Mutzel, P. Glocalized Weisfeiler-Lehman kernels: Global-local feature maps of graphs. In *IEEE International Conference on Data Mining*, pp. 327–336, 2017.

Nikolentzos, G., Meladianos, P., and Vazirgiannis, M. Matching node embeddings for graph similarity. In *AAAI Conference on Artificial Intelligence*, pp. 2429–2435, 2017.

Nikolentzos, G., Meladianos, P., Limnios, S., and Vazirgiannis, M. A degeneracy framework for graph similarity. In *International Joint Conference on Artificial Intelligence*, pp. 2595–2601, 2018.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009.

Shervashidze, N., Vishwanathan, S. V. N., Petri, T. H., Mehlhorn, K., and Borgwardt, K. M. Efficient graphlet kernels for large graph comparison. In *International Conference on Artificial Intelligence and Statistics*, pp. 488–495. PMLR, 2009.

Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. Weisfeiler-Lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, 2011a.

Shervashidze, N., Schweitzer, P., van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. Weisfeiler-Lehman graph kernels. *PMLR*, 12:2539–2561, 2011b.

Sugiyama, M. and Borgwardt, K. M. Halting in random walk kernels. In *Advances in Neural Information Processing Systems*, pp. 1639–1647, 2015.

Wang, M., Yu, L., Zheng, D., Gan, Q., Gai, Y., Ye, Z., Li, M., Zhou, J., Huang, Q., Ma, C., Huang, Z., Guo, Q., Zhang, H., Lin, H., Zhao, J., Li, J., Smola, A. J., and Zhang, Z. Deep graph library: Towards efficient and scalable deep learning on graphs. *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. URL `https://arxiv.org/abs/1909.01315`.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *International Conference on Learning Representations*, 2019.

Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.