Department of Mathematics and Informatics, Faculty of Sciences,
University of Novi Sad

Conference paper

Evaluation of LID-Aware Graph Embedding Methods for Node Clustering

Dušica Knežević, Jela Babić

Similarity Search and Applications - 15th International Conference, SISAP 2022, Bologna, Italy, October 5-7, 2022, Proceedings

Abstract. In this paper, we present a comparative evaluation of graph embeddings when used to cluster graph nodes in the embedded space. The examined graph embedding methods are node2vec and two recently proposed extensions of this algorithm based on local intrinsic dimensionality. The results of both intrinsic and external clustering evaluation on real-world graphs indicate that LID-aware extensions improve node clustering, especially when detecting a small number of clusters.

All authors contributed equally to the production of this paper.

supervised by
Miloš Savić & Miloš Radovanović

Evaluation of LID-Aware Graph Embedding Methods for Node Clustering

Dušica Knežević[0000-0002-4854-6577], Jela Babić[0000-0001-5244-3725], Miloš Savić[0000-0003-1267-5411], Miloš Radovanović[0000-0003-2225-7803]

Department of Mathematics and Informatics, Faculty of Sciences, University of Novi Sad
Trg Dositeja Obradovića 4, 21000 Novi Sad, Serbia
{lucy, jelab, svc, radacha}@dmi.uns.ac.rs

Abstract. Data generated by everyday applications may appear in different forms. Various important and frequently used machine learning and data mining techniques have been designed assuming the tabular data form. To apply those techniques to graph structured data, it necessary to form graph embeddings. The crucial moment in creating a graph embedding is to choose the best embedding technique that preserves all the vital information when converting a graph into its tabular representation. Determining the best approach requires some form of evaluation of the internal qualities of potential embeddings and their utility in concrete applications. In this paper, we present a comparative evaluation of graph embeddings when used to cluster graph nodes in the embedded space. The examined graph embedding methods are node2vec and two recently proposed extensions of this algorithm based on local intrinsic dimensionality. The results of both intrinsic and external clustering evaluation on real-world graphs indicate that LID-aware extensions improve node clustering, especially when detecting a small number of clusters.

Keywords: Clustering · Graph embedding · LID-aware node2vec

1 Introduction

In the era of big data large amounts of information are produced that require analysis and data mining (DM). Information from sources such as social media and sensors from IoT devices can be used to identify patterns, trends, and new insights that can help with evidence-based decision making and strategic planning. This data can also be used to train predictive models with machine learning (ML) algorithms. There is a variety of ways to organize and store large-scale data: tables, graphs, time series, etc. Numerous data processing techniques developed in the last decades in the research communities of ML and DM usually require input data in tabular form. In the case of graph-structured data, this requirement necessitates transformation of the original data into a suitable format.

The main task of graph embedding methods is to transform graph data into tabular data without losing essential information, but also without explicitly specifying which two nodes are connected by an edge in the produced embedding. It is expected that the application of a graph embedding method before applying ML or DM algorithms produces equal or better results than algorithms natively designed for graphs. The process of graph analysis is then conditioned by the quality of the produced embedding. It should be emphasized that this approach may also face potential problems characteristic for tabular data forms such as the curse of dimensionality.

Tabular data is represented by rows and columns. Columns are features used to describe objects of interest that correspond to rows in the table. The number of columns is the dimensionality of the space in which data is located. When creating the tabular representation in the form of a graph embedding, one of the main problems is to select the suitable value for dimensionality. The measure used to represent the minimal number of features required to describe a point of information in the space is called intrinsic dimensionality (ID). Intrinsic dimensionality is usually viewed as a concept related to the entire dataset, and corresponding ID measures are typically used in dimensionality reduction algorithms. The use of such global ID measures is not always suitable because data dimensionality may vary locally for different parts of a dataset. Local intrinsic dimensionality (LID) was introduced by shifting the focus of ID estimation from the global data view to the data space around a data point [6, 1]. A recent paper by Savić et al. [16] addresses the issue of using LID-related measures in the context of graph embedding generation. The authors proposed NC-LID, a LID-related measure for quantifying the discriminative power of the shortest path distance with respect to natural communities of nodes as their intrinsic localities. Based on this measure, new extensions of the node2vec [4] graph embedding algorithm have been introduced and evaluated by examining to what extent the produced embeddings preserve the structure of input graphs. This paper expands the evaluation given in [16], with the main motivation to analyze the proposed node2vec extensions for the particular application to node clustering.

Clustering is the task of dividing data points into a number of groups (clusters) where similar data points are in the same cluster, and more distant points reside in separate clusters. In terms of graph analysis we use nodes as data points. In other words, clusters in a graph are groups of similar nodes (similar by the shortest-path distance or some other node similarity metric). By performing clustering validation over identified clusters, it is possible to compare the quality of the used embedding methods. The clustering method used in this paper is KMeans [11] which is the most commonly used algorithms to cluster tabular data. KMeans belongs to the group of partitional clustering algorithms in the sense that it generates a single partition with a specified number of non-overlapping clusters. Clustering validation can be intrinsic and external. Intrinsic evaluation focuses on the cohesion of identified clusters based on a distance measure. In our analysis the Silhouette score [15] is used to measure cluster cohesiveness. In external evaluation, identified clusters are compared to explicitly stated node groups determined by labels assigned to nodes. For this purpose we use Normalized Mutual Information [9].

The rest of the paper is structured as follows. A brief summary of analyzed graph embedding methods is given in Section 2. In the following Section 3 we describe methods and measures used to evaluate graph embeddings in the context of node clustering. The obtained results are presented in Section 4. The last section concludes the paper and gives directions for future work.

2 Graph Embedding Methods

The embedding algorithms we analyze belong to the class of random walk algorithms. The random walk approach has become a common technique for the graph embedding problem in recent studies [3]. The main idea of random walk based graph embedding algorithms is to sample a certain number of random walks emanating from each node in a graph. In the case of the node2vec algorithm, the random walks are treated as ordinary sets over the alphabet of node identifiers. This means that the problem of generating graph embeddings is essentially reduced to the problem of generating text embeddings.

The node2vec algorithm is an improvement of DeepWalk [14] with the main idea to use biased random walks. The algorithm is based on finding the best neighborhood selection strategy that allows seamless interpolation between depth first search (DFS) and breadth first search (BFS) when creating a random walk. Node2vec depends on two parameters: p and q. These parameters control how fast the algorithm explores and exits the neighborhood of a given starting node. The p parameter (return parameter) controls the probability of returning to the previous node in the walk. The q parameter (in-out parameter) controls how closely the walk resembles the DFS or BFS exploration strategy. The generated walks are then passed to the Word2Vec algorithm [12], which creates the desired graph embedding.

The aforementioned paper by Savić et al. [16] has raised the question of a different direction to graph embedding generation in which hyperparameters controlling random walk sampling are not globally fixed, but personalized per node and edge with automatic adjustments from initially stated base values. Inspired by the LID model introduced by Houle [6–8], the authors first defined the NC-LID measure for quantifying the discriminatory power of the shortest-path distance considering natural communities [10] of nodes as their intrinsic localities. The lowest possible value of NC-LID is equal to zero and corresponds to the case in which the shortest-path distance perfectly separates the natural community of a node from the rest of the graph. Higher NC-LID values imply more structurally complex natural communities. The authors then proposed two LID-aware node2vec variants (lid-n2v-rw and lid-n2v-rwpq) in which personalized hyperparameters are adjusted according to NC-LID values.

Lid-n2v-rw is the first LID-aware node2vec modification that is focused on the personalization of the number of random walks sampled per node, and the length of random walks. The number of random walks for an arbitrary node n is computed by the equation $\lfloor(1 + \text{NC-LID}(n)) \cdot B\rfloor$, where B is the base number of random walks (by default $B = 10$). Additionally, the length of random walks sampled for n is equal to $\lfloor W/(1 + \text{NC-LID}(n)) \rfloor$ (by default $W = 80$). This means that more random walks are sampled for nodes with more complex natural communities. Additionally, random walks for such nodes are shorter in order to keep the computational budget approximately constant and to lower the probability of "escaping" from their natural communities.

Lid-n2v-rwpq is the second variant of LID-aware node2vec and it considers a more biased approach to graph embedding construction. The main idea is to personalize p and q parameters controlling biases during the random walk sampling. The base values of p and q are $p_b = 1$, and $q_b = 1$, by default. The lid-n2v-rwpq variant incorporates the following adjustments of p and q for a pair of nodes x and y, denoted by $p(x, y)$ and $q(x, y)$, respectively, where x is the node on which the random walk currently resides and y is one of its neighbors:

1. If x is in the natural community of y then $p(x, y) = p_b$, otherwise $p(x, y) = p_b + \text{NC-LID}(y)$. This adjustment lowers the probability of transitioning between different natural communities.
2. If y is in the natural community of x then $q(x, y) = q_b$, otherwise $q(x, y) = q_b + \text{NC-LID}(x)$. This rule increases the probability of staying within more complex natural communities.

3 Evaluation Methods

The evaluation of clustering algorithms applied to graph embeddings can be done in two ways depending on whether nodes have explicit labels indicating cluster assignments (external evaluation) or not (intrinsic evaluation). Let O denote the partitioning of nodes into clusters according to explicit labels and let C be the partitioning of nodes obtained by a clustering algorithm applied to a graph. Then, the similarity between O and C can be obtained by computing partitioning similarity metrics such as normalized mutual information (NMI). Additionally, clustering algorithms can be intrinsically evaluated without having explicit labels by computing metrics reflecting cohesiveness of obtained clusters, such as the Silhouette score.

The Silhouette score indicates how similar data point d (node in our case) is to its own cluster c compared to other identified clusters. It calculates the average distance of d to data points in c and the average distance of d to data points in all other clusters. Its values range from -1 to 1. In our evaluation, the Silhouette score is used as the measure of intrinsic clustering validation being one of the most commonly used indices for assessing clustering quality. Higher Silhouette scores indicate better clustering quality suggesting a more adequate graph embedding for the purpose of clustering, which enables us to examine the impact of graph embedding algorithms in this particular task.

NMI is the normalized variant of the mutual information measure. It is calculated for two partitions of data points into clusters and its value varies from 0, meaning that sets have no mutual information, to 1 which denotes perfect correlation between sets. The value of NMI does not depend on the label naming scheme. Clustering partitions we compare by computing NMI are the following:

- the partition induced by explicit labels present in graph data,
- partitions obtained by community detection on original graphs, and
- partitions identified by KMeans clustering of graph embeddings.

The KMeans algorithm requires parameter K which is the number of non-overlapping clusters that will be detected in the given dataset. In particular, we use the number of detected communities and the number of explicitly stated labels in the original graph. Additionally, we include fixed values for K from $\{2,3,4,5,10\}$. In our evaluation we rely on the implementation of KMeans from the scikit-learn library [13]. The greedy modularity clustering algorithm [2] implemented in the NetworkX library [5] is used for detecting communities.

4 Results

This section describes the results of the comparison between clustering on graph embeddings obtained by node2vec and its two LID-aware variants. The experimental corpus of datasets used for the evaluation is the same as in [16] and consists of five citation networks (Cora, Cora ML, Citeseer, DBLP and Pubmed), two Amazon datasets (Amazon electronics computers and Amazon electronics photo), and one small social network (Zachary karate club). All datasets from the corpus are explicitly labeled.

The results of our analysis indicate that the best results, in terms of NMI and Silhouette scores, are obtained when embeddings are generated for dimensionality equal to 10. Other parameters can be seen in Table 4. These node2vec parameters are determined by reconstructing graphs from embeddings according to Euclidean distance and comparing reconstructed graphs to original ones. It should be emphasized that we use the same values of hyperparameters for our LID-elastic node2vec extensions (base values of p and q as in Table 4) in order to have an unbiased comparison to node2vec. The best value of K in KMeans varies between datasets. In case of Zachary karate club the best K is 2, for Cora ML $K = 7$, Citeseer $K = 6$, Amazon electronics photo $K = 8$, Amazon electronics computers $K = 10$, DBLP $K = 4$, Pubmed $K = 3$, and Cora $K = 70$.

4.1 Intrinsic Evaluation

Figure 1 shows Silhouette scores for node2vec, lid-n2v-rw and lid-n2v-rwpq for KMeans when K values ($K \leq 10$) are chosen such that the Silhouette score is maximal possible. It can be seen that the highest Silhouette score for the majority of datasets is obtained for embeddings generated by lid-n2v-rw. The largest difference in the Silhouette score can be observed for Cora ML where LID-aware node2vec variants have significantly higher values compared to node2vec. Lid-n2v-rwpq is the best performing embedding algorithm for Zachary karate club and Pubmed. The original node2vec is the best option only for one dataset (Cora). Thus, it can be concluded that clusters obtained from LID-aware variants of node2vec are more cohesive suggesting that LID-aware extensions improve the node clustering process.

Table 1: The best values of embedding parameters p and q for node2vec and its LID-aware variants.

DATASET	node2vec		lid-n2v-rw		lid-n2v-rwpq	
	p	q	p	q	p	q
Zachary karate club	0.25	2	0.25	2	0.25	2
Cora	4	0.25	4	0.25	4	0.25
Cora ML	4	0.25	4	0.25	4	0.25
Citeseer	0.5	0.25	0.5	0.25	0.5	0.25
Amazon electronics computers	4	0.5	4	0.5	4	0.5
Amazon electronics photo	4	0.5	4	0.5	4	0.5
DBLP	4	1	4	1	4	1
Pubmed	2	0.5	2	0.5	2	0.5

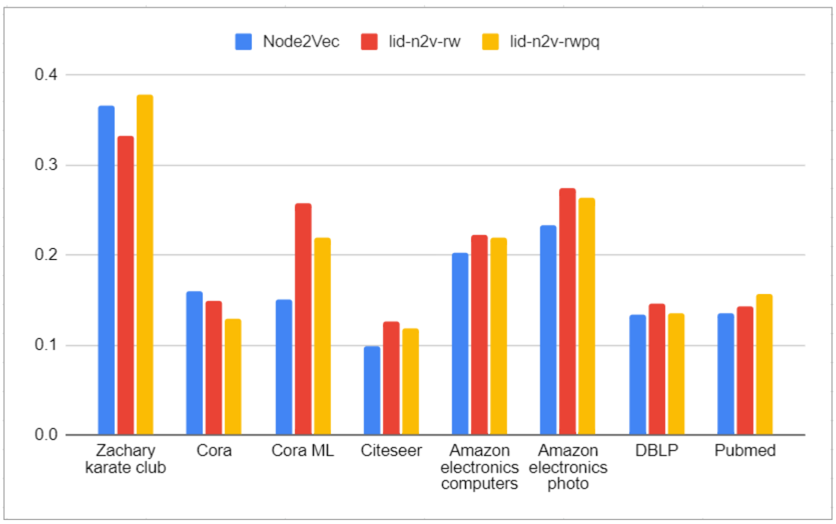


Fig. 1: The best Silhouette scores for KMeans clustering when $K \leq 10$.

Values of Silhouette scores for KMeans clustering when K is equal to the number of communities detected by the greedy modularity optimization (GMO) are shown in Figure 2. For 6 datasets from our experimental corpus, Silhouette scores for different graph embedding methods are very similar. Only for 2 datasets (Zachary and Citesser) we have that node2vec performs slightly better than its LID-aware variants. The number of detected communities is larger than 100 for all datasets except Zacahary where GMO detected 3 clusters. With a larger number of smaller clusters, Silhouette scores tend to have higher values. Consequently, it can be concluded that the intrinsic evaluation for the best K that is lower than or equal to 10 is more reliable than the same evaluation when

$K > 100$ for to two reasons: (1) it is hard to expect an extremely large number of clusters in our datasets considering their size, and (2) for large K the obtained Silhouette scores are almost equal indicating that the clustering results do not depend on the used graph embedding method.

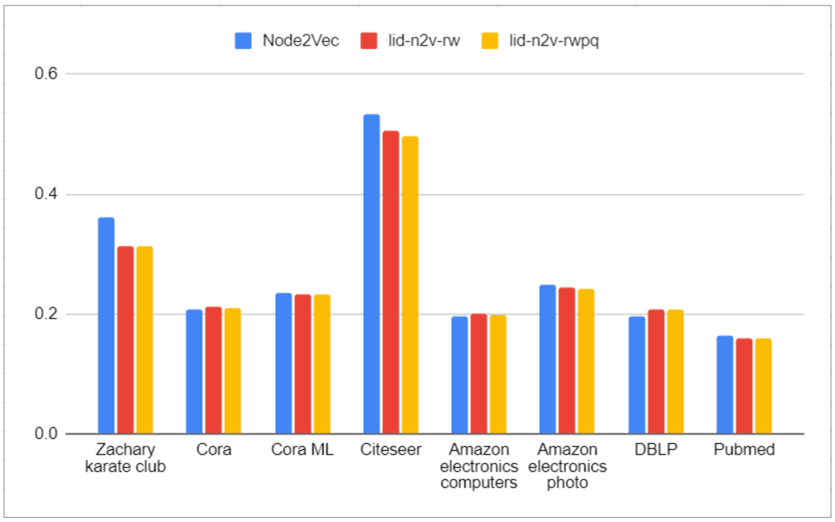


Fig. 2: Silhouette score for KMeans clustering when K is equal to the number of detected communities.

4.2 External Evaluation

In the external evaluation we compute NMI scores between explicitly stated labels in datasets and labelling assignments resulting from KMeans for different K values. The obtained results are summarized in Tables 2 and 3. Table 2 shows the best NMI scores when $K \leq 10$. It can observed that on only two datasets (DBLP and Cora) node2vec has higher NMI scores than its LID-aware variants. For other datasets one of the LID-aware variants is the best performing algorithm. Considerable improvements in the NMI score are present for 3 graphs (Pubmed, Citeseer and Zachary) where NMI of LID-aware variants is higher by 0.1 than NMI of node2vec.

NMI scores when K is equal to the number of clusters detected by GMO are given in Table 3. The largest NMI for Amazon electronics photo is achieved by two algorithms, node2vec and lid-n2v-rw. However, on the same dataset lid-n2v-rwpq has slightly lower NMI indicating that all three algorithms perform similarly. The situation is similar for other datasets where NMI of the best performing algorithm is not considerably higher than NMI of two other alternatives. For 2 datasets (DBLP, Pubmed) node2vec achieves the highest NMI, lid-n2v-rw

Table 2: NMI scores for explicit labels and labeling assignments obtained from KMeans for the best $K \leq 10$.

DATASET	node2vec	lid-n2v-rw	lid-n2v-rwpq
Zachary karate club	0.693	0.826	0.727
Cora	0.545	0.523	0.418
Cora ML	0.548	0.583	0.597
Citeseer	0.489	0.577	0.572
Amazon electronics computers	0.554	0.554	0.569
Amazon electronics photo	0.649	0.675	0.657
DBLP	0.557	0.478	0.471
Pubmed	0.368	0.479	0.483

is also the best option for 2 datasets (Cora and Amazon electronics computers), whereas lid-n2v-rwpq reaches the highest NMI for 3 datasets. The best NMI score per dataset considering both cases (when $K \leq 10$ and K equal to the number of detected clusters) is indicated in Table 4, where it can be seen that for 6 out of 8 datasets LID-elastic node2vec extensions outperform the original node2vec.

Table 3: NMI scores for explicit labels and labeling assignments from KMeans when K is equal to the number of detected communities.

DATASET	node2vec	lid-n2v-rw	lid-n2v-rwpq
Zachary karate club	0.727	0.826	0.861
Cora	0.546	0.548	0.545
Cora ML	0.640	0.639	0.651
Citeseer	0.857	0.855	0.858
Amazon electronics computers	0.403	0.404	0.401
Amazon electronics photo	0.489	0.489	0.486
DBLP	0.574	0.557	0.557
Pubmed	0.574	0.529	0.523

We also examined the correlation between Silhouette and NMI scores for LID-aware node2vec variants. Figure 3 shows the highest Silhouette and NMI across all datasets from our experimental corpus. It can be seen that those two metrics are perfectly correlated, i.e. larger Silhouette score implies larger NMI. Consequently, it can be concluded that the metric we selected for intrinsic evaluation is consistent with the metric used for external evaluation. This is also evident in the results of the evaluation itself:

Table 4: The best NMI score per dataset.

DATASET	Best NMI score	Method
Zachary karate club	0.861	lid-n2v-rwpq
Cora	0.548	lid-n2v-rw
Cora ML	0.651	lid-n2v-rwpq
Citeseer	0.858	lid-n2v-rwpq
Amazon electronics computers	0.569	lid-n2v-rwpq
Amazon electronics photo	0.675	lid-n2v-rw
DBLP	0.574	node2vec
Pubmed	0.574	node2vec

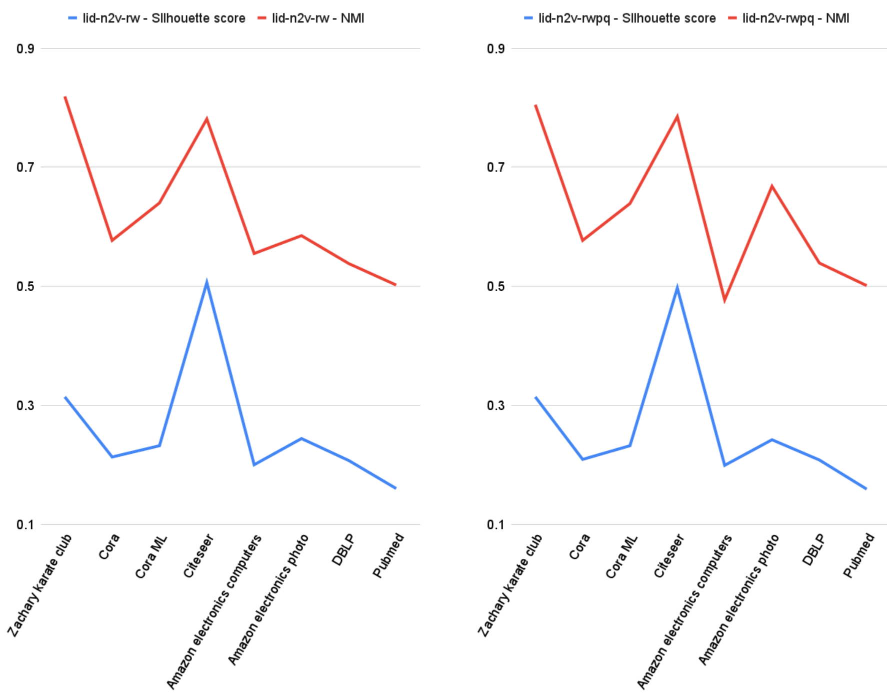
1. for a small number of detected clusters ($K \leq 10$) LID-aware node2vec variants perform better than pure node2vec according to both intrinsic and external evaluation, while
2. for a large number of detected clusters corresponding to the number of communities detected by GMO all three algorithms achieve similar performance.

5 Conclusion and Future Work

The main focus of this paper was on the evaluation of LID-aware graph embedding methods when they are used prior to node clustering. Both intrinsic and external evaluation of KMeans were conducted on embeddings produced by pure node2vec and two its LID-aware extensions that personalize hyperparameters controlling random walk sampling. For the purpose of intrinsic evaluation we selected the Silhouette score to quantify cohesiveness of clusters. The external evaluation was based on NMI considering labels explicitly given in datasets and label assignments obtained by KMeans for different K values.

The obtained results indicate that LID-aware node2vec extensions in general achieve better intrinsic and external evaluation scores, especially when detecting a small number of clusters (equal to or less than 10). In the case of a large number of clusters, corresponding to the number of communities detected by greedy modularity optimization applied directly to graphs, all three examined graph embedding algorithms achieve comparable evaluation scores. The intrinsic evaluation should be considered more reliable since external evaluation depends on explicitly assigned labels that do not necessarily represent natural clusters. However, it was shown that the selection of metrics in different evaluation methods gives consistent results.

Regarding future work, our evaluation could be expanded by including additional datasets, community detection algorithms and other clustering algorithms designed for tabular data (e.g., agglomerative hierarchical clustering, DBSCAN, etc.). LID-aware extensions evaluated in this paper are based on local community detection, but in addition we could also detect communities globally. For



(a) lid-n2v-rw (b) lid-n2v-rwpq

Fig. 3: Comparison of maximal Silhouette and NMI scores across all datasets.

example, an additional parameter controlling the probability of a random walk leaving a global community could be incorporated into LID-aware extensions of node2vec. In this way we force random walks to stay within the global community of a starting node. Furthermore, this idea might as well be expanded to overlapping communities where special attention is given to nodes belonging to multiple communities.

Acknowledgments. This research is supported by the Science Fund of the Republic of Serbia, #6518241, AI – GRASP.

References

1. Amsaleg, L., Chelly, O., Furon, T., Girard, S., Houle, M.E., Kawarabayashi, K.i., Nett, M.: Estimating local intrinsic dimensionality. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 29–38. KDD'15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2783258.2783405

2. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Physical Review E **70**(6) (2004). https://doi.org/10.1103/physreve.70.066111
3. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems **151**, 78–94 (2018). https://doi.org/10.1016/j.knosys.2018.03.022
4. Grover, A., Leskovec, J.: Node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 855–864. KDD'16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939754
5. Hagberg, A.A., Schult, D.A., Swart, P.J.: Exploring network structure, dynamics, and function using networkx. In: Varoquaux, G., Vaught, T., Millman, J. (eds.) Proceedings of the 7th Python in Science Conference. pp. 11 – 15. Pasadena, CA USA (2008)
6. Houle, M.E.: Dimensionality, discriminability, density and distance distributions. In: 2013 IEEE 13th International Conference on Data Mining Workshops. pp. 468–473 (2013). https://doi.org/10.1109/ICDMW.2013.139
7. Houle, M.E.: Local intrinsic dimensionality i: An extreme-value-theoretic foundation for similarity applications. In: Beecks, C., Borutta, F., Kröger, P., Seidl, T. (eds.) Similarity Search and Applications. pp. 64–79. Springer International Publishing, Cham (2017)
8. Houle, M.E.: Local intrinsic dimensionality III: Density and similarity. In: Satoh, S.e.a. (ed.) Similarity Search and Applications. pp. 248–260. Springer International Publishing, Cham (2020)
9. Kvalseth, T.O.: On normalized mutual information: Measure derivations and properties. Entropy **19**(11) (2017). https://doi.org/10.3390/e19110631
10. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. New Journal of Physics **11**(3), 033015 (mar 2009). https://doi.org/10.1088/1367-2630/11/3/033015
11. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Cam, L.M.L., Neyman, J. (eds.) Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability. vol. 1, pp. 281–297. University of California Press (1967)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings (2013), http://arxiv.org/abs/1301.3781
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
14. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 701–710. KDD'14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2623330.2623732
15. Rousseeuw, P.J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics **20**, 53–65 (1987). https://doi.org/https://doi.org/10.1016/0377-0427(87)90125-7

16. Savić, M., Kurbalija, V., Radovanović, M.: Local intrinsic dimensionality and graphs: Towards lid-aware graph embedding algorithms. In: Reyes, N., Connor, R., Kriege, N., Kazempour, D., Bartolini, I., Schubert, E., Chen, J.J. (eds.) Similarity Search and Applications. pp. 159–172. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-89657-7"̇13