Local Intrinsic Dimensionality and Graphs: Towards LID-aware Graph Embedding Algorithms

Miloš Savić, Vladimir Kurbalija, and Miloš Radovanović(✉)

Department of Mathematics and Informatics, Faculty of Sciences,
University of Novi Sad, Trg Dositeja Obradovića 3, 21000 Novi Sad, Serbia
{svc,kurba,radacha}@dmi.uns.ac.rs

Abstract. Local intrinsic dimensionality (LID) has many important applications in the field of machine learning (ML) and data mining (DM). Existing LID models and estimators have mostly been applied to data points in Euclidean spaces, enabling LID-aware ML/DM algorithms for tabular data. To the best of our knowledge, prior research works have not considered LID for designing or evaluating graph-based ML/DM algorithms. In this paper, we discuss potential applications of LID to graph-structured data considering graph embeddings and graph distances. Then, we propose NC-LID – a LID-related measure for quantifying the discriminatory power of the shortest-path distance with respect to natural communities of nodes as their intrinsic neighborhoods. It is shown how NC-LID can be utilized to design LID-elastic graph embedding algorithms based on random walks by proposing two LID-elastic variants of Node2Vec. Our experimental evaluation on real-world graphs demonstrates that NC-LID can point to weak parts of Node2Vec embeddings that can be improved by the proposed LID-elastic extensions.

Keywords: Intrinsic dimensionality · Graph embeddings · Graph distances · Natural communities · LID-elastic Node2Vec

1 Introduction

The intrinsic dimensionality (ID) of a dataset is the minimal number of features that are needed to form a good lower-dimensional representation of the dataset without a large information loss. The estimation of ID is highly relevant for various machine learning and data mining tasks, especially when dealing with high-dimensional data. Namely, lower-dimensional data representations can be exploited to train machine learning models in order to improve their generalizability by alleviating negative effects of high dimensionality. Due to a smaller number of features, such models are more comprehensible and their training, tuning and validation is more time efficient.

N. Reyes et al. (Eds.): SISAP 2021, LNCS 13058, pp. 159–172, 2021.
https://doi.org/10.1007/978-3-030-89657-7_13

The notion of the local intrinsic dimensionality (LID) has been developed in recent years motivated by the fact that the ID may vary across a dataset. The main idea of LID is to focus the estimation of ID to a data space surrounding a data point. In a seminal paper, Houle [7] defined the LID considering the distribution of distances to a reference data point. Additionally, Houle showed that for continuous distance distributions with differentiable cumulative density functions the LID and the indiscriminability of the corresponding distance function are actually equivalent. Let x be a reference data point and let F denote the cumulative distribution function of distances to x. It can be said that the underlying distance function is discriminative at a given distance r if $F(r)$ has a small increase for a small increase in r. Thus, the indiscriminability of the distance function at r w.r.t x, denoted by $ID(r)$, can be quantified as the limit of the ratio of (a) the proportional rate of increase of $F(r)$, and (b) the proportional rate of increase in r. Then, the LID of x is given as $\lim_{r \to 0} ID(r)$. For practical applications, the LID of x can be estimated considering the distances of x to its k nearest data points [1,2]. Recent research works showed that the LID can be exploited for density-based clustering [9], outlier detection [9,10], training deep neural network classifiers on datasets with noisy labels [13], detection of adversarial data points when training deep neural networks [12], subspace clustering and estimating the local relevance of features [3] and similarity search [4,8].

The applications of machine learning and data mining algorithms designed for tabular datasets to graphs are enabled by various graph embedding algorithms [5]. Here we consider graph embedding algorithms translating graph nodes into n-dimensional real-valued vectors with the goal of preserving graph-based distances in the embedding space. Besides applications in node classification, node clustering and link prediction tasks, graph embeddings may be also utilized for similarity search applications. Namely, similarity search when performed directly on large-scale graphs may pose several difficulties due to the small-world phenomenon [16], i.e. for a given node (similarity search query) the number of nodes at a given shortest-path distance (potential similarity search hits) grows at a very fast rate with the shortest path distance.

In this paper we discuss potential applications of LID to graphs (Sect. 2). To the best of our knowledge, this is the first work considering LID for designing and evaluating ML/DM algorithms operating on graph-structured data. As the main contributions, we propose a LID-related measure called NC-LID to quantify the discriminability of the shortest-path distance locally per node with respect to their natural communities as intrinsic subgraph boundaries (Sect. 3) and two extensions of the Node2Vec graph embedding algorithm [6] that personalize and adjust Node2Vec parameters according to NC-LID values (Sect. 4). In the experimental evaluation presented in Sect. 5, it is demonstrated that NC-LID can indicate weak parts of Node2Vec embeddings prior to their construction and that our LID-elastic Node2Vec extensions provide better embeddings w.r.t. reconstruction errors. In the last section we discuss possible directions for future research.

2 LID and Graphs

Existing LID models and corresponding estimators have been designed for tabular datasets with real-valued features and smooth distance functions. There are two ways in which they can be applied to graphs: (a) by transforming graphs into tabular data representations using graph embedding algorithms, and (b) by using graph-based distances instead of distances of vectors in Euclidean spaces. To the best of our knowledge, we are not aware of any previous research study investigating the LID of graph embeddings or applying LID estimators to graph-based distances.

The first approach enables the LID-based evaluation of graph embeddings and their analysis in the context of distance-based machine learning and data mining algorithms. For example, Amsaleg et al. [1] proposed the maximum-likelihood LID estimator (MLE-LID). By computing MLE-LID for each node in a graph on embeddings produced by different graph embedding algorithms we can study which of the embeddings is the most effective for distance-based machine learning and data mining algorithms (under the assumption that the embeddings preserve the structure of the graph to a similar extent). Additionally, obtained MLE-LID values can indicate whether we can benefit from LID-aware data mining and machine learning algorithms for a concrete embedding.

LID estimates for graph nodes obtained by applying LID estimators on graph embeddings are relative to the selected graph embedding dimension that is explicitly required by graph embedding algorithms. Additionally, the usefulness of LID estimates depends on the ability of the selected graph embedding algorithm to preserve the structure of the input graph.

The MLE-LID estimator mentioned above (or any other LID estimator, e.g. the estimator also proposed by Amsaleg et al. [2] that estimates LID within tight localities) can be applied "directly" on graphs by taking shortest path distances instead of distances in Euclidean spaces (in the most general case since graph embedding algorithms try to preserve shortest path distances in embedded spaces). However, LID estimates based on shortest path distances will suffer from negative effects of the small-world property, i.e. for a randomly selected node n there will be an extremely large fraction of nodes at the same and relatively small shortest-path distance from n. The hubness property of large-scale real-world graphs (i.e., the existence of nodes with an extremely high degree that are called hubs) will also have a big impact on such LID estimates. For example, LID for hubs will be estimated as 0 by the MLE-LID estimator due to a large number of nearest neighbors at the shortest-path distance 1. Another problem with this approach is the shortest-path distance itself. The notion of LID is based on the assumption that the radius of a ball around a data point can be increased by a small value that tends to 0. However, the shortest-path distance does not have an increase that can go to 0 (the minimal increase is 1) in contrast to distances in Euclidean space.

3 NC-LID: LID-related Measure for Graph Nodes Based on Natural Communities

Following the discussion from the previous section, we consider a somewhat different conceptual approach to designing LID-related measures for nodes in a graph. The main idea is to substitute a ball around a data point with a subgraph around a node in order to estimate the discriminatory power of a graph-based distance of interest. Here we observe the most basic case which is a fixed subgraph that can be considered as the intrinsic locality of the node.

Let n denote a node in a graph $G = (V, E)$ and let S be a subgraph containing n. The graph-based distance of interest can be the shortest-path distance, but also any other node similarity function, including hybrid node similarity measures for attributed graphs. Assuming that S is a natural (intrinsic) locality of n, d can be considered as a perfectly discriminative distance measure if it clearly separates nodes in S from the rest of the nodes in G.

To measure the degree of discriminatory power of d considering S as the intrinsic locality of n we define a general limiting form of the local intrinsic discriminability of d as

$$\text{GB-LID}(n) = -\ln\left(\frac{|S|}{T(n,S)}\right), \tag{1}$$

where $|S|$ is the number of nodes in S. $T(n, S)$ is the number of nodes whose distance from n is smaller than or equal to r, where r is the maximal distance between n and any node from S:

$$T(n,S) = \left|\left\{y \in V \;:\; d(n,y) \leq \max_{z \in S} d(n,z)\right\}\right|. \tag{2}$$

Similarly to standard LID for tabular data, GB-LID assesses the local neighborhood size of n at two ranges: the number of nodes in a neighborhood of interest (S) and the total number of nodes that are within relevant distances from n considering distances from n to nodes in S. The more extreme the ratio between these two, the higher the intrinsic dimensionality (local complexity) of n. Unlike standard LID, GB-LID depends on the complexity of a fixed subgraph around the node rather than some measure reflecting the dynamics of expanding subgraphs (this will be part of our future work). Compared to other measures capturing the local complexity of a node (e.g., degree centrality and clustering coefficient), GB-LID is not restricted to ego-networks of nodes or regularly expanding subgraphs capturing all nodes within the given distance (e.g., LID-based intrinsic degree proposed by von Ritter et al. [15]).

GB-LID is a class of LID-related scores effectively parameterized by $\langle S_n, d\rangle$, where S_n is the subgraph denoting the intrinsic local neighborhood of node n and d is an underlying distance measure. From GB-LID we derive one concrete measure called NC-LID (NC is the abbreviation for “Natural Community”). In NC-LID we fix S_n to the natural (local) community of n determined by the fitness-based algorithm for recovering natural communities [11] and d is the shortest path distance.

A community in a graph is a highly cohesive subgraph. This means that the number of links within the community (so-called intra-community links) is significantly higher than the number of links connecting nodes from the community to nodes outside the community (so-called inter-community links). The natural or local community of node n is a community recovered from n. When computing NC-LID we use the fitness-based algorithm for identifying natural communities proposed by Lancichinetti et al. [11]. Starting from n, this algorithm recovers the natural community C of n by maximizing the community fitness function that is defined as:

$$f_C = \frac{k_{in}(C)}{(k_{in}(C) + k_{out}(C))^\alpha}, \tag{3}$$

where $k_{in}(C)$ is the total intra-degree of nodes in C, $k_{out}(C)$ is the total inter-degree of nodes in C, and α is a real-valued parameter controlling the size of C (larger α implies smaller C). The intra-degree and inter-degree of a node s are the number of intra-community and inter-community links incident to s, respectively. The most natural choice for α is $\alpha = 1$, which corresponds to the Raddichi notion of weak communities [14].

NC-LID(n) is equal to 0 if all nodes from the natural community of n are at shorter shortest-path distances to n than nodes outside its natural community. Higher values of NC-LID(n) imply that it is harder to distinguish the natural community of n from the rest of the graph using the shortest-path distance, i.e. the natural community of n tends to be more "concave" and elongated in depth with higher NC-LID(n) values. Nodes with such complex natural communities may also be brokers having large values of node centrality metrics that connect different parts of the graph by their long-range links (i.e., links whose removal significantly increase the average shortest path distance).

4 LID-elastic Node2Vec Variants

Having an appropriate LID-based score for graph nodes such as NC-LID, it is possible to design LID-aware or LID-elastic graph embedding algorithms. In this work we propose two LID-elastic variants of Node2Vec [6].

Node2Vec is a random-walk based algorithm for generating graph embeddings. The main idea of random-walk based graph embedding algorithms is to sample a certain number of random walks starting from each node in a graph. Sampled random walks are then treated as ordinary sentences over the alphabet encompassing node identifiers. This means that the problem of generating graph embeddings is reduced to the problem of generating text embeddings. Node2Vec relies on Word2Vec to produce node embedding vectors from random-walk sentences.

Node2Vec employs a second order random walk scheme with two parameters p and q which guide the walk. Let us assume that a random walk just transitioned from node t to node v. The parameter p (return parameter) controls the probability of intermediately returning back to t. The parameter q (in-out parameter) controls to what extent random walk resembles BFS or DFS graph exploration strategies. For $q > 1$, the random walk is more biased to nodes close to t (BFS-like graph exploration). If $q < 1$ then the random walk is more inclined to visit nodes that are further away from t (DFS-like graph exploration).

Our Node2Vec LID-elastic extensions are based on the premise that high NC-LID nodes have higher link reconstruction errors than low NC-LID nodes due to more complex natural communities. More specifically, the quality of graph embeddings can be assessed by comparing original graphs to graphs reconstructed from embeddings. Let G denote an arbitrary graph with L links and let E be an embedding constructed from G using some graph embedding algorithm. The graph reconstructed from E has the same number of links as G. The links in the reconstructed graph are formed by joining the L closest node vector pairs from E. Then, the following metrics quantifying the quality of E according to the principle that nodes close in G should be also close in E can be computed for each node n:

1. Link precision $P(n)$ is the number of correctly reconstructed links incident to n divided by the total number of links incident to n in the reconstructed graph.
2. Link recall $R(n)$ is the number of correctly reconstructed links incident to n divided by the total number of links incident to n in the original graph.
3. Link F_1 score $F_1(n)$ is a metric aggregating $P(n)$ and $R(n)$ into a single score that is defined as their harmonic mean: $F_1(n) = 2 \cdot P(n) \cdot R(n)/(P(n) + R(n))$.

Higher values of $P(n)$, $R(n)$ and $F_1(n)$ imply lower link reconstruction errors for n.

The sampling mechanism of Node2Vec is controlled by 4 parameters: the number of random walks sampled per node, the length of each random walk, p and q. The first two parameters are fixed for each node in a graph, while p and q are fixed for each pair of nodes. Our Node2Vec LID-elastic extensions are based on Node2Vec parameters personalized for nodes and pairs of nodes that are adjusted according to their NC-LID values.

The first LID-elastic variant of Node2Vec, denoted by `lid-n2v-rw`, personalizes the number of random walks sampled per node and the length of random walks according to the following rules:

1. The number of random walks sampled for n is equal to $\lfloor (1 + \text{NC-LID}(n)) \cdot B \rfloor$, where B is the base number of random walks (by default $B = 10$).
2. The length of random walks sampled for n is equal to $\lfloor W/(1 + \text{NC-LID}(n)) \rfloor$ (by default $W = 80$).

`lid-n2v-rw` samples a proportionally higher number of random walks for high NC-LID nodes while keeping the computational budget (the total number of random walk steps per node) approximately constant. The main idea is to increase the frequency of high NC-LID nodes in sampled random walks in order to better preserve their close neighborhood in formed embeddings. Additionally, the probability of the random walk leaving the natural community of the starting node is lowered for high NC-LID nodes due to shorter random walks.

The second LID-elastic variant of Node2Vec, denoted by `lid-n2v-rwpq`, extends `lid-n2v-rw` by personalizing p and q parameters controlling biases when sampling random walks. Let p_b and q_b denote the base values of p and q (by default $p_b = q_b = 1$). The `lid-n2v-rwpq` variant incorporates the following adjustments of p and q for a pair of nodes x and y, where x is the node on which the random walk currently resides and y is one of its neighbours:

1. If x is in the natural community of y then $p(x, y) = p_b$, otherwise $p(x, y) = p_b + \text{NC-LID}(y)$.
2. If y is in the natural community of x then $q(x, y) = q_b$, otherwise $q(x, y) = q_b + \text{NC-LID}(x)$

The first rule controls the probability of returning back from x to y if the random walk transitioned from y to x in the previous step. By increasing the base p value if x is not in the natural community of y `lid-n2v-rwpq` lowers the probability of making a transition between different natural communities. The increase is equal to NC-LID(y) which implies that the backtrack step is penalized more if y has a more complex natural community.

The second rule controls the probability of going to nodes that are more distant from the previously visited node in the random walk. The base q value is increased for nodes not belonging to the natural community of x meaning that again `lid-n2v-rwpq` penalizes transitioning between different natural communities. The increase in q_b is equal to NC-LID(x) implying that `lid-n2v-rwpq` biases the random walk to stay within more complex natural communities.

5 Experiments and Results

Our experimental evaluation of the NC-LID measure and LID-elastic Node2Vec extensions is performed on datasets (graphs) listed in Table 1. The experimental corpus encompasses three small social networks (Karate club, Les miserables and Florentine families), five paper citation networks (CORAML, CORA, CITESEER, PUBMED and DBLP) and two co-purchasing networks of Amazon products (AE photo and AE computers) that are commonly used to evaluate graph embedding methods. For each graph, Table 1 shows the number of nodes (N), the number of links (L), the number of connected components (C), the fraction of nodes in the largest connected component (F), the average degree ($\bar{d}$) and the skewness of the degree distribution (S). It can be observed that the experimental corpus encompasses both small and large sparse graphs ($\bar{d} \ll N - 1$). All graphs, except CITESEER, are either connected graphs ($C = 1$) or have a giant connected component ($F > 0.9$). The degree distribution of large graphs has a high positive skewness implying that those graphs contain so-called hubs (nodes whose degree is significantly higher than the average degree).

Table 1. Experimental datasets.

Graph	N	L	C	F	$\bar{d}$	S
Karate club	34	78	1	1.00	4.59	2.00
Les miserables	77	254	1	1.00	6.60	1.89
Florentine families	15	20	1	1.00	2.67	0.62
CORAML	2995	8158	61	0.94	5.45	12.28
CITESEER	4230	5337	515	0.40	2.52	8.44
AE photo	7650	119081	136	0.98	31.13	10.42
AE computers	13752	245861	314	0.97	35.76	17.34
PUBMED	19717	44324	1	1.00	4.50	5.21
CORA	19793	63421	364	0.95	6.41	7.87
DBLP	17716	52867	589	0.91	5.97	9.43

5.1 Natural Communities and NC-LID

Since natural communities are the base for the NC-LID measure, we first examine their characteristics. Figure 1 shows the complementary cumulative distribution (CCD) of the size of natural communities on a log-log plot. The size of a natural community is the number of nodes it contains. It can be seen that CCDs for large graphs have very long tails. This implies that a large majority of nodes have relatively small natural communities (10 or less nodes), but there are also nodes having exceptionally large natural communities (100 or more nodes). For example, 76.56% of CORA nodes have natural communities with 4 or less nodes, while the largest natural community in CORA contains 146 nodes.

The average NC-LID and the maximal NC-LID of nodes in examined graphs are presented in Fig. 2 sorted from the graph having the most compact natural communities to the graph with the most complex natural communities on average. The social network of Florentine families has the lowest average NC-LID equal to 0.48. This NC-LID level means that approximately 38% of nodes within the shortest-path radius of the natural community of a randomly selected node do not belong to its natural community. The largest average NC-LID for examined graphs is 5.12 (AE computers). This NC-LID value corresponds to situations in which approximately 0.6% of nodes within the shortest-path radius of a natural community belong to the natural community. It should be emphasized that NC-LID positively correlates with the size of the natural community (Spearman's correlations higher than 0.3) for 5 graphs, for 3 graphs negatively (correlations lower than −0.15), while for 2 graphs (PUBMED and AE Computers) significant correlations are absent.

5.2 Node2Vec Evaluation

Prior to evaluating LID-elastic Node2Vec modifications, we examine characteristics of Node2Vec embeddings. Graph reconstruction metrics (mean link precision,

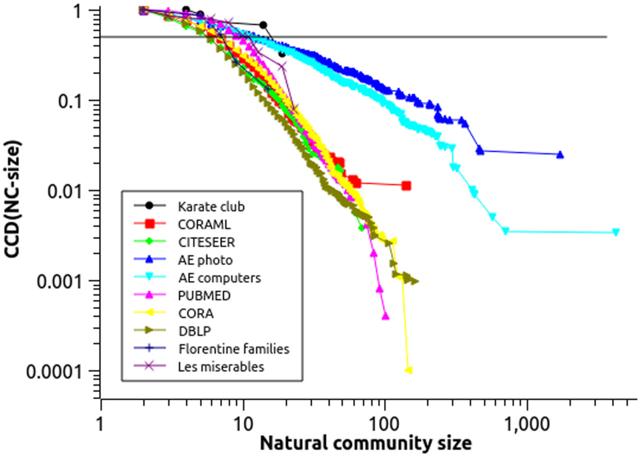

Fig. 1. The complementary cumulative distribution of sizes of natural communities. The solid line represents the 0.5 probability.

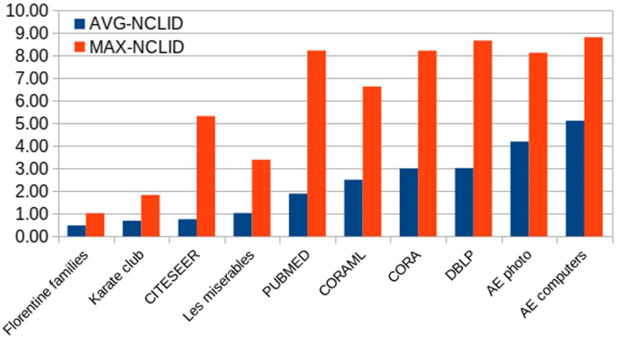

Fig. 2. The average and the maximal NC-LID for graphs from our experimental corpus.

recall and F_1 scores, see Sect. 4) were computed for 125 Node2Vec embeddings per graph in order to find the best embedding in the following parameter space: p and q were varied to take values in $\{0.25, 0.5, 1, 2, 4\}$, and the embedding dimension in $\{10, 25, 50, 100, 200\}$. The number of sampled random walks per node and the length of random walk was set to their default values (10 and 80, respectively) as suggested in [6]. The parameters for the best embeddings, selected according to the average F_1 score, are shown in Table 2 (P denotes the mean link precision and R the mean link recall). It can be seen that for all graphs except CITESEER, Node2Vec preserves the structure of examined graphs to a fairly good extent (F_1 in the range from 0.39 to 0.96).

The basic assumption of LID-elastic Node2Vec modifications is that high NC-LID nodes have higher graph reconstruction errors compared to low NC-LID

Table 2. Characteristics of the best Node2Vec embeddings.

Graph	Dim.	p	q	P	R	F_1
Karate club	100	0.25	4	0.7814	0.7762	0.7788
Les miserables	100	0.25	4	0.7889	0.8325	0.8101
Florentine families	100	0.25	4	0.9667	0.9611	0.9639
CORAML	25	0.5	0.25	0.6300	0.6682	0.6485
CITESEER	10	0.5	0.25	0.2284	0.2438	0.2359
AE photo	50	0.5	0.5	0.5076	0.4835	0.4953
AE computers	50	4	0.25	0.4856	0.4231	0.4522
PUBMED	50	4	0.25	0.3152	0.5245	0.3937
CORA	25	4	0.25	0.5803	0.5648	0.5724
DBLP	25	0.5	1	0.4431	0.3693	0.4029

nodes when applying the original Node2Vec to generate graph embeddings. To check this assumption we first examine Spearman's correlations between NC-LID of nodes and their F_1 scores in the best Node2Vec embeddings described in Table 2. The obtained results are presented in Fig. 3. It can be seen that for all graphs except two small graphs (Karate club and Les miserables) there are notable negative Spearman's correlations between NC-LID and F_1 ranging from -0.2 to -0.4 (please recall that lower F_1 scores imply higher graph reconstruction errors).

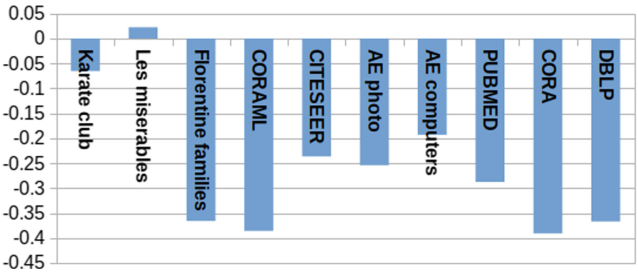

Fig. 3. The Spearman correlation between NC-LID of nodes and their F_1 scores in the best Node2Vec embeddings.

Second, we divide nodes into two groups: H – nodes that have high NC-LID values higher than the average NC-LID and L – nodes with low NC-LID values lower than the average NC-LID. Then, we apply the Mann-Whitney U (MWU) test to those two groups of nodes considering their F_1 scores. The MWU test checks the null hypothesis that scores in one group do not tend to be either higher or lower than scores in the other group. The results of conducted MWU tests are summarized in Table 3. The table shows the average F_1 score for H and

L ($F_1(H)$ and $F_1(L)$, respectively), the value of the MWU test statistic (U), the p-value of U (p) and values of two probabilities of superiority:

- PS(H) – the probability that the F_1 score of a randomly selected node from H (denoted by h) is strictly higher than the F_1 score of a randomly selected node from L (denoted by l), and
- PS(L) – the probability that the F_1 of l is strictly higher than the F_1 of h.

We accept the null hypothesis of MWU (no statistically significant differences between in F_1 scores of H and L) if $p > 0.01$ (column "acc." in Table 3). It can be observed that the null hypothesis of MWU is accepted only for the three smallest graphs from our experimental corpus. For large graphs we have that F_1 scores of high NC-LID nodes tend to be significantly lower than F_1 scores of low NC-LID nodes ($F_1(H) < F_1(L)$ and $\text{PS}(H) \ll \text{PS}(H)$).

Table 3. Comparison of F_1 scores of high NC-LID nodes (H) and low NC-LID nodes (L) using the Mann-Whitney U test.

Graph	$F_1(H)$	$F_1(L)$	U	p	acc.	PS(H)	PS(L)
Karate club	0.70	0.71	132	0.44	yes	0.44	0.48
Les miserables	0.76	0.76	734	0.50	yes	0.47	0.47
Florentine families	0.93	0.98	19	0.10	yes	0.07	0.39
CORAML	0.44	0.62	699380	$<10^{-2}$	no	0.29	0.67
CITESEER	0.10	0.25	1707420	$<10^{-2}$	no	0.19	0.31
AE photo	0.32	0.43	5239408	$<10^{-2}$	no	0.36	0.64
AE computers	0.29	0.38	17900546	$<10^{-2}$	no	0.38	0.61
PUBMED	0.19	0.32	31448278	$<10^{-2}$	no	0.28	0.59
CORA	0.36	0.54	29695497	$<10^{-2}$	no	0.28	0.68
DBLP	0.20	0.42	26684749	$<10^{-2}$	no	0.25	0.57

By taking into account both the observed Spearman's correlations and the results of the MWU tests it can be concluded that high NC-LID nodes tend to have significantly higher graph reconstruction errors than low NC-LID nodes. This implies that the NC-LID measure is able to point to "weak" parts of Node2Vec embeddings prior to their constructions. Consequently, Node2Vec embeddings could be possibly improved by adjusting Node2Vec parameters individually per node according to its NC-LID value.

5.3 LID-elastic Node2Vec Evaluation

Embeddings by LID-elastic Node2Vec variants proposed in Sect. 4 are generated according to the best configurations of original Node2Vec (Table 2). More specifically, for a given graph and embedding dimension we set base p and base q of

LID-elastic Node2Vec variants to p and q of the best corresponding Node2Vec embedding. As for Node2Vec embeddings examined in the previous section, the base number of random walks and the base length of random walks are set to their default values. The embedding dimension is varied in the same way as for Node2Vec. Then, we examine `lid-n2v-rw` and `lid-n2v-rwpq` embeddings by computing their average link F_1 scores, selecting the best embedding across considered embedding dimensions, and comparing the best LID-elastic Node2Vec embedding to the best embedding generated by Node2Vec (`n2v`). The obtained results are summarized in Table 4 showing the best F_1 score of `n2v` and the embedding dimension in which it is achieved and the best F_1 scores of LID-elastic Node2Vec variants and the corresponding embedding dimensions. The column "Best" indicates the best graph embedding algorithm according to F_1 and I is the percentage improvement in F_1 of a better LID-elastic Node2Vec variant over `n2v`.

Table 4. Comparison of Node2Vec and LID-elastic Node2Vec embeddings.

	`n2v`		`lid-n2v-rw`		`lid-n2v-rwpq`			
Graph	F_1	Dim.	F_1	Dim.	F_1	Dim.	Best	I [%]
Karate club	0.78	100	0.83	50	0.85	100	`lid-n2v-rwpq`	9.4
Les miserables	0.81	100	0.80	100	0.83	200	`lid-n2v-rwpq`	2.7
Florentine families	0.96	100	0.96	100	0.96	100	all	0.0
CORAML	0.65	25	0.66	50	0.63	25	`lid-n2v-rw`	1.3
CITESEER	0.24	10	0.25	10	0.28	10	`lid-n2v-rwpq`	18.7
AE photo	0.50	50	0.52	50	0.49	50	`lid-n2v-rw`	4.9
AE computers	0.45	50	0.47	100	0.42	50	`lid-n2v-rw`	4.7
PUBMED	0.39	50	0.43	50	0.42	50	`lid-n2v-rw`	9.4
CORA	0.57	25	0.60	50	0.59	50	`lid-n2v-rw`	3.9
DBLP	0.40	25	0.44	25	0.53	50	`lid-n2v-rwpq`	31.7

For Florentine families (the smallest graph in our experimental corpus) both LID-elastic Node2Vec variants achieve the same F_1 score as `n2v`. In all other cases at least one LID-elastic variant is better than `n2v`. For 5 graphs (out of 10) both LID-elastic variants have higher F_1 scores than `n2v`. The `lid-n2v-rw` variant achieves the highest F_1 score for 5 graphs, while `lid-n2v-rwpq` wins in 4 cases. The largest improvements in F_1 are achieved by `lid-n2v-rwpq` for DBLP and CITESEER. For those two graphs `lid-n2v-rwpq` significantly outperforms `n2v`: F_1 is improved by 31.7% and 18.7%, respectively. Significant improvements (approximately 5% or higher) are also present for 4 other graphs (Karate club, AE Photo, AE Computers and PUBMED).

6 Conclusions and Future Work

In this work we have discussed the notion of local intrinsic dimensionality in the context of graphs, which is the first step towards LID-aware ML/DL algorithms for graph-structured data. Since graphs are dimensionless objects, existing LID models could be applied to graphs by computing LID estimators either on graph embeddings or on graph-based distances.

Inspired by the fundamental connection between the local intrinsic dimensionality and the discriminability of distance functions in Euclidean spaces, we have proposed the NC-LID metric quantifying the discriminability of the shortest path distance considering natural communities of nodes in graphs. Then, we have suggested two LID-elastic modifications of the Node2Vec graph embedding algorithm in which Node2Vec parameters are personalized per node and adjusted according to their NC-LID values. Our experimental evaluation of the proposed LID-elastic Node2Vec modifications on 10 real-world graphs revealed that they are able to improve Node2Vec embeddings with respect to graph reconstruction errors.

The current work could be continued in two directions. One direction is to investigate possibilities for designing LID-related metrics reflecting the discriminability of graph-based distance functions considering expanding subgraph localities. In the same way as NC-LID, such metrics could be exploited to personalize and adjust parameters of graph embedding algorithms. Having in mind that nodes with complex intrinsic localities may have a significant brokerage role, it would also be interesting to examine correlations between LID-related scores and node centrality metrics.

The second research direction is related to natural communities. Namely, we will investigate alternative random walk strategies for graph embedding algorithms that explicitly take into account the inner structure of natural communities and characteristics of nodes within them.

Acknowledgments. This research is supported by the Science Fund of Republic of Serbia, #6518241, AI – GRASP. The authors would like to thank the anonymous reviewers for their insightful suggestions and comments that helped improve the quality of the paper.

References

1. Amsaleg, L., et al.: Estimating local intrinsic dimensionality. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2015, pp. 29–38. Association for Computing Machinery, New York (2015). https://doi.org/10.1145/2783258.2783405
2. Amsaleg, L., Chelly, O., Houle, M.E., Kawarabayashi, K.I., Radovanović, M., Treeratanajaru, W.: Intrinsic dimensionality estimation within tight localities. In: Proceedings of the 2019 SIAM International Conference on Data Mining, pp. 181–189. Society for Industrial and Applied Mathematics, May 2019. https://doi.org/10.1137/1.9781611975673.21

3. Becker, R., Hafnaoui, I., Houle, M.E., Li, P., Zimek, A.: Subspace determination through local intrinsic dimensional decomposition: theory and experimentation. arXiv 1907.06771 (2019)
4. Casanova, G., et al.: Dimensional testing for reverse k-nearest neighbor search. Proc. VLDB Endow. **10**(7), 769–780 (2017). https://doi.org/10.14778/3067421.3067426
5. Goyal, P., Ferrara, E.: Graph embedding techniques, applications, and performance: a survey. Knowl. Based Syst. **151**, 78–94 (2018). https://doi.org/10.1016/j.knosys.2018.03.022
6. 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, KDD 2016, pp. 855–864. Association for Computing Machinery, New York (2016). https://doi.org/10.1145/2939672.2939754
7. 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
8. 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.) SISAP 2017. LNCS, vol. 10609, pp. 64–79. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68474-1_5
9. Houle, M.E.: Local intrinsic dimensionality III: density and similarity. In: Satoh, S., et al. (eds.) SISAP 2020. LNCS, vol. 12440, pp. 248–260. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-60936-8_19
10. Houle, M.E., Schubert, E., Zimek, A.: On the correlation between local intrinsic dimensionality and outlierness. In: Marchand-Maillet, S., Silva, Y.N., Chávez, E. (eds.) SISAP 2018. LNCS, vol. 11223, pp. 177–191. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02224-2_14
11. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. New J. Phys. **11**(3), 033015 (2009). https://doi.org/10.1088/1367-2630/11/3/033015
12. Ma, X., et al.: Characterizing adversarial subspaces using local intrinsic dimensionality. In: International Conference on Learning Representations (2018). https://openreview.net/forum?id=B1gJ1L2aW
13. Ma, X., et al.: Dimensionality-driven learning with noisy labels. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, 10–15 July 2018, vol. 80, pp. 3361–3370. PMLR (2018)
14. Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D.: Defining and identifying communities in networks. Proc. Natl. Acad. Sci. **101**(9), 2658–2663 (2004). https://doi.org/10.1073/pnas.0400054101
15. von Ritter, L., Houle, M.E., Günnemann, S.: Intrinsic degree: an estimator of the local growth rate in graphs. In: Marchand-Maillet, S., Silva, Y.N., Chávez, E. (eds.) SISAP 2018. LNCS, vol. 11223, pp. 195–208. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02224-2_15
16. Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. Nature **393**(6684), 440–442 (1998). https://doi.org/10.1038/30918