

Understanding and Evaluating Structural Node Embeddings

Junchen Jin

University of Michigan, Ann Arbor
kinmark@umich.edu

Di Jin

University of Michigan, Ann Arbor
dijin@umich.edu

Mark Heimann

University of Michigan, Ann Arbor
mheimann@umich.edu

Danai Koutra

University of Michigan, Ann Arbor
dkoutra@umich.edu

ABSTRACT

While most network embedding techniques model the proximity between nodes in a network, recently there has been significant interest in *structural embeddings* that are based on node *equivalences*, a notion rooted in sociology: equivalences or positions are collections of nodes that have similar roles—i.e., similar functions, ties or interactions with nodes in other positions—irrespective of their distance or reachability in the network. Unlike the proximity-based methods that are rigorously evaluated in the literature, the evaluation of structural embeddings is less mature. It relies on small synthetic or real networks with labels that are arbitrarily defined, and its connection to sociological equivalences has hitherto been vague and tenuous. To fill in this gap, we set out to understand *what* types of equivalences structural embeddings capture. We are the first to contribute rigorous intrinsic and extrinsic evaluation methodology for structural embeddings, along with carefully-designed, diverse datasets of varying sizes. We observe a number of different evaluation variables that can lead to different results (e.g., choice of similarity measure or label definitions). We find that degree distributions within nodes' local neighborhoods can lead to simple yet effective baselines. We hope that our findings can influence the design of further node embedding methods and also pave the way for future evaluation of existing methods.

ACM Reference Format:

Junchen Jin, Mark Heimann, Di Jin, and Danai Koutra. 2020. Understanding and Evaluating Structural Node Embeddings. In *Proceedings of KDD Workshop on Mining and Learning with Graphs (MLG'20)*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/nmnnnnn.nnnnnnn>

1 INTRODUCTION

Node embeddings capture similarity between nodes in a multi-dimensional space: the closer two nodes are embedded, the more similar they are in the network. Two broad categories of node similarity are prevalent in the representation learning literature: (i) proximity, which intuitively embeds similarly nodes that belong to communities or cohesive groups [28, 32]; and (ii) equivalence or structural similarity, which aims to similarly embed nodes that

have similar patterns of relations with other nodes irrespective of their exact location in the graph [31, 34].

In this work, we focus on the latter *structural node embeddings*. Unlike proximity-based embeddings that build on models of first- or second-order proximity [32], or sample context via random walks [13, 15, 28, 31], structural embeddings are inspired by the notions of roles and positions in sociology. A *position* or *equivalence class* describes a collection of individuals with similar roles, i.e., similar functions, ties or interactions with individuals in other positions [34]. Depending on the type of equivalence (e.g., automorphic, regular—cf. § 3.1), different positions and roles arise that enable both multi-network tasks (e.g., network alignment and classification [18, 31], transfer learning [19]) and single-network tasks, including structural role classification, anomaly detection, and identity resolution [20]. To capture roles in the network, structural embedding methods use feature-based matrix factorization [18, 19] or random walks [29], graphlets [1], or more recently LSTMs [33].

While proximity-based methods are evaluated rigorously on a set of well-understood tasks using established datasets, the evaluation of structural embeddings is less mature. It relies mostly on limited, small synthetic or real datasets (mainly air-traffic networks) with contrived node labels. It also lacks rigorous connections to the types of equivalence from which role discovery in networks stems. To address this gap, we provide a **novel, comprehensive evaluation methodology for systematic analysis of structural embedding methods with respect to the sociological theories of equivalence**. Our main contributions are:

- **Evaluation Methodology.** This is the first paper to introduce a variety of evaluation methods for *unsupervised structural node embeddings*. These are based on: (i) intrinsic measures related to equivalence definitions (§ 3.1), which help us decouple the effectiveness of methods from classifiers in downstream tasks, and (ii) extrinsic measures that characterize their performance in downstream tasks.

- **Appropriate Datasets.** We introduce new benchmark datasets, and ways to obtain ground truth roles (§ 4). We hope that these datasets will change the way structural embeddings are evaluated.

- **Understanding.** Our empirical analysis of 11 methods (§ 3.2) on 31 real and synthetic datasets (§ 4) and a variety of tasks shows that different methods seem best based on different label definitions, embedding similarity functions (e.g., cosine vs. Euclidean), and so on. This analysis highlights that there is no one optimal structural embedding. Moreover, we evaluate the extent to which sociological equivalences are captured by different structural embedding methods (§ 5). Also, besides merely comparing the performance of different methods on downstream tasks, we further analyze their

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MLG'20, August 2020, Online

© 2020 Association for Computing Machinery.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM... \$15.00

<https://doi.org/10.1145/nmnnnnn.nnnnnnn>

performance at a finer granularity to understand for *which types of nodes* current methods perform best (§ 6.2).

- **New Design Insights.** We find that degree distribution in nodes' local neighborhoods is effective as a feature representation in its own right as well as the building block for some of the most successful embedding methods. This can influence the design of future structural embedding methods and/or serve as a standalone baseline for structural embedding tasks.

After reviewing the related work, we present key concepts from social science that have inspired the work on *structural* embeddings.

2 RELATED WORK

Understanding Latent Representations. In NLP, word embedding evaluation is the subject of much study, to the point where a multi-year workshop has arisen dedicated to the evaluation of word embeddings¹, and word embedding evaluation now warrants a survey [2]. Node embeddings are only recently starting to follow suit. A few works [14, 16] benchmark the performance of popular node embedding algorithms on common datasets, a form of *extrinsic* evaluation of the embeddings in the context of various downstream tasks. *Intrinsic* evaluation of node embedding is not as common, with a study of which embeddings could predict various node centrality measures [9] being a recent start. These works focus on proximity-preserving and not structural embeddings.

Node Embeddings. Node embedding is a function mapping nodes V in a network G to d -dimensional feature vectors $\mathbf{x} \in \mathbb{R}^d$ such that "similar" vertices have similar feature vectors, based on some similarity measure. Among the many existing techniques, our study focuses on methods that assume two nodes are similar if they have similar structural roles (defined in § 3.1) *regardless* of their proximity in the network, and 'hybrid' approaches (i.e., that capture both proximity and structural similarity to some extent). We refer the reader to [31] for more information on this distinction.

- **Proximity methods.** In our analysis, we consider two embedding methods that are primarily proximity-based. (1) **node2vec** [15] uses the skip-gram architecture [25] to learn an embedding for each node that preserves its similarity to other nodes in its context, sampled with biased random walks. (2) **LINE** [32] optimizes an embedding objective that maximizes the probability of the first and second-order proximities in the network (direct edges between any two nodes and mutual neighbors that any two nodes share, resp.). Proximity-preserving methods are the topic of numerous surveys [13, 31], and we refer the interested reader to those.

- **Structural methods.** We also evaluate eight structural embedding approaches: (3) **struc2vec** [29] uses the same skip-gram architecture, but samples context with random walks performed over an auxiliary multilayer graph capturing structural similarity (mainly *degree*) of nodes' neighborhoods at several hop distances. (4) **GraphWave** [10] computes the heat kernel matrix for a graph and embeds each node by sampling the empirical characteristic function of the distribution of heat it sends to other nodes. (5) **xNetMF** [18] finds node embeddings that implicitly factorize a structural similarity matrix, defined by comparing the distribution of node degrees in k -hop neighborhoods. Subsequently, (6)

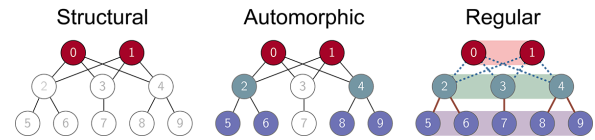


Figure 1: Different types of equivalence. Nodes filled with the same color belong to the same equivalent roles.

SEGK [27] factorizes a structural similarity matrix using graph kernels to compare the nodes' k -hop neighborhoods. (7) **role2vec** [1] applies the skip-gram model to a corpus sampled using *attributed* random walks which record the structural type of each node. The method learns the same embedding for nodes of each structural type, which enhances space efficiency. (8) **RiWalk** [35] also uses the skip-gram model, but learns an embedding for each node based on the structural types of nodes in its context. (9) **DRNE** [33] contends that feature propagation is similar to the recursive definition of regular equivalence, and uses an LSTM to learn node embeddings by aggregating the features of their neighbors sorted sequentially by degree. (10) **MultiLENS** [21], similar to xNetMF, derives embeddings based on matrix factorization that captures the distribution of structural features in nodes' local neighborhoods. In our analysis, we study the connections between these methods and the equivalence theory in social science.

3 PRELIMINARIES

3.1 Equivalence in Social Science

Structural embeddings are related to the notions of *social roles* or *positions*, which are central in sociology for understanding how the society or groups are organized. *Role* refers to the patterns of relations between individuals, or the ways in which individuals relate to each other. *Position* or *equivalence class* describes a collection of individuals with similar activity, ties or interactions with individuals in other positions [34]. The formal definitions of these terms are based on network methods, which led to their wide adoption in social network analysis. In network analysis, (structural) roles of nodes include centers of stars, peripheral nodes, bridge nodes, members of cliques, and more [19].

There are different types of equivalence, each of which is based on an equivalence relation that defines a partition of a node-set to mutually exclusive and exhaustive equivalence classes such that the nodes that are equivalent are assigned to the same class. Among the various types of equivalence, we focus on three main types: structural, automorphic, and regular equivalence.

Structural equivalence [24] is the simplest and most restrictive notion of equivalence:

DEFINITION 1. *Two nodes are structurally equivalent iff they have identical connections with identical nodes.*

For example, in Fig. 1 nodes 0 and 1 are structurally equivalent. Structural equivalence is rarely seen in real-world networks, and it is very strict form of structural similarity that is closely related to proximity: *two structurally equivalent nodes are at most two hops away from each other* [30, 34]. We confirm empirically that proximity-preserving embedding methods best capture this in § 5.

Automorphic equivalence [3] was proposed to relax the notion of structural equivalence. Intuitively, two automorphically

¹<https://repeval2019.github.io/>

equivalent nodes are identical with respect to *all* graph theoretic properties (e.g., in-/out-degree, centralities) and may differ only in terms of their labels. Examples include the nodes in each node-set $\{0, 1\}$, $\{2, 4\}$, and $\{5, 6, 8, 9\}$ of Fig. 1. More formally:

DEFINITION 2. *Two nodes are automorphically equivalent iff there is an automorphism (i.e., an isomorphism in the same graph) that maps one node to the other.*

Although automorphic equivalence is less restricted than structural equivalence (and also a superset of structural equivalence), its exact format is still expected to be rare in real networks.

Regular equivalence [3] is among the most interesting and prevalent types of equivalence in real networks:

DEFINITION 3. *Two nodes are regularly equivalent if they relate in the same way to equivalent nodes.*

For example, similarly colored nodes in Fig. 1 correspond to regularly equivalent classes—e.g., nodes $\{2,3,4\}$ are regularly equivalent because they connect to nodes of the ‘red’ and ‘purple’ roles, although they do not have the same degree (and, thus, it is more relaxed notion than automorphic equivalence).

3.2 Selection of Structural Embedding Methods

Our main goal is to introduce methodology for understanding and evaluating structural embedding methods, not to exhaustively evaluate *all* methods. Thus, we analyze **11 representative methods that use different mechanisms to generate node embeddings**. All methods are unsupervised, unlike graph convolutional networks [22]; this is necessary as our intrinsic evaluation does not depend on a downstream task. We discuss their hyperparameter settings for our analysis in § A.

In addition to the ten ‘hybrid’ and structural methods that we presented in § 2, we also construct variants of **degree distributions** over different neighborhoods, which can be seen as simple, yet strong, baselines for embedding nodes. We represent each node with the degree distribution of its k -hop neighbors—i.e, a histogram of dimension Δ_{\max} , the maximum node degree in each dataset, in which the i -th entry counts the number of neighbors that are k hops away with degree i . We refer to the 11^{th} family of structural approaches that we consider as **degree** that is simply the node’s degree, and **degree1** and **degree2** that are histograms based on 1- and 2-hop neighborhoods.

4 DATA AND GROUND TRUTH ROLES

To gain insights into the type of information that is encoded in structural embeddings, we consider several real datasets (Tab. 1),

Table 1: Real Datasets

Dataset	# Nodes	# Edges	Labels
BlogCatalog [15]	10,312	333,983	centralities
Facebook [15]	4,039	88,234	equivalences (§ 3.1)
ICEWS [6]	1,255	1,414	military vs media entities
Email-300	318	752	professional roles
Email-2K	2,414	11,995	professional roles
PPI [17]	56,944	818,786	protein cellular functions
BR air-traffic [29]	131	1,038	# landings & take-off, equival. (§ 3.1)
EU air-traffic [29]	399	5,995	# landings & take-off, equival. (§ 3.1)
US air-traffic [29]	1,190	13,599	# passengers, equivalences (§ 3.1)
DD6 [5]	4,152	20,640	amino acid properties

and introduce synthetic data (Fig. 2, Tab. 2), the structure of which we can control and understand better than that of real networks.

4.1 Real Network Data

4.1.1 Limitations of existing datasets. The most commonly used real datasets for evaluating the quality of structural embeddings are **air-traffic networks** from [29], which capture the existence of commercial flights (edges) between airports (nodes) and are thus undirected and unweighted [29]. Their node labels are defined based on either the number of landings and take-offs, or the number of passengers passed by each airport in a given time period: four labels are obtained by splitting the data into quartiles. Although the balanced classes simplify the evaluation, this arbitrary labeling has two drawbacks: (1) it is not clear that splitting the data into four quartiles reflects a real-world phenomenon; and (2) to a large extent, we find the labels simply capture degree information.

To experiment with the effect of different node labelings to the performance, we also construct an alternative set of node labels constructed by splitting the airport-related statistics (number of landings and take-offs, or passengers) into logarithmic bins (Fig. 6). This results in imbalanced classes but produces a distribution of “roles” following the well-known power-law distribution.

4.1.2 New datasets for structural embeddings. Besides the existing datasets used in prior works on structural embeddings, we also consider large real-world datasets (Tab. 1), where we can define the node labels based on the different definitions of equivalence (§ 3.1.5). We use the **BlogCatalog** and **Facebook** networks [15], two social network datasets containing various structural roles.

We also propose several additional datasets whose node labels may relate to structural roles. The first is a knowledge graph of the relationships among socio-political actors from the Integrated Crisis Early Warning System (**ICEWS**) [6] based on events on October 4, 2018. Our task is to distinguish between “media” entities and “military” entities. Another real dataset we use is the **PPI network** from [17], a multi-network dataset which is claimed to have node labels corresponding to structural roles rather than communities. We also use a network called **DD6**, one of the larger networks from the D&D dataset of protein structures; nodes, which represent amino acids, have labels representing various properties of the amino acid [5]. These labels exhibit very low homophily and are known to be challenging for proximity-based methods [23]. We also use two proprietary email communication networks, **Email-300** and **Email-2K**, for the users in which we have professional roles (e.g., CEO, manager) that are related to regular equivalence [34].

4.1.3 Ground-truth Node Equivalences or Roles. For our intrinsic evaluation, instead of arbitrarily defining roles in networks, we leverage existing (exact or approximate) algorithms that aim to identify equivalence classes. Given the adjacency matrix \mathbf{A} of a graph, these approaches produce a pairwise node similarity matrix \mathbf{S} based on their respective equivalence definitions. For *structural equivalence*, CONCOR [7] creates a similarity matrix with entries $s_{ij} = s_{ji}$ corresponding to the Pearson correlation between nodes i and j (i.e., the correlation of their respective rows, $\mathbf{A}_{i,:}$ and $\mathbf{A}_{j,:}$). For *automorphic equivalence*, MAXSIM [12] first creates a matrix of geodesic proximities from the adjacency matrix \mathbf{A} , and then creates

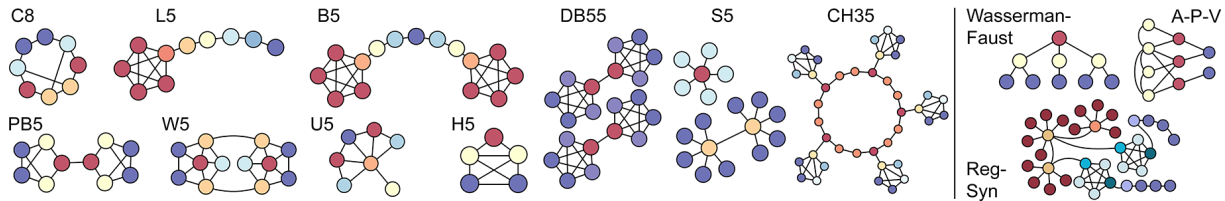


Figure 2: Per synthetic base graph, nodes with the same color are automorphically equivalent on the left & regularly equivalent on the right.

Table 2: Enlarged synthetic graphs

Large Graph	Base	Generation
H10_S_L	H5	10 H5 on a circle with 2 circular nodes between each connecting circular node with house's side.
H10_T_L	H5	10 H5 on a circle with 2 circular nodes between each connecting circular node with house's roof.
Barbell L-A	B5	Connecting the out-most nodes on the chain of B5 into a circle.
Barbell L-B	B5	Connecting the out-most nodes on the chain of B5 into a circle. Additional 5-clique at each connector.
Ferris Wheel	C8	Enlarged version of C8 with similar perturbation.
City of Stars	S5	10 normal stars and 5 binary stars as in S5
PB-L	PB5	10 half-sided PB5 connected to each node of a 10-node circular graph. All the node degrees are 3.
Conference	A-P-V	Mimicking the real-world scenario, we simulate 80 papers with 4~6 collaborators out of the 120 authors, and assign them to one of the 30 venues.
Reg-Syn-L	Reg-Syn	Based on the connection rules in Reg-Syn, we connect 9 stars, 7 cliques and 7 chains of different sizes.
Knitting Wheel	B5	10 different sized cliques connected onto a circle with three circular nodes apart each connection.

S by comparing the node distributions of geodesic proximities pairwise. For *regular equivalence*, CATREGGE [4] searches for matches in successive node neighborhoods, and encodes in S the iteration in which two nodes were separated into different groups or classes.

CONCOR also produces a partition that we use as the *ground-truth* equivalence classes (i.e., groups of nodes with similar roles). To obtain the ground truth for MAXSIM and CATREGGE, we apply hierarchical clustering on S (with default settings).

4.2 Synthetic Network Data

We also evaluate structural embedding techniques on a variety of synthetically-generated networks—beyond just the commonly-used barbell graph—as shown in Fig. 2 (left).

We define two sets of roles per node, based on *structural* and *automorphic* equivalence—using the methods CONCOR and MAXSIM (§ 4.1.3), respectively. We also enlarge the small synthetic graphs to enable further extrinsic evaluation (Table 2). For *regular* equivalence, since nodes should be assigned to different classes according to their roles, we generate the synthetic graphs accordingly (Fig. 2, right). Similarly, we enlarge the synthetic graphs by adding more nodes with different roles and connecting them following the rules in the base case (Table 2). For all the synthetic graphs generated for the regular equivalence evaluation, the edge type is indicated by the pre-defined roles of the end-points (e.g., hub vs. clique node). The output of CATREGGE (§ 4.1.3) generates the *same* role assignment as the pre-defined roles.

5 EMBEDDINGS AND EQUIVALENCES

In the literature, there are various claims about the types of equivalence that embedding methods capture, some of which are imprecise. We investigate this by designing experiments for both

intrinsic and extrinsic evaluation. Our **intrinsic evaluation** aims to evaluate the quality of embeddings in the context of different types of equivalences *directly*, decoupled from a downstream task. Here, *ground-truth labels* are defined by the equivalence methods (§ 3.1, 4.1.3). Our **extrinsic evaluation** relies on classification and clustering, both of which are typically used to evaluate embeddings.

5.1 Intrinsic Evaluation

The intrinsic evaluation of structural embeddings seeks to characterize the agreement between the similarities of nodes defined by the different types of equivalence and the node similarities in the embedding space \mathbb{R}^d .

5.1.1 Methodology. Given a similarity matrix S based on a notion of node equivalence (4.1.3), for each node we calculate the Kendall rank correlation coefficient between its embedding similarity (based on Euclidean distance or cosine similarity²) and its structural similarity to all other nodes given by S .

For structural and automorphic equivalence, we perform analysis on a total of 16 synthetic networks (Fig. 2 left plus the enlarged datasets in the top section of Table 2, CH35 excluded as near-duplication of Small Town-S) and 4 real networks (three air-traffic networks + Facebook). One exception is that for structural equivalence, CONCOR encounters an error for City of Stars, for which we skipped evaluation. For regular equivalence, we analyze 5 synthetic datasets (Fig. 2 right, plus the enlarged datasets in the bottom section of Table 2, A-P-V excluded as duplication of Conference). CATREGGE cannot compute regular equivalence on our real networks for an intrinsic evaluation, as the implementation handles up to 255 nodes. For each type of equivalence, we report the average and the standard deviation of the Kendall rank correlation coefficient across different subsets of our datasets.

5.1.2 Results. Figure 3 gives a summarized view of our intrinsic evaluation. It shows, per embedding method, the rank correlation and its standard deviation averaged over all the corresponding datasets. LINE and node2vec rank top in our intrinsic evaluation for **structural equivalence**. This is expected, as despite its name, structural equivalence is actually by definition best captured by proximity-based embedding methods [30, 34]. It is defined between two nodes in terms of how many neighbors they share: two nodes are structurally equivalent if they are connected to the exact same nodes. **Structural equivalence as defined in mathematical sociology is distinct from the structural similarity that role-based node embeddings try to capture.**

On the other hand, structural embedding methods such as GraphWave, xNetMF and SEGK, as well as degree2, work well in terms of

²It is not defined for a scalar (e.g., degree), in which case we list "N/A" in Figs. 3-4.

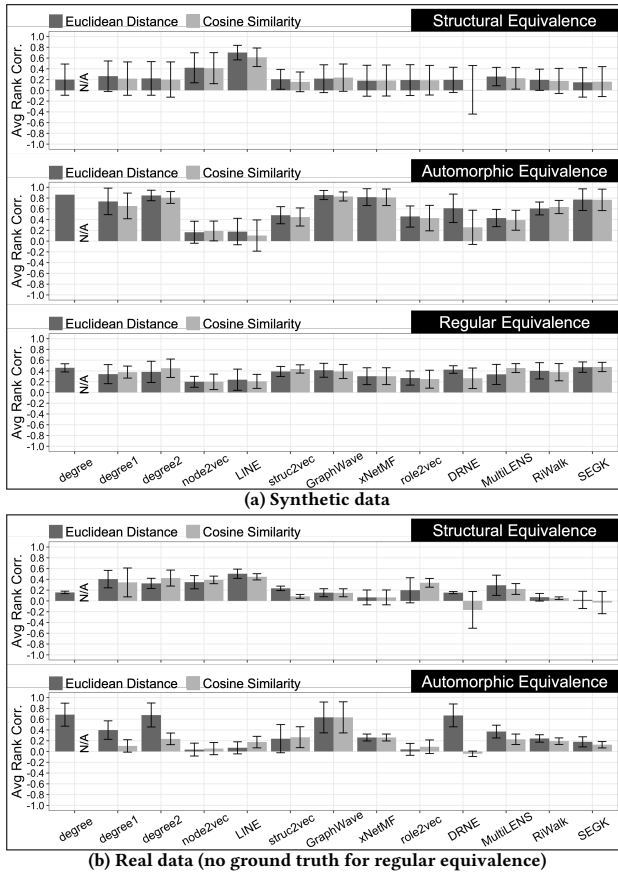


Figure 3: Summarized view of intrinsic evaluation: Average correlation (and stdev) between node embeddings and different types of equivalences across all synthetic data (top) and all real data (bottom). Structural embeddings tend to capture automorphic and regular equivalence, while primarily proximity embeddings capture structural equivalence. The choice of distance affects the results.

automorphic equivalence, while the proximity-based methods, like LINE and node2vec do not. This is also expected, as **automorphically similar nodes need not be in close proximity in the graph**. We conjecture the difference in degree distribution and network structure on the synthetic datasets and real world datasets might account for the difference in role2vec’s performance.

Similarly, the proximity-based node2vec and LINE struggle to capture **regular equivalence**, which among structural embedding methods is generally best captured by degree, DRNE, and GraphWave based on Euclidean distance, and degree2, MultiLENS, and struc2vec based on cosine similarity. The strong performance of degree distribution features in the intrinsic evaluation using automorphic and regular equivalence is noteworthy. This suggests that **node degree, generalized to include the distribution in its k -hop neighborhood, may indeed be a good indicator of the structural position or role of the node in the network**.

In Fig. 4, we look deeper into these results on a per-dataset basis. While trends are largely similar, some datasets are worth noting individually. For example, we see that the base “L5” has a distinctive “lollipop” shape, where equivalent nodes (in the head) and comparatively near-equivalent nodes (in the stem) are also in close proximity. As a result, proximity-preserving and structural

embeddings do comparably well at capturing both structural and automorphic equivalence. We see larger gaps on the remaining synthetic datasets. On real datasets, GraphWave and DRNE capture extremely high automorphic equivalence on the air-traffic datasets, but the difference between them and the other methods disappears on Facebook, a social network dataset.

Our findings confirm that none of the embedding methods are optimized to capture these sociological equivalences.

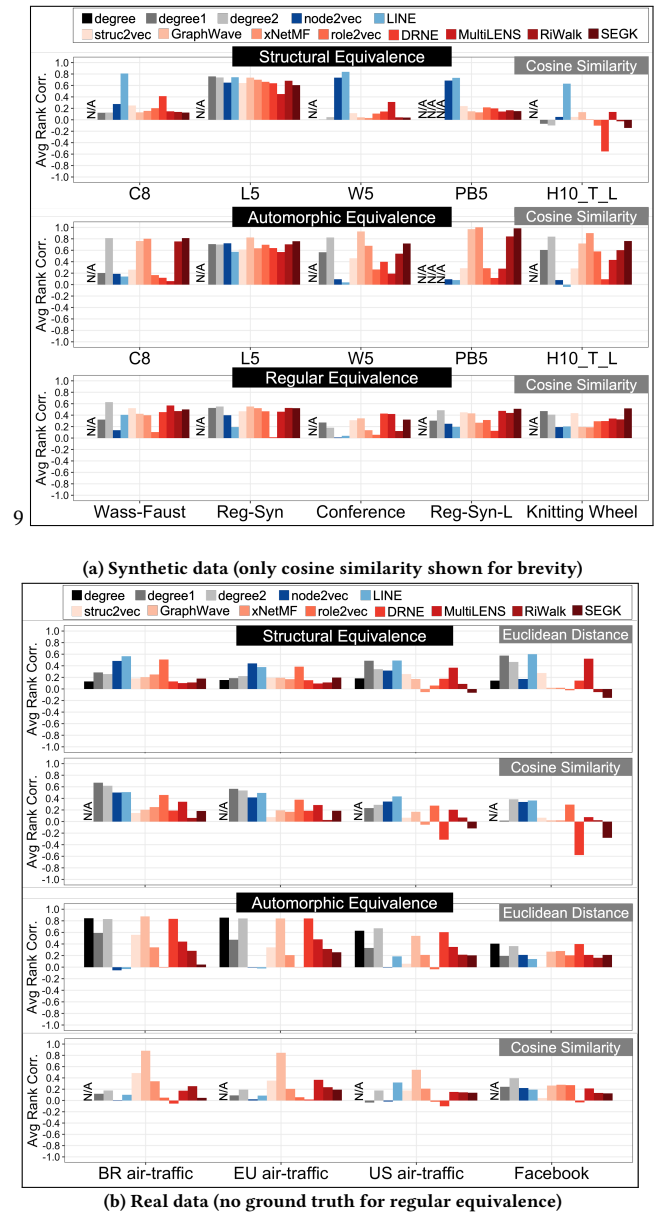


Figure 4: [Best viewed in color] Detailed view of intrinsic evaluation: correlation with different types of equivalence for specific synthetic (top) and real (bottom) datasets. Performance of embedding methods varies across different datasets and distance choices.

Although we find that they do capture them to some extent incidentally, it depends on how well the equivalences correspond in any given dataset with the types of similarities each embedding is optimized to preserve (the choice of distance, Euclidean or cosine, has significant impact for some methods, especially in the real data.).

5.2 Extrinsic Evaluation

We also evaluate the structural embeddings extrinsically by defining *equivalence-specific* node labels.

5.2.1 Methodology. As in § 5.1.1, we consider the equivalence-specific similarity matrix S and the network embeddings E . To obtain the ground-truth *equivalence classes* (i.e., node labels), we perform hierarchical clustering on S for MAXSIM and CATREGE, and use the CONCOR partitioning output directly (§ 4.1.3). Again, for the synthetic datasets used for automorphic equivalence evaluation, we manually define the *exact* automorphically equivalent classes (instead of using MAXSIM’s approximation). With the classes generated or pre-defined, we perform classification and clustering for extrinsic evaluation. (Details of our setup are provided in § B.)

In Fig. 5 we show the results for all three types of equivalence on synthetic (left) and real (right) data. For structural and automorphic equivalence evaluation, we use the enlarged synthetic graphs described in the top section of Table 2. Again, we exclude City of Stars for structural equivalence evaluation as in § 5.1.1. For the real data evaluation, we use the three air-traffic networks and Facebook. For regular equivalence, we use the enlarged synthetic graphs described in the bottom section of Table. 2. No real world dataset is appropriate for regular equivalence evaluation as discussed before.

5.2.2 Results. We generally see similar trends to the intrinsic evaluation. For example, proximity-based methods node2vec and LINE are generally best at capturing structural equivalence in both real and synthetic datasets, in supervised and unsupervised downstream tasks. They take a backseat to most other methods, however, at predicting automorphic or regular equivalences. We observe, however, that MultiLENS improves considerably in downstream tasks.

Differences between methods are often more pronounced in synthetic datasets, which are designed to exhibit highly distinctive structural roles. For instance, LINE and node2vec are over 4× more accurate at predicting structural equivalence than structural embeddings GraphWave and xNetMF, a gap that remains but shrinks considerably in the real datasets. Similarly, in synthetic datasets, GraphWave and xNetMF achieve near-perfect clustering scores, as do 1-hop and 2-hop degree distribution features (which perform competitively at capturing equivalences across our extrinsic evaluations). However, the trend for MultiLENS reverses: extremely poor prediction of structural equivalence on synthetic datasets but strong predictive power on the real datasets.

5.2.3 Discussion. In general, **we observe similar results between intrinsic and extrinsic evaluation as well as synthetic versus real networks.** This suggests that intrinsic evaluation of structural embeddings can *often* be a good proxy of its ability to perform in a downstream task. Similarly, synthetic networks that can be manufactured to exhibit distinctive structural roles that are known *a priori* are *often* a good controlled experimental environment for structural node embedding. However, **there may be exceptions**

to these trends: MultiLENS is one in both cases, performing far better in extrinsic evaluation and on real data. The word embedding literature has noted that intrinsic evaluations of embeddings may not always accurately predict performance in downstream tasks [8]. Thus, both forms of analysis are worthwhile to perform.

6 MINING WITH STRUCTURAL EMBEDDING

We now compare methods for structural node embedding on real-world networks and task-specific settings, including classification and clustering, on graph mining tasks with *externally given node labels* (unlike § 5.2 that relied on equivalence-defined labels). Before

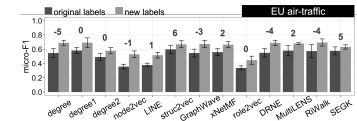


Figure 6: Different labeling schemes: Numbers represent decrease in ranking under new labeling.

presenting comparative results, we identify two important real-world observations that can confound the fair evaluation of structural embeddings on real datasets. We thus perform analysis of how methods’ performance varies as a function of these factors.

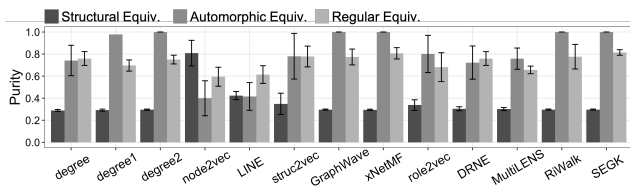
6.1 The Effect of Label Definitions

In Fig. 6, we show the results of different embedding methods on the EU air-traffic datasets for two different labeling schemes: the original ones resulting in balanced classes, and our relabeling in § 4.1.1. We report Micro-F1 scores obtained using logistic regression, and annotate the decrease in ranking under the new labeling, per method. **We see noticeable differences in performance under the two different labeling methods; future works should be mindful of methods’ sensitivity to artificially defined label definitions.** In several cases, this can change the comparative ranking of the different methods. For example, MultiLENS and RiWalk are in the middle of the pack under the old labels but the best methods at predicting the new labels.

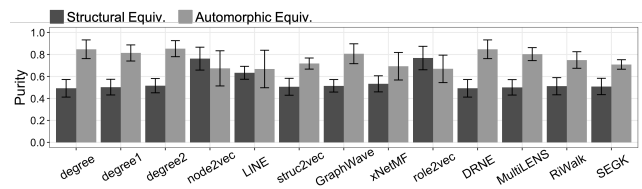
Recent works have observed that node classification involves a labeling process that may be uncorrelated with the graph itself, which may complicate evaluation [11]. In these airport datasets, where the labels were arbitrarily discretized, this issue is even more pronounced. The fact that two (reasonable) ways of generating node labels can yield different results among structural embedding methods suggest that **as each structural embedding method best captures certain structural roles in the network, it is an empirical question how well these roles are correlated with the labels.** (Note that the airport labels are not connected to any particular roles.) This motivates our intrinsic analysis.

6.2 Deeper View Into the Performance Scores

Aggregate performance of a classifier over the whole dataset does not tell the whole story. It is also worth exploring what kinds of nodes (e.g., high degree) can be most easily classified by the various structural embedding methods. For degree-based analysis, per dataset with maximum degree Δ_{\max} , we categorize the nodes into low-degree $[0, \Delta_{\max}^{\frac{1}{3}})$, medium-degree $[\Delta_{\max}^{\frac{1}{3}}, \Delta_{\max}^{\frac{2}{3}})$ and



(a) Synthetic data



(b) Real data

Figure 5: Extrinsic Evaluation on downstream tasks. Mean and standard deviation is presented for each method on all corresponding synthetic datasets and real datasets for three types of equivalence. Generally, the extrinsic evaluation aligns with the intrinsic evaluation.

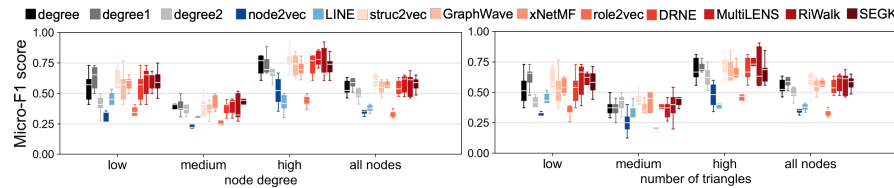


Figure 7: [Best viewed in color] Performance by node degree and participating triangles on the original label on EU air-traffic: nodes with more “extreme” degrees are more accurately classified. Box plot based on 5-fold CV results.

high-degree $[\Delta_{min}^2, \Delta_{max}^2]$ buckets. We then perform classification evaluation per bucket. We apply the same partitioning methodology for the analysis of participating triangles.

In Fig. 7 we present the results of both degree- and participating triangle-based analysis for the EU air-traffic network (we see similar trends in other data). Its maximum degree and maximum number of participating triangles are 202 and 3450, respectively. We observe that in general, **all methods perform best at classifying nodes with high connectivity, as measured by either degree and/or participating in a large number of triangles.** This is not surprising and corroborates the literature, as these nodes’ local neighborhoods contain richer information [26]. Slightly more surprisingly, the least-connected nodes are the next easiest to classify. This suggests that **it may be easier for structural embedding methods to distinguish “extreme” network positions in the latent feature space than moderate ones.**

Some network positions are easy to identify. For instance, simply using the node degree as a feature (degree) performs best at classifying high degree nodes, but is less effective at classifying low- and medium-degree nodes even compared to degree1, where neighbors’ degrees are considered as features. In general, however, relative ranks of methods are fairly well-preserved across buckets.

6.3 A Comprehensive Embedding Comparison

Having carefully considered the effects of several external factors, we now offer a more comprehensive comparison of embedding methods in Fig. 8: we give their general rankings (lower is better) across all real datasets. We observe that there is no clear winner of an embedding method on all datasets. However, we can see that proximity-preserving embeddings—node2vec and LINE—generally have poorer rankings, as is to be expected. In general, we note that **methods capturing**

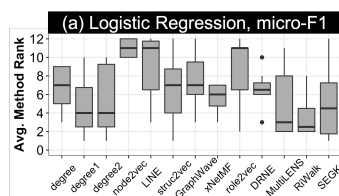


Figure 8: Lower is better: performance summarized across all the real datasets. Methods based on local degree distribution tend to be consistently top performers.

the degree distributions in local neighborhoods are among the most effective. These include xNetMF, MultiLENS, SEGK and variations of our degree distribution features: all perform competitively, notching among the best rankings across the board. The expressive power of local degree distributions has strong implications for future work in structural embedding, as a baseline and an inspiration for methodological design.

7 DISCUSSION AND CONCLUSIONS

We conducted a comprehensive empirical study to gain a better understanding of the *equivalence* of the nodes in the networks within the context of embeddings. Our study of the various sociological equivalences confirms that structural equivalence is best captured by proximity-preserving embedding methods like node2vec and LINE, as its definition implies despite its name. On the other hand, methods like struc2vec, xNetMF and GraphWave perform well in automorphic and regular equivalence.

We have split our analysis into two parts (§ 5): intrinsic evaluation, which explores the relationship of nodes’ embedding similarities and other measures of similarity given by sociological equivalence, and extrinsic evaluation of the embeddings’ performance in the context of downstream tasks such as classification or clustering. Our work is one of the first to perform intrinsic *and* extrinsic evaluation of node embeddings (either structural or proximity-based).

While we largely observe similar performance trends in intrinsic and extrinsic evaluation, we also notice some inconsistent trends, a phenomenon which has also been observed in word embedding [8]. For example, MultiLENS is far from a standout in intrinsic evaluation but a top runner in extrinsic evaluation. In both intrinsic and extrinsic clustering evaluation, we have found a complex relationship between the distance metric used (cosine or Euclidean) and the results, which perhaps surprisingly is not always consistent with the metric used in the various embedding objectives. We also found that different ways of defining node labels can significantly alter the relative rankings of many different methods.

Comparing comprehensively across datasets, we see that the simple structural property, node degree, can be the building block for some of the most effective methods. Our *local degree histograms* are a simple baseline that proves surprisingly effective across all of our experiments. They may inspire the design of future methods:

indeed, they are highly related to xNetMF and MultiLENS, two existing embedding methods that also generally perform well.

Overall, we hope that our findings can influence the design of further node embedding methods and also pave the way for future evaluation of existing methods. With new node embedding methods being developed at a breakneck pace, proper evaluation will, as the word embedding community has found, be essential to progress.

REFERENCES

- [1] Nesreen K. Ahmed, Ryan A. Rossi, John Boaz Lee, Theodore L. Willke, Rong Zhou, Xiangnan Kong, and Hoda Eldardiry. role2vec: Role-based network embeddings. In *DLG KDD*, 2019.
- [2] Amir Bakarov. A survey of word embeddings evaluation methods. *arXiv preprint arXiv:1801.09536*, 2018.
- [3] Stephen Borgatti and Martin Everett. Notions of position in social network analysis. *Sociological Methodology*, 22, 01 1992.
- [4] Stephen P. Borgatti and Martin G. Everett. Two algorithms for computing regular equivalence. *Social Networks*, 15(4):361 – 376, 1993.
- [5] Karsten M. Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *ICDM. IEEE*, 2005.
- [6] Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, and James Starz. ICEWS Automated Daily Event Data, 2018.
- [7] Ronald L. Breiger, Scott A. Boorman, and Phipps Arabie. An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling. *J. Math. Psychol.*, 12(3):328–383, 1975.
- [8] Billy Chiu, Anna Korhonen, and Sampo Pyysalo. Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *Proceedings of the 1st workshop on evaluating vector-space representations for NLP*, pages 1–6, 2016.
- [9] Ayushi Dalmia and Manish Gupta. Towards interpretation of node embeddings. In *Companion Proceedings of the The Web Conference 2018*, pages 945–952, 2018.
- [10] Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. Learning structural node embeddings via diffusion wavelets. In *KDD*, volume 24, 2018.
- [11] Alessandro Epasto and Bryan Perozzi. Is a single embedding enough? learning node representations that capture multiple social contexts. In *WebConf*, pages 394–404, 2019.
- [12] Martin G. Everett and Steve Borgatti. Calculating role similarities: An algorithm that helps determine the orbits of a graph. *Social Networks*, 10(1):77 – 91, 1988.
- [13] Palash Goyal and Emilio Ferrara. Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94, 2018.
- [14] Palash Goyal, Di Huang, Ankita Goswami, Sujit Rokka Chhetri, Arquimedes Canedo, and Emilio Ferrara. Benchmarks for graph embedding evaluation. *arXiv preprint arXiv:1908.06543*, 2019.
- [15] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *KDD. ACM*, 2016.
- [16] Saket Gurukar, Priyesh Vijayan, Aakash Srinivasan, Goonmeet Bajaj, Chen Cai, Moniba Keymanesh, Saravana Kumar, Pranav Maneriker, Anasua Mitra, Vedang Patel, et al. Network representation learning: Consolidation and renewed bearing. *arXiv preprint arXiv:1905.00987*, 2019.
- [17] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.
- [18] Mark Heimann, Haoming Shen, Tara Safavi, and Danai Koutra. Regal: Representation learning-based graph alignment. In *CIKM. ACM*, 2018.
- [19] Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. Rolx: structural role extraction & mining in large graphs. In *KDD*, 2012.
- [20] Di Jin, Mark Heimann, Ryan A. Rossi, and Danai Koutra. Node2bits: Compact time- and attribute-aware node representations for user stitching. In *PKDD*, 2019.
- [21] Di Jin, Ryan A. Rossi, Eunye Koh, Sungchul Kim, Anup Rao, and Danai Koutra. Latent network summarization: Bridging network embedding and summarization. In *KDD*, 2019.
- [22] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.
- [23] John Boaz Lee, Ryan Rossi, Xiangnan Kong, Sungchul Kim, Eunye Koh, and Anup Rao. Graph convolutional networks with motif-based attention. In *CIKM*, 2019.
- [24] Francois Lorrain and Harrison C. White. Structural equivalence of individuals in social networks. *The Journal of Mathematical Sociology*, 1(1):49–80, 1971.
- [25] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 2013.
- [26] Sharad Nandanwar and M Narasimha Murty. Structural neighborhood based classification of nodes in a network. In *KDD*, pages 1085–1094, 2016.
- [27] Giannis Nikolentzos and Michalis Vazirgiannis. Learning structural node representations using graph kernels. *TKDE*, 2019.
- [28] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD. ACM*, 2014.
- [29] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. struc2vec: Learning node representations from structural identity. In *KDD. ACM*, 2017.
- [30] Ryan A. Rossi and Nesreen K. Ahmed. Role discovery in networks. *TKDE*, 27(4):1112–1131, 2015.
- [31] Ryan A. Rossi, Di Jin, Sungchul Kim, Nesreen K. Ahmed, Danai Koutra, and John Boaz Lee. From community to role-based graph embeddings. *arXiv preprint arXiv:1908.08572*, 2019.
- [32] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. Line: Large-scale information network embedding. In *WWW*, 2015.
- [33] Ke Tu, Peng Cui, Xiao Wang, Philip S. Yu, and Wenwu Zhu. Deep recursive network embedding with regular equivalence. In *KDD*, pages 2357–2366, 2018.
- [34] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*. Structural Analysis in the Social Sciences. Cambridge University Press, 1994.
- [35] Ma Xuewei, Geng Qin, Zhiyang Qiu, Mingxin Zheng, and Zhe Wang. Riwalk: Fast structural node embedding via role identification. In *ICDM. IEEE*, 2019.

A EMBEDDING HYPERPARAMETERS

Unless otherwise mentioned, we set parameters to default values reported in the papers and/or official implementations. For fair comparison, we transform all the input networks to be *undirected* and *unweighted*. We learn 128-dimensional embeddings by default.

- For node2vec [15], we bias the random walks with parameters $p = 1$ and $q = 4$, the parameter values considered in the original paper [15] that capture the most structural similarity.
- For the skip-gram methods (node2vec, struc2vec [29], RiWalk [35], and role2vec [1]), we sample context with 10 random walks per node (80 for struc2vec's more complex multi-layer structural similarity network) of length 80. We set the window size to 10 and optimize the objective using 10 iterations of gradient descent. We use all three scalability optimizations for struc2vec and degree (or motifs, if applicable) as role2vec's features.
- For LINE [32], we set the order to be 2 and the total number of training samples to be 100 million and negative samples to be 5.
- For GraphWave, we use exact calculation of the heat kernel matrix with automatic scale selection [10].
- For struc2vec, xNetMF [18], and SEGK [27], we consider up to 2-hop neighborhoods. In RiWalk, k , we used its default node neighborhood radius $k = 4$. In xNetMF, we set the hop distance discount factor to 0.1 and set the similarity resolution $\gamma = 1$. For SEGK [27], we compare neighborhoods using the Weisfeiler-Lehman graph kernel, also used in RiWalk to identify structural roles of nodes based on their local neighborhoods [35].
- For DRNE [33], we follow the example usage to set the batch size to be 256 and the learning rate to be 0.0025.
- For MultiLENS [21], we set the cat input with all nodes having the same category/type.

B SETTINGS FOR DOWNSTREAM TASKS

- **Classification Setup.** For each dataset, we use 5-fold cross validation to get the average performance and standard deviation. A multinomial logistic regression with l_2 penalty $C = 1.0$ is trained to perform multi-class classification.
- **Clustering Setup.** Per dataset, we use k -means to cluster the embeddings E , setting k to be the number of ground truth clusters. To mitigate the effects of algorithmic instability, we run k -means for 1,000 times with different centroid seeds and use the best output in terms of the inertia criterion.