Hindawi
Complexity
Volume 2021, Article ID 5571006, 17 pages
https://doi.org/10.1155/2021/5571006

Research Article

An Automorphic Distance Metric and Its Application to Node Embedding for Role Mining

Víctor Martínez, Fernando Berzal, and Juan-Carlos Cubero

Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain

Correspondence should be addressed to Fernando Berzal; berzal@acm.org

Received 2 February 2021; Accepted 14 September 2021; Published 13 October 2021

Academic Editor: Atila Bueno

Role is a fundamental concept in the analysis of the behavior and function of interacting entities in complex networks. Role discovery is the task of uncovering the hidden roles of nodes within a network. Node roles are commonly defined in terms of equivalence classes. Two nodes have the same role if they fall within the same equivalence class. Automorphic equivalence, where two nodes are equivalent when they can swap their labels to form an isomorphic graph, captures this notion of role. The binary concept of equivalence is too restrictive, and nodes in real-world networks rarely belong to the same equivalence class. Instead, a relaxed definition in terms of similarity or distance is commonly used to compute the degree to which two nodes are equivalent. In this paper, we propose a novel distance metric called automorphic distance, which measures how far two nodes are from being automorphically equivalent. We also study its application to node embedding, showing how our metric can be used to generate role-preserving vector representations of nodes. Our experiments confirm that the proposed automorphic distance metric outperforms a state-of-the-art automorphic equivalence-based metric and different state-of-the-art techniques for the generation of node embeddings in different role-related tasks.

1. Introduction

Role discovery is defined as the process of finding sets of nodes following similar connectivity patterns or structural behaviors [1]. The role of a node can be understood as the function that node plays in the network. Different studies have shown the importance of roles in different domains, including predator-prey food webs [2], international relations [3], or the function of proteins in proteomes [4].

Unfortunately, this problem has received limited attention when compared to community detection [5–7], despite that role discovery identifies complementary information and has found applications in several useful network data mining tasks. For example, roles can be used to model and characterize the behavior of entities in a network, to predict structural changes, and to detect anomalies [8]. Since the same roles can be observed across different networks, this information has been successfully exploited for transfer learning [9]. Role information can also be used for enhancing the visualization of interesting patterns in graphs [10]. Additional applications of role discovery have been described in the scientific literature [1].

Formally, two nodes have the same role if, given an equivalence relation, they belong to the same equivalence class [11, 12]. Different equivalence classes have been studied for nodes in networks.

Structural equivalence, where two nodes play the same role if they are connected to exactly the same neighbor nodes, has been widely studied [13, 14]. These nodes will have exactly the same topological properties, such as degree, clustering coefficient, or centrality, since they are indistinguishable from a structural point of view. However, different authors have pointed out the limitations of structural equivalence for modeling roles or positions, which is the name that roles receive in sociology, since structural equivalence is more related to the concept of locality than the actual concept of role [15]. If the constraint of having to be connected to exactly the same neighbors is relaxed to being connected to neighbors with exactly the same topological function, we obtain automorphic equivalence classes, where two nodes are equivalent if they can swap their labels to form an isomorphic graph [16, 17]. Automorphically equivalent nodes will also have exactly the same topological properties, but, without the requirement of locality imposed by structural equivalence, pairs of nodes at distances larger than two can still have the same role. Therefore, automorphic equivalence is more closely related to the intuitive concept of role, which is understood as the function of a node within a network.

Other equivalence classes, less relevant than the previously mentioned ones, are not covered in this work. Regular equivalence deserves a special mention due to its importance as a relaxation of automorphic equivalence that only requires being connected to nodes with the same function, omitting the actual count of connections [18]. Regular equivalence does not preserve topological properties and is more suited to hierarchically organized networks [2].

These binary equivalences are strict mathematical abstractions that rarely occur in real-world networks, leading to all nodes being assigned a different role. In practice, equivalences are relaxed to similarities, allowing two nodes to play the same role by partially satisfying the constraints imposed by the mathematical definition of structural, automorphic, or regular equivalence.

In this paper, we present a novel automorphic distance metric, capturing distances between nodes in terms of automorphic equivalence. According to the network structure, two nodes will be at a distance that is proportional to how far they are from being automorphically equivalent. This leads to a softer definition of automorphic roles, instead of forcing all nodes to fit in strict classes of roles. However, when needed, these distances can be used to discover and instantiate specific role classes. Our distance function satisfies metric axioms, as we prove below, does not require external parameters nor feature engineering, and is computable for nodes across different networks. We also present different applications of our proposal, with special emphasis on generating node embeddings that preserve node roles. Node embeddings are vector representations of nodes capturing relevant information in terms of pairwise distances [19]. Much work has been done in embedding techniques that preserve neighborhoods or communities [20–22]. However, role-preserving embeddings have only recently begun to be studied [23, 24].

Our paper is structured as follows. In Section 2, we discuss the relevant related work in automorphic distance metrics. In Section 3, we describe our proposed automorphic distance metric and study its admissibility as a distance metric, as well as its computational complexity. In addition, we present an approximated algorithm to compute our proposed automorphic distance that offers higher scalability. In Section 4, we analyze its performance for different role-related tasks and show how it outperforms previously proposed approaches. Finally, conclusions and suggestions for future research are presented in Section 5.

2. Related Work

Different metrics have been proposed to measure node similarity. One of the most popular metrics is SimRank [25], which iteratively computes similarity scores based on the hypothesis that two nodes are similar if they link to similar nodes. Different extensions of SimRank have been proposed [26]. SimRank recursively computes the similarity of two nodes according to the average similarity of all their neighbor pairs, which can also be interpreted, as suggested by its original authors, as how soon two random walkers will meet if they start from these nodes. Thus, this definition is not suitable as a metric of similarity capturing automorphic equivalence because it requires the two nodes to be close to play the same role. Other similarity measures not based on SimRank have been proposed, such as PageSim [27] and Leicht's vertex similarity [28]. However, these similarities have been formally rejected as valid metrics for capturing automorphic equivalence [29].

Since automorphic equivalence ensures the same topological properties, some authors have tried to capture automorphic equivalence by defining a similarity function over a set of network topological properties [30]. The problem of these feature-based methods is that they require combining different complex hand-crafted features provided by experts, which is far from a trivial process in practice. In addition, they cannot guarantee, which set of features will correctly approximate automorphic equivalence, resulting in a very limited approach for automorphic equivalence discovery.

As far as we know, RoleSim [29, 31] is the only proposed metric that tries to formally capture the concept of automorphic equivalence without using limited approximations based on hand-crafted topological features. Omitting the decay factor they introduce, by setting it to 0 in order to capture the global network topology, this similarity measure is iteratively computed until convergence as

$$s(x,y) = \max_{M(x,y)} \frac{\sum_{(u,v)\in M(x,y)} s(u,v)}{\deg(x) + \deg(y) - |M(x,y)|}, \tag{1}$$

where $\deg(n)$ is the degree of a node n and $M(x,y)$ is the optimal assignment of nodes in the neighborhood of x to nodes in the neighborhood of y maximizing the expression, that is, the pairs of neighbors of x and y with maximal similarity. In the original manuscript, this function is presented as a role similarity metric by proving the corresponding distance metric axioms. RoleSim is a form of generalized Jaccard coefficient based on a recursive definition of the similarity of neighbor roles. Despite the admissibility of RoleSim, their approach presents several limitations. The RoleSim similarity can be considered an automorphic distance by taking its complementary or Jaccard distance: $d(x,y) = 1 - s(x,y)$. The problem is that the Jaccard coefficient is a normalized metric, which leads to a normalized distance. As will be shown in our experimentation, this normalization has a negative impact on the results obtained by RoleSim. In addition, this similarity function exhibits serious inconsistencies. For example, in the graph shown in Figure 1, where node d has a one-to-many relationship to x_i nodes, the node pair (a,c) has the same exact similarity as any pair (a,x_i), independent of the number of x_i nodes. This simple example shows the limitations of RoleSim when trying to capture the automorphic similarity.

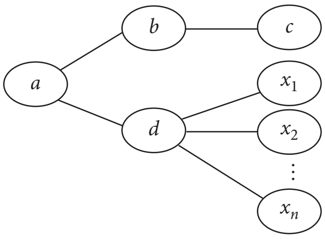


Figure 1: Example graph where RoleSim yields inconsistent values. Node d has a one-to-many relation to x_i nodes.

As far as we know, no distance metric has been proposed that is able to capture the concept of automorphic distance in a consistent way, without relying on approximations based on extracted topological properties nor forcing the normalization of the distance function.

3. A Novel Automorphic Distance Metric

An isomorphism is a bijection between the nodes of two graphs where two nodes are adjacent in one graph if and only if the nodes that result from applying the bijective function are also adjacent in the other graph. An automorphism is an isomorphism from one graph to itself. Therefore, two nodes are automorphically equivalent if there exists an automorphism creating a correspondence between them.

One form of testing for automorphic equivalence is computing the canonical form of graphs. Graph canonicalization is the task of computing a labeling for nodes in a graph such that every isomorphic graph yields the same canonical labeling. Given a canonicalized graph, two automorphically equivalent nodes must have been assigned the same label. As previously stated, automorphic equivalence is too restrictive to appear in real-world networks, leading to most nodes having different canonical labels.

The solution that we propose to this problem is the definition of distances between labels, which ultimately allows the definition of distances between nodes based on the concept of automorphic equivalence. This distance will be proportional to the number of changes that need to be done in the network to transform one label or equivalence class into another. A zero distance implies that two nodes are automorphically equivalent and play exactly the same role. According to this distance d, we can say that nodes x and y are more automorphically similar or have a more similar role to u and v, if $d(x, y) < d(u, v)$. In order to propose a valid distance metric, we must also prove that our metric satisfies the metric distance axioms.

Our work is based on the 1-dimensional Weisfeiler–Lehman test of isomorphism [32, 33], also known as color refinement, which is an algorithm to compute the canonical labeling of graphs. These canonical labels can be used to solve related problems, such as the computation of efficient graph kernels [34]. The Weisfeiler–Lehman algorithm works by initially assigning a label to each node according to its degree, so nodes with the same degree have the same initial label. Then, the algorithm iteratively updates these labels by the following procedure. First, it takes the labels from neighbor nodes, concatenates them according to certain arbitrary order (the same ordering must be applied for all nodes), and finally appends the label of the node at the beginning of the obtained list. Each different sequence is substituted by a newly generated unique label, so nodes exhibiting exactly the same sequence are assigned the same label. This refinement process is repeated until labels stabilize, that is, when every pair of nodes with the same label in the previous iteration have the same label in the current iteration. Therefore, after m iterations, which depend on the network diameter, the canonical form is achieved, and an additional iteration is required for testing the stabilization condition. These final labels are the canonical form of the graph and, therefore, two nodes with the same final label are automorphically equivalent. An example of running the algorithm in a simple graph is shown in Figure 2.

It can be noted that some pairs of the labels appearing in the same iteration of the Weisfeiler–Lehman algorithm are more similar than others. The automorphic distance between two nodes can be defined as the distance between their canonical labels. We propose a scheme to compute distances between the labels that are obtained by the Weisfeiler–Lehman algorithm. Since distances are only defined for labels appearing in the same iteration, a special label associated with nodes of degree 0, which we call the empty label $\ell_{\varnothing}$, is considered for convenience. Isolated nodes are directly assigned this label and left out of the iterative process.

Since labels created in the initial assignment are based on the node degree, we define the distance of labels of nodes x and y as the number of links that must be added to or removed from node x to transform it into node y. This can be easily computed as their absolute degree difference as follows:

$$d(\ell_0(x), \ell_0(y)) = |\deg(x) - \deg(y)|, \tag{2}$$

where $\ell_0(n)$ is the initially assigned label to node n. This definition of distance for initial labels is also valid for isolated nodes, with degree 0, which have been assigned the empty label.

Given these distances for initial labels, the distance between labels for the subsequent iterations can be computed as the distance of the optimally matched pairs of labels of their neighbors from the previous iteration. The distance of labels from the i-th iteration can be computed as

$$d(\ell_i(x), \ell_i(y)) = \min_{M_{i-1}(x,y)} \sum_{(u,v)\in M_{i-1}(x,y)} d(\ell_{i-1}(u), \ell_{i-1}(v)), \tag{3}$$

where $M_{i-1}(x, y)$ is the optimal assignment of neighbors of x to neighbors of y that minimizes the expression and, therefore, it is just the sum of distances between neighbors of x and y. If the neighborhood of one node is larger than the neighborhood of the other, leading to unmatched nodes, these nodes are directly matched with virtual nodes, which are labeled $\ell_{\varnothing}$. The distances of unmatched nodes to the empty label can be seen as the cost, in terms of distance, of inserting and transforming a virtual isolated node to obtain a

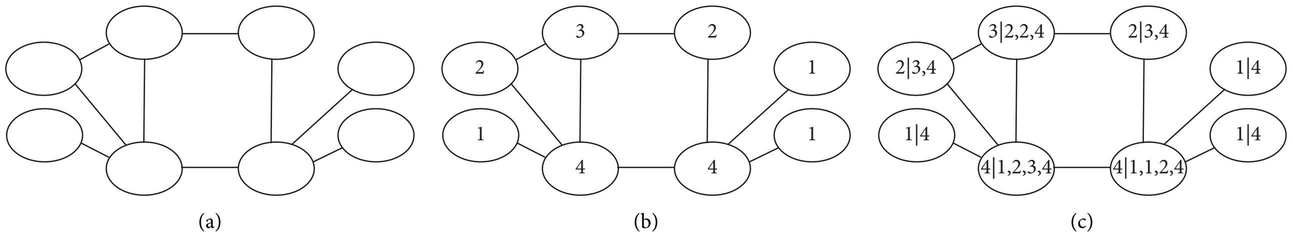


Figure 2: The Weisfeiler–Lehman canonicalization algorithm applied to a simple graph. (a) Original graph, (b) initial labeling, (c) first iteration, (d) first relabeling, (e) second iteration, and (f) second and final relabeling.

node with the label of the unmatched node. The optimal assignment, which would consider $O(n!)$ alternatives using a naive brute force approach, can be computed in polynomial time using the Hungarian algorithm [35].

Initially, equation (2) and, in subsequent iterations, equation (3) are used to compute a distance table. At any given time, only distances from two iterations need to be maintained: the distances currently being computed and the distances from the most recent previous iteration.

The described iterative process is carried out for each iteration of the 1-dimensional Weisfeiler–Lehman algorithm until label stabilization. The automorphic distance between a pair of nodes is defined as the distance between their canonical labels. It should be noted that labels can be represented using any set of symbols. However, for simplicity, we represent labels as positive integers.

The complete algorithm is shown in Algorithm 1. The function $\text{neighbors}(x)$ returns the set of labels of the neighbor nodes of x. The function $\text{sort}(s)$ sorts a set of elements. The ordering among elements is not relevant for the algorithm, but the same ordering must always be applied. The function $\text{concatenate}(x_1, \ldots, x_n)$ returns the concatenation of elements $x_1, \ldots, x_n$. Finally, the function $\text{unique}(s)$ generates and returns a unique symbol, such as an integer, for each observed unique string s, where $\text{unique}(s) = \text{unique}(s')$ if and only if $s = s'$.

3.1. Example.

3.1. Example. In this section, we show an illustrative example applying the proposed automorphic distance metric to the network shown in Figure 2(a).

The proposed algorithm for computing the automorphic distance initially assigns a label to each node according to its degree, as shown in Figure 2(b). Therefore, two nodes will have the same label if and only if they have the same degree. The initial distance table, represented as an upper triangular matrix due to the symmetry of distances, which will be proved in Section 3.2.3, is shown in Table 1. This distance table is computed using equation (2) according to the initial label assignments. For example, the distance between the labels 1 and 4 is 3, since this value is the absolute degree difference of the corresponding nodes.

After initialization, the algorithm enters into its main loop and performs its first iteration. For each node, the labels of its neighbors are ordered and concatenated with its own label, as shown in Figure 2(c). For example, the only node with label 3 has the associated string $3|2, 2, 4$, since its neighbors have labels 2, 4, and 2. These concatenated strings are replaced by a new label, chosen so that two nodes are assigned the same new label if and only if they had the same concatenated string. This process generates a new labeling as shown in Figure 2(d). Given these new labels, the algorithm computes their pairwise distances using equation (3), which are shown in Table 2.

For example, in order to compute the distance between labels 5 and 9, the optimal assignment between their neighbors minimizing the summation of distances, according to the previous iteration, must be obtained. In the previous iteration, 3 and 4 were the labels of the neighbors of nodes with label 5. Likewise, 1, 1, 2, and 4 were the labels of the neighbors of nodes with label 9. The Hungarian algorithm matches these neighbor labels to minimize their sum of distances: $(3, 2)$ and $(4, 4)$ according to Table 1. Since the two neighbor labels 1 of label 9 were left unmatched, they are both matched with the empty label as $(1, \ell_\varnothing)$. Given this optimal assignment, we can compute the distance between labels 5 and 9 as

$$\begin{aligned} d_1(5,9) &= d_0(3,2) + d_0(4,4) + d_0(1,\ell_\varnothing) + d_0(1,\ell_\varnothing) \\ &= 1 + 0 + 1 + 1 = 3. \end{aligned} \tag{4}$$

We can use the same algorithm to compute the distance between any other pair of labels. For example, to compute the distance between labels 6 and 7, we first identify their neighbor labels in the previous iteration: 2, 2, and 4 for label 6 and 4 for label 7. Using the Hungarian algorithm, both labels 4 are matched with each other and both labels 2 are

```
procedure Automorphic Distance
Input: Set of nodes $N$ of an undirected graph.
Output: Pairwise automorphic distances $d(x, y)$ for each pair of nodes
 $(x, y) \in N \times N$.
   for each $x$ in $N$ do
      $\ell_0(x) \longleftarrow \deg(x)$
   end for
   for each $x, y$ in $N \times N$ do
      $d(\ell_0(x), \ell_0(y)) \longleftarrow |\deg(x) - \deg(y)|$
   end for
   $i \longleftarrow 1$
   stabilized $\longleftarrow$ false
   while not stabilized do
      stabilized $\longleftarrow$ true
      $R \longleftarrow$
      for each $x$ in $N$ do
         $h_i(x) \longleftarrow$ concatenate(sort(neighbors($x$)))
         $c_i(x) \longleftarrow$ concatenate($\ell_{i-1}(x), h_i(x)$)
         $\ell_i(x) \longleftarrow$ unique($c_i(x)$)
         if $R \cdot$ contains($\ell_i(x)$) and not $R[\ell_i(x)] = \ell_{i-1}(x)$ then
            stabilized $\longleftarrow$ false
         else
            $R[\ell_i(x)] = \ell_{i-1}(x)$
         end if
      end for
      if not stabilized then
         for each $x, y$ in $N \times N$ do ▷ Using the Hungarian algorithm.
            $d(\ell_i(x), \ell_i(y)) \longleftarrow \min_{M_{i-1}(x,y)} \sum_{(u,v) \in M_{i-1}(x,y)} d(\ell_{i-1}(u), \ell_{i-1}(v))$
         end for
         $i \longleftarrow i + 1$
      end if
   end while
   for each $x, y$ in $N \times N$ do
      $d(x, y) \longleftarrow d(\ell_i(x), \ell_i(y))$
   end for
 end procedure
```

Algorithm 1: Automorphic distance algorithm.

Table 1: Initialization of the distance table.

ℓ/ℓ	$\ell_\varnothing$	1	2	3	4
$\ell_\varnothing$	0	1	2	3	4
1		0	1	2	3
2			0	1	2
3				0	1
4					0

Table 2: Distance table after the first relabeling.

ℓ/ℓ	$\ell_\varnothing$	5	6	7	8	9
$\ell_\varnothing$	0	7	8	4	10	8
5		0	3	3	3	3
6			0	4	4	2
7				0	6	4
8					0	2
9						0

matched with the empty label, since there are no remaining labels to match them with. Therefore, the distance between labels 6 and 7 is computed as follows:

$$d_1(6,7) = d_0(2, \ell_\varnothing) + d_0(2, \ell_\varnothing) + d_0(4,4) = 2 + 2 + 0 = 4. \qquad (5)$$

Following this iterative process, the algorithm performs a second iteration. Concatenated strings are computed as shown in Figure 2(e) and labels are updated as shown in Figure 2(f). It can be easily seen that labels have stabilized, obtaining the canonical labeling of this graph. The stabilization condition can be tested by performing an additional iteration and observing that nodes with label 13 are assigned the same label, while the other nodes are assigned an unique new label. This new labeling would be equivalent to the labeling obtained in the current iteration, the condition required to achieve stabilization. The pairwise distances computed in this iteration are shown in Table 3.

For instance, let us compute the distance between labels 11 and 16. We start by finding the optimal assignment that minimizes the pairwise distances of their neighbor labels in the previous iteration, which are 5, 5, and 8 for label 11 and 5, 7, 7, and 8 for label 16. The Hungarian algorithm obtains

Table 3: Distance table after the second and final relabeling.

ℓ/ℓ	$\ell_\varnothing$	10	11	12	13	14	15	16
$\ell_\varnothing$	0	18	24	16	8	10	27	25
10		0	10	2	12	8	13	11
11			0	12	16	14	13	7
12				0	8	10	11	13
13					0	2	19	17
14						0	21	15
15							0	6
16								0

the optimal matching $(5,5)$, $(5,7)$, and $(8,8)$, with an additional $(\ell_\varnothing, 7)$, due to the difference of the node degrees associated with labels 11 and 16. Once this optimal matching has been obtained, we can easily compute the distance between labels 11 and 16 using equation (3) as follows:

$$\begin{aligned} d_2(11,16) &= d_1(5,5) + d_1(5,7) + d_1(8,8) + d_1(\ell_\varnothing, 7) \\ &= 0 + 3 + 0 + 4 = 7. \end{aligned} \tag{6}$$

The automorphic distance between a pair of nodes is defined as the distance between their canonical labels, which are the final labels assigned in the iteration where stabilization is achieved. Therefore, in our example, the distance between nodes is given by Table 3. For example, we can see how nodes with canonical labels 13 and 14 are close to being automorphically equivalent, since their automorphic distance is only 2. In contrast, nodes with canonical labels 14 and 15 have a large automorphic distance, equal to 21, which indicates that they are far from being automorphically equivalent.

3.2. Metric Admissibility. In this section, we prove that the distance function that we have defined is a valid metric or distance function. In order to assert this statement, we must prove the following four conditions: nonnegativity, identity of indiscernibles, symmetry, and triangle inequality.

To prove these conditions, we note that equation (3) can be recursively decomposed as

$$\begin{aligned} d(\ell_i(x), \ell_i(y)) &= \min_{M_{i-1}(x,y)} \sum_{\substack{(u,v)\in \\ M_{i-1}(x,y)}} d(\ell_{i-1}(u), \ell_{i-1}(v)) \\ &= \min_{M_{i-1}(x,y)} \sum_{\substack{(u,v)\in \\ M_{i-1}(x,y)}} \min_{M_{i-2}(u,v)} \sum_{\substack{(u',v')\in \\ M_{i-2}(u,v)}} d(\ell_{i-2}(u'), \ell_{i-2}(v')) \\ &= \cdots \\ &= \sum_{(x',y')\in M'(x,y)} d(\ell_0(x'), \ell_0(y')) \\ &= \sum_{(x',y')\in M'(x,y)} \left|\deg(x') - \deg(y')\right|, \end{aligned} \tag{7}$$

where $M'(x,y)$ is the set of pairs of nodes that appear in the recursive summation at the deepest level of recursion as a result of choosing the optimal assignment in each iteration.

3.2.1. Proof of Nonnegativity. Nonnegativity requires that the distance function satisfies $d(\ell_i(x), \ell_i(y)) \geq 0$ for any possible pair of nodes x and y. Given the decomposition of our metric as shown in equation (7), it is straightforward to see that the summation of absolute values is guaranteed to be always equal to or greater than 0.

3.2.2. Proof of the Identity of Indiscernibles. The identity of indiscernibles implies that the distance function satisfies $d(\ell_i(x), \ell_i(y)) = 0$ if and only if $\ell_i(x) \equiv \ell_i(y)$. The Weisfeiler–Lehman algorithm guarantees that automorphically equivalent pairs of nodes are assigned the same canonical label, and nonautomorphically equivalent pairs of nodes are assigned different canonical labels.

Equation (7) is only equal to zero when $\deg(x') = \deg(y')$ for every pair of nodes in $M'(x,y)$. Two nodes can only have the same canonical label if they are automorphically equivalent, as guaranteed by the Weisfeiler–Lehman algorithm. If two nodes are assigned the same canonical label, their neighbors must have been assigned equivalent labels in all the iterations. Since the distance of a label to itself is 0, we can see that the summation yields 0, leading to a zero distance for nodes when $\ell_i(x) \equiv \ell_i(y)$.

On the other side, when two nodes have different canonical labels, their neighbors must have been assigned different labels during the execution of the algorithm. This implies that, in the recursive decomposition shown in equation (7), at least one pair of nodes will not match nodes with the same initial labels, leading to a distance greater than 0 for nodes $\ell_i(x) \not\equiv \ell_i(y)$.

3.2.3. Proof of Symmetry. The condition of symmetry requires that the proposed distance function must satisfy the property $d(\ell_i(x), \ell_i(y)) = d(\ell_i(y), \ell_i(x))$. We can prove that equation (7) is symmetric as

$$\begin{aligned} d(\ell_i(x), \ell_i(y)) &= \sum_{(x',y') \in M'(x,y)} \left|\deg(x') - \deg(y')\right| \\ &= \sum_{(y',x') \in M'(y,x)} \left|\deg(y') - \deg(x')\right| \\ &= d(\ell_i(y), \ell_i(x)), \end{aligned} \quad (8)$$

since $M'(x, y)$ is equal to $M'(y, x)$ when we swap the nodes. The order of the nodes in each pair does not affect the result of our distance function.

3.2.4. Proof of the Triangle Inequality. The triangle inequality requires that the inequality $d(\ell_i(x), \ell_i(y)) \leq d(\ell_i(x), \ell_i(z)) + d(\ell_i(z), \ell_i(y))$ is satisfied by the proposed automorphic distance function.

By the definition of the proposed distance, we know that

$$\begin{aligned} &d(\ell_i(x), \ell_i(y)) \leq d(\ell_i(x), \ell_i(z)) + d(\ell_i(z), \ell_i(y)) \Rightarrow \\ &\sum_{(a,b) \in M'(x,y)} |\deg(a) - \deg(b)| \\ &\leq \sum_{(a,c) \in M'(x,z)} |\deg(a) - \deg(c)| \\ &+ \sum_{(c,d) \in M'(z,y)} |\deg(c) - \deg(d)|. \end{aligned} \quad (9)$$

We know that the absolute value satisfies the triangle inequality and thus the lower value that the right side can take is

$$\sum_{(a,b) \in M'(x,y)} |\deg(a) - \deg(b)| \leq \sum_{(a,d) \in M''(x,y,z)} |\deg(a) - \deg(d)|, \quad (10)$$

where $M''(x, y, z)$ is the set of pairs resulting from chaining or combining $M'(x,z)$ and $M'(z,y)$ so that $(a,d) \in M''(x,y,z)$ if and only if, $(a,c) \in M'(x,z)$ and $(c,d) \in M'(z,y)$.

For this inequality to hold, it requires the nonexistence of a pairing of nodes better than the matching done in the left-hand side. Since the Hungarian algorithm ensures that matchings are optimal, minimizing their sum of distances, the matching at the right-hand side cannot be better than optimal and, therefore, the value of the right-hand side can only be equal to or greater than the value on the left-hand side, satisfying the triangle inequality condition.

3.3. Metric Computational Complexity. In this section, we analyze the time and spatial computational complexity of our proposed metric.

The initialization of labels based on the degree of nodes has $O(n)$ time and spatial complexity, where n is the number of nodes in the network. The computation of the table for the initial distances has $O(n^2)$ time and spatial complexity, since the distance is computed for every pair of nodes.

Each iteration of the algorithm requires computing the sorted list of labels of the neighbors for each node. This task can be accomplished for each node with computational and spatial complexity $O(k)$, where k is the degree of the node, when using radix, bucket, or counting sort. Thus, computing these strings for all nodes has $O(nk)$ time and spatial complexity. Renaming these labels can be done in $O(n)$ time using hash-based data structures. To compute the pairwise distances between labels in the current iteration, the Hungarian algorithm, with computational complexity $O(k^3)$, must be computed for each pair of nodes, leading to $O(n^2k^3)$ time complexity and $O(n^2)$ spatial complexity. Finally, checking if the labels have stabilized can be done in $O(n)$ using a hash-based index.

The number of iterations, m, required for 1-dimensional Weisfeiler–Lehman algorithm to converge is closely related to the diameter of the network [32]. Even though the number of iterations is theoretically bounded by n, it has been widely observed that real-world networks tend to exhibit the small-world phenomenon, presenting a small diameter [36, 37] and leading to a small number of iterations required for convergence.

Therefore, by combining these partial results, the total time complexity is $O(mn^2k^3)$, where n is the number of nodes in the graph, k is the degree of nodes, and m is the number of iterations required for convergence. The spatial complexity of the algorithm is $O(n^2)$, since only the distances and labels from the previous and the current iteration must be maintained at any given time.

Most of these steps can be easily parallelized, since most of them are independent of each node and are only based on the results from the previous iteration. For example, the initial labels for each node can be assigned independently. Once we have assigned these labels, the computation of their pairwise distances can be split among all the available processors, since they are independent. The iterative assignment of labels in each loop iteration can also be parallelized using a concurrent data structure to ensure that the new labels are properly generated. Furthermore, the pairwise distances between these new labels can be easily computed in a parallel way, since they are completely independent as they only rely on the distance table computed in the previous iteration.

3.4. A More Efficient Approximation. The quadratic time complexity with respect to the number of nodes in the network limits the applicability of our proposed approach in large networks. In this section, we introduce a more scalable approximation of our approach.

The proposed alternative approach is based on the idea of efficiently clustering similar labels just before computing distances between labels. This clustering only applies to the computation of distances, not to the steps related to the Weisfeiler–Lehman algorithm. Instead of requiring a distance matrix including all pairs of nodes, this approximated approach only requires computing the distance matrix between the obtained clusters. The distances between pairs of labels are approximated as the distance between the clusters to which they belong to. Therefore, using this approximation, the step where distances between pairs of labels are computed becomes quadratic with respect to the number of clusters, which are set by the experimenter and can be much lower than the number of nodes in the network.

The proposed approximation algorithm is shown in Algorithms 2–4. The base structure of the algorithm remains unchanged, but a new label clustering step is introduced just before computing the distances between pairs of labels. This clustering procedure is based on choosing a representative label for each cluster. A set for storing representative labels, initially containing only the empty label, is created. While the size of the set of representative labels is less than the number of clusters c specified as a parameter, a new representative label is added to this set as follows. A number of labels s is sampled from a set of remaining labels, which initially contains all the labels obtained except the empty label. The minimal distance to every representative label, i.e., the smallest distance to any label in the set of representative labels, is computed for each sampled label. The sampled label with the largest minimal distance is chosen as a representative label. Next, the chosen representative label is added to the set of representative labels and removed from the set of remaining labels. This greedy procedure approximates a set of representative labels with large distances among them.

Each representative label represents a cluster. A cluster is assigned to each nonrepresentative label by choosing the closest cluster according to the distance to its representative label.

The proposed approximation has a significant impact on our algorithm computational complexity. The spatial complexity becomes $O(c^2 + nk)$, where c is the number of clusters and can be set to $c \ll n$, since only distances between pairs of clusters obtained in the same iteration must be stored.

The time complexity of the proposed approximation can be obtained as follows. The steps, which are also present in the original algorithm, without the clustering step, now have a computational complexity $O(nk + mc^2k^3)$ as a result of computing only distances between pairs of clusters. The time complexity of the greedy clustering step is $O(sc^2k^3)$ for the step where representative labels are chosen, where s is the number of the sampled labels, and for the step where clusters are assigned to each nonrepresentative label is $O(nck^3)$. The greedy clustering procedure must be executed m times until the convergence of the Weisfeiler–Lehman algorithm. Therefore, the computational complexity introduced by the clustering procedure is $O(msc^2k^3 + mnck^3)$.

Given these complexities, the time complexity of the complete algorithm is $O(msc^2k^3 + mnck^3)$. Since c and s can be manually set to be much smaller than the number of nodes in the network, this approach is more scalable and can be applied to large networks.

4. Experimental Evaluation

This section contains our experimental evaluation of the proposed automorphic distance metric. In addition to the theoretical results described in the previous sections of the manuscript, we aim to show how our proposal properly captures the concept of node role based on automorphic equivalence and its different practical applications.

The experimental evaluation of the proposed distance metric is a complicated task due to the lack of a standard evaluation methodology for role-based metrics. In order to evaluate our distance metrics and compare the obtained results with other state-of-the-art approaches, we propose different experimental settings, which are described in the following sections.

4.1. Performance Evaluation of Distance Metrics. In this section, we evaluate and compare the proposed original automorphic distance metric with RoleSim considered as a distance metric. We follow an evaluation methodology based on the idea that nodes playing the same role in the network must show similar topological properties. Therefore, close nodes with regard to an automorphic distance metric should have similar topological properties, and their distance should be correlated with the similarity between these properties. For example, nodes with a similar role in the network should also exhibit a similar centrality, and similar nodes playing the role of bridges between different communities should exhibit similar topological properties measurable by different metrics proposed in the field of network analysis.

We propose measuring the performance of these distance metrics by evaluating how well these metrics capture global network structural properties. This can be done by computing the Pearson correlation coefficient between the distances given by the evaluated methods for a given pair of nodes and the absolute difference of their scores associated with a given network structural property, as follows:

$$\text{score} = \text{corr}\left(\left(d\left(\ell_i(x), \ell_i(y)\right): \quad \forall x, y \in N \times N\right), \left(|s(x) - s(y)|: \quad \forall x, y \in N \times N\right)\right), \tag{11}$$

```
procedure APPROXIMATED AUTOMORPHIC DISTANCE
Inputs:
-Set of nodes $N$ of an undirected graph.
-Number of clusters $c$.
-Number of samples $s$.
Outputs:
-Approximated pairwise automorphic distances $d(x, y)$ for each pair of clusters $(x, y) \in C_i \times C_i$.
-Mapping $g_i(x)$ from each label to its cluster representative.
  for each x in N do
    $\ell_0(x) \leftarrow \deg(x)$
  end for
  $C_0, g_0 \leftarrow$ GREEDY_LABEL_CLUSTERING$(N, c, s, 0)$
  for each $x, y$ in $C_0 \times C_0$ do
    $d(x, y) \leftarrow$ COMPUTE_DISTANCE$(x, y, 0, g_0)$
  end for
  $i \leftarrow 1$
  stabilized←false
  while not stabilize $d$ do
    stabilized←true
    $R \leftarrow$
    for each x in N do
      $h_i(x) \leftarrow$ concatenate(sort(neighbors$(x)$))
      $c_i(x) \leftarrow$ concatenate$(\ell_{i-1}(x), h_i(x))$
      $\ell_i(x) \leftarrow$ unique$(c_i(x))$
      If $R \cdot$ contains$(\ell_i(x))$ and not $R[\ell_i(x)] = \ell_{i-1}(x)$ then
        stabilized←false
      else
        $R[\ell_i(x)] \leftarrow \ell_{i-1}(x)$
      end if
    end for
    if not stabilized then
      $C_i, g_i \leftarrow$ GREEDY_LABEL_CLUSTERING$(N, c, s, i, g_{i-1})$
      for each $x, y$ in $C_i \times C_i$ do
        $d(x, y) \leftarrow$ COMPUTE_DISTANCE$(x, y, i, g_{i-1})$
      end for
      $i \leftarrow i + 1$
    end if
  end while
end procedure
```

Algorithm 2: Approximated automorphic distance algorithm.

```
procedure COMPUTE_DISTANCE
Inputs:
-Labels $x$ and $y$.
-Iteration $i$.
-Mapping from each label to its cluster representative $g$ (optional).
▷ The mapping is only used for $i > 0$.
Output: Computed distance $d$.
  If $i = 0$ then
    $d \leftarrow |x - y|$
  else ▷ Using the Hungarian algorithm.
    $d \leftarrow \min_{M_{i-1}(x,y)} \sum_{(u,v) \in M_{i-1}(x,y)} d(g(\ell_{i-1}(u)), g(\ell_{i-1}(v)))$
  end if
end procedure
```

Algorithm 3: Distance computation.

```
procedure GREEDY_LABEL_CLUSTERING
Inputs:
-Set of nodes $N$ of an undirected graph.
-Number of clusters $c$.
-Number of samples $s$.
-Iteration $i$.
-Mapping from each label to its cluster representative $g$ (optional).
▷ The mapping is only used for $i > 0$.
Outputs:
-Set of cluster representatives $C$.
-Mapping $g_n(x)$ from each label to its cluster representative.
  $C \longleftarrow \{\ell_\varnothing\}$
  $L \longleftarrow \{\ell_i(x)\colon\ \forall x \in N\}$
  while $|C| < c$ do
    $S \longleftarrow \text{sample}(L, s)$
    $r \longleftarrow \text{argmax}_{u\in S}\text{argmin}_{v\in C}\text{COMPUTE\_DISTANCE}(u, v, i, g)$
    $C.\text{add}(r)$
    $L.\text{remove}(r)$
end while
  $L' \longleftarrow \{\ell_i(x)\colon\ \forall x \in N\}$
  for each $l$ in $L'$ do
    $g_n(l) \longleftarrow \text{argmin}_{r\in C}\text{COMPUTE\_DISTANCE}(l, r, i, g)$
  end for
end procedure
```

Algorithm 4: Greedy label clustering.

where $\text{corr}(A, B)$ is the Pearson correlation between two sequences A and B, N is the set of nodes in the network, $d(\ell_i(x), \ell_i(y))$ is the distance between a pair of nodes x and y, and $s(n)$ is the score given by the network structural property for a node n.

In order to capture the most important role-related topological features, we included several network structural properties in our experimentation. First, we included PageRank [38], a well-known algorithm for measuring the importance of each node in a network. This iterative algorithm outputs a probability distribution that represents the likelihood of reaching each node by randomly moving through the network. Second, we considered closeness [39], which measures node centrality as the reciprocal of the sum of the shortest path lengths to all other nodes. Finally, we also included betweenness [40], which measures node centrality proportionally to the number of shortest paths passing through each node.

Our experimentation was carried out using five networks from very diverse domains. Due to the high computational complexity of the evaluated methods, we restricted this experiment to small networks with a few hundred nodes at most. The included networks are listed as follows:

(i) A communication network of 36 nodes and 62 links between employees in a sawmill [41]. Two nodes, each one representing an employee, are linked if they contacted a number of times higher than a given threshold.

(ii) A social network of bottlenose dolphins, with 62 nodes and 159 links, based on observations between 1994 and 2001 [42]. A link represents frequent association between dolphins, which are represented as nodes.

(iii) A network containing 77 nodes and 254 links of character coappearances in the novel "Les Misérables" [43], written by Victor Hugo. Each node represents a character and each link represents the coappearance of two characters in the same chapter.

(iv) A network, with 112 nodes and 425 links, of noun and adjective adjacencies for the novel "David Copperfield" [44], written by Charles Dickens. Each node represents a word that is a noun or an adjective, whereas a link represents adjacency of two words in the text.

(v) A network, composed of 131 nodes and 1074 links, representing the Brazilian air-traffic in 2016 [24], which was extracted from the National Civil Aviation Agency (ANAC). Nodes represent airports and links represent the existence of commercial flights between pairs of airports.

The results we obtained are shown in Table 4. The distances computed by the automorphic distance are highly correlated with the considered role-related structural properties, outperforming RoleSim in most cases. Except for the dolphin network, our distance shows a high correlation with PageRank and closeness. Although positive, indicating that it is capable of capturing part of this information, RoleSim clearly performs worse in terms of capturing these structural properties. Both distances present a lower correlation with betweenness, especially in the dolphin and Les Misérables networks, where RoleSim is not correlated at all.

These results suggest that our proposal outperforms RoleSim in capturing role-based topological information and that distances computed by our approach are more

Table 4: Pearson correlation coefficients illustrating the correlation of automorphic distance metrics and topological properties.

Network	Method	Page Rank	Closeness	Betweenness
Sawmill	Automorphic	**0.7987**	**0.7633**	**0.7856**
	RoleSim	0.3249	0.4858	0.1389
Dolphins	Automorphic	**0.7310**	0.4176	**0.2185**
	RoleSim	0.4476	**0.4229**	0.0032
Les misérables	Automorphic	**0.6976**	**0.6758**	**0.5205**
	RoleSim	0.2545	0.3197	0.0720
Copperfield	Automorphic	**0.9490**	**0.7636**	**0.8811**
	RoleSim	0.4480	0.6583	0.2493
Brazil airports	Automorphic	**0.8858**	**0.9272**	**0.5344**
	RoleSim	0.4271	0.5942	0.1328

The highest correlation for each network and each structural metric is marked in bold.

closely related to the roles of nodes. In the following section, we will evaluate our distance in the task of generating continuous representations for nodes based on their role.

4.2. Computing Role-Based Node Embeddings. A central problem in machine learning is finding representations that ease the visualization or extraction of useful information from data [45]. A common solution is the computation of embeddings that represent complex objects in a vector space preserving certain properties [46]. Node embedding, also known as graph embedding, is the task of mapping each node in a graph to a dimensional space trying to preserve the similarity or distance between pairs of nodes. Therefore, similar nodes will be located in similar regions of the space. Node embeddings have lately gained attention, since they have achieved good results in different machine learning tasks [46]. Several models have been proposed for node embedding. However, these techniques try to preserve the connectivity of the network by obtaining embeddings that preserve the neighborhood and the community of nodes [22, 47]. This information has proven to be useful due to the presence of homophily, also known as assortativity, in real-world networks [48], where entities tend to be connected to similar ones, a feature that allows us to explain certain features of the nodes. Even though the connectivity information captured by these techniques is relevant, these techniques fail to capture information related to the role or function of the nodes in the network, which is a highly valuable information that is complementary to the information obtained by locality-based embedding techniques.

Recently, different techniques for computing role-based embeddings, capturing the roles of nodes by placing nodes that play a similar function in the network close in the resulting vector space, have been proposed. In this work, we consider two state-of-the-art techniques that have received a lot of attention: node2vec [23] and struc2vec [24]. Both methods learn continuous representations capturing node structural equivalence and identity, respectively. These approaches are based on applying the skip-gram model [49], which was initially proposed as a model for natural language, over node contexts extracted from the network using biased random walks.

On the one hand, the node2vec method computes these contexts combining the breadth-first search (BFS), with the intention of reflecting structural equivalence, and depth-first search (DFS), with the intention of reflecting homophily. This behavior is achieved using two parameters: p and q. The return parameter p controls the likelihood of returning back to the previous node in the random walk. The in-out parameter q controls the tendency of moving to close or far nodes. If $q > 1$, the random walks are biased towards close nodes. Otherwise, the random walks are biased towards visiting more distant nodes. We used the node2vec implementation available at https://github.com/aditya-grover/node2vec.

On the other hand, the struc2vec method computes these contexts by inducing a hierarchy that captures different levels of information. This hierarchy is used to build a weighted multilayer network, with a layer for each level in the hierarchy, which is used to generate node contexts by performing biased random walks with the intention of reflecting structural identity. We used the implementation available at https://github.com/leoribeiro/struc2vec.

In order to compute node embeddings using role-based distances, we apply the classical multidimensional scaling (MDS) [50, 51] to the distance matrices.

The following sections are devoted to the detailed study of two particular cases, where we evaluate the embeddings obtained using the proposed definition of automorphic distance compared to the previously introduced approaches. For each case, we show how 2-dimensional embeddings can capture relevant information related to the function of the nodes in the network. These sections are intended to provide some insights into the properties of the obtained embeddings, whereas the numerical evaluation is left for the next section.

4.2.1. Zachary's Karate Club Network. Zachary's karate club network is a popular social network representing the 34 members of a university karate club as its nodes and their interactions outside the club, as its links [52]. During the study carried out by Zachary, a conflict arose between the two club administrators, leading to the split of the club into two groups according to the leader each member decided to follow. For this reason, this network has served as a prototypical case study for community detection algorithms and some network analysis techniques.

Zachary observed that the formed groups were highly homogeneous and assortative. Therefore, nodes tended to be connected to nodes that took the same decision. In this context, different embedding techniques have been proposed in the past for generating embeddings distributing nodes in multidimensional spaces according to the leader they decided to follow [47]. This information is crucial when the task is related to community detection. However, these embeddings fail, by nature, to reveal the role of each node in the network. In this study, we evaluate and compare the results obtained by the previously introduced techniques in the task of computing embeddings that are able to capture the role of each node in Zachary's karate club network.

We assigned a class to each node according to objective role-related properties. The two leaders are colored in red. Nodes interacting with the two leaders are colored in green. Nodes not interacting with any of the two leaders are colored in yellow. Finally, the remaining nodes, which interact with only one of the leaders, are colored in blue. Figure 3 shows the network drawn using the Fruchterman–Reingold force-directed layout algorithm [53] and the embeddings obtained by applying our automorphic distance, RoleSim, node2vec, and struc2vec.

Due to the large number of possible parameter combinations in node2vec and struc2vec, the authors decided to use the same values that were used by their original authors in case studies included in their respective papers. For struc2vec, we used the same parameters they used in their Section 4.2, where they also studied this network for a different task. Therefore, we set the number of walks per source to 5, the walk length to 15, and the skip-gram window size to 3. For node2vec, we used the same parameters that they used in their case study of the Les Misérables network with the intention of capturing structural equivalence, as shown in their Section 4.1. Therefore, we set $p = 1$ and $q = 2$. Since they did not provide the values of the other parameters, we used the same as those for struc2vec.

In the resulting embeddings, shown in Figure 3, it can be seen how all techniques seem to capture at least partial information about the function of the nodes. For example, all techniques place the two leaders, represented by the red nodes, in close positions. This observation also applies to green nodes, which represent persons who interact with both leaders. Furthermore, green nodes are highly clustered in all the embeddings.

However, both node2vec and struc2vec show a poor separation of the different classes. RoleSim seems to separate the classes better, but still places green nodes inside the cloud of blue nodes. In the three embeddings, the two leaders are placed pretty close to normal club members, without capturing their unique function in the network.

In contrast, it can be seen that the four roles are linearly separated in the embedding generated using our distance metric. In addition, it can be seen how the two leaders are clearly mapped as outliers and placed notably apart from the other nodes representing normal members of the club.

4.2.2. World Trade Network. In network data mining, homophily is commonly exploited in node classification tasks, since nodes tend to exhibit the same class as their neighbors [54]. Even though this situation occurs in a large number of networks from very diverse domains, homophily based classification techniques fail when the classes of the nodes are defined by the role they play in the network, instead of the community they belong to.

An illustrative example of this situation is a network containing data on trade of miscellaneous metal manufactures among 80 countries, according to data gathered in 1993 and 1994 from the Commodity Trade Statistics published by the United Nations [55, 56]. Each country is represented by a node in the network. Each commercial relationship is represented by an arc, which we consider an undirected edge in practice. In this case, arcs correspond to trading high technology products or heavy manufactures between countries.

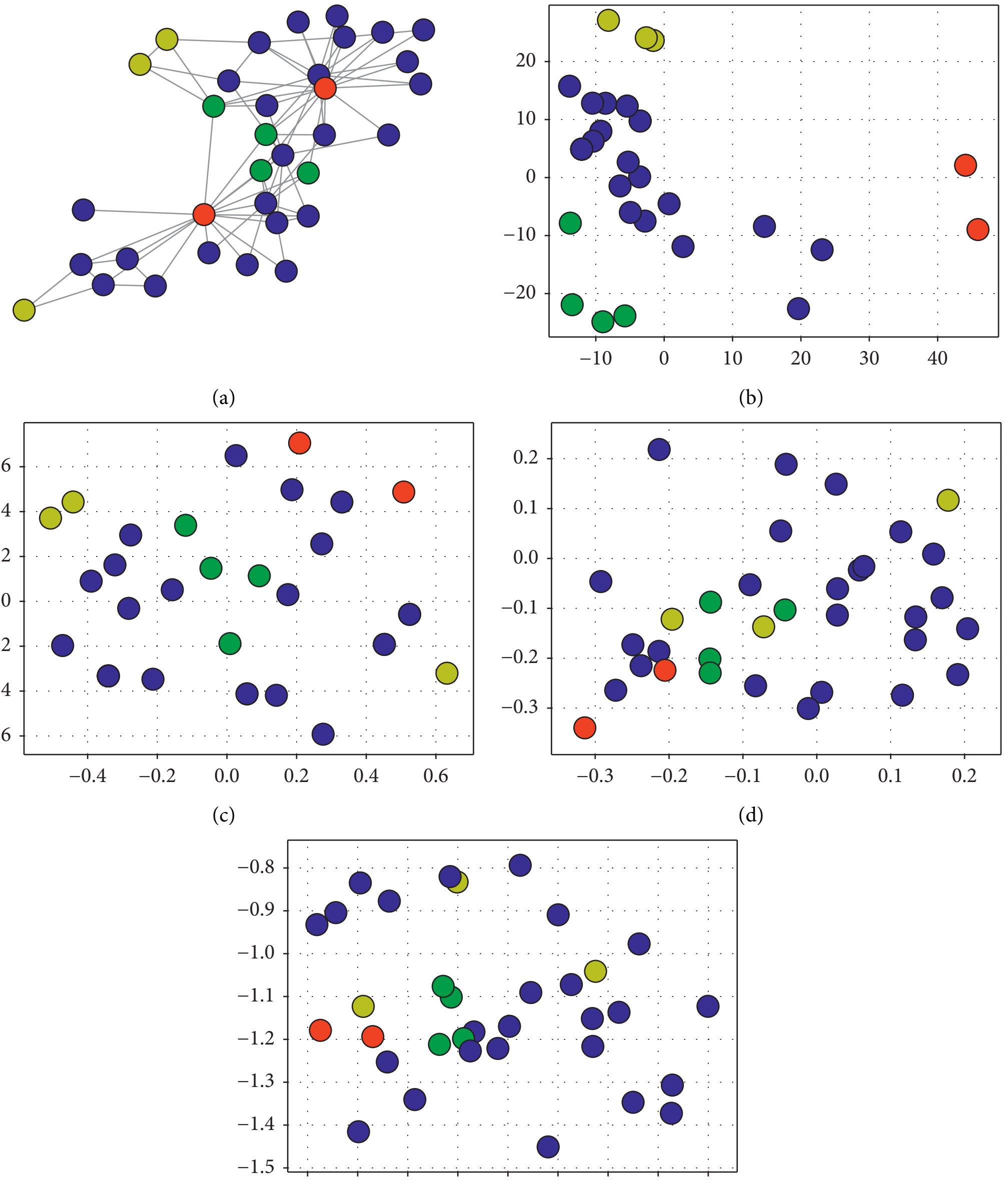
In addition, the authors of this dataset annotated countries in the network with their structural world economic position in 1994. World economic positions are a classification of countries in the context of the world-system theory that explains some complex dynamics observed in the real-world [57]. This classification splits countries into three possible categories: core countries (colored in green), semiperiphery countries (colored in blue), and periphery countries (colored in red). In short, core countries have a high economic, military, and political power, which allows them to control the world economic system. The periphery is composed of less developed countries, owning a disproportionately small share of global wealth. Finally, the semiperiphery consists of countries that do not clearly fall in the previous two categories and exhibit a more intermediate status.

In this case, for struc2vec, we used the same parameters their authors used in the analysis carried out in Section 4.1 of their manuscript. Therefore, we set the number of walks per source to 20, the walk length to 80, and the skip-gram window size to 5. For node2vec, we also used these parameters with $p = 1$ and $q = 2$.

Figure 4 shows the trade network drawn using the Fruchterman–Reingold force-directed layout algorithm and the embeddings obtained by applying different techniques. Again, all techniques seem to capture role-related information as shown by the elongated shape of all the embeddings, curved in the case of RoleSim, with core and periphery countries placed at both ends.

It can be seen that the automorphic distance, RoleSim, and struc2vec achieve a good separation of core and periphery countries, with some overlapping with countries in the semiperiphery. However, struc2vec shows a highly elongated shape, also present in the case studies included in their manuscript. Finally, node2vec seems to perform worse than the other approaches. In spite of being able to separate the peripheral countries, it does not perform well in the separation of the central countries and those in the semiperiphery.

4.3. Performance Evaluation of Computed Node Embeddings. The case studies analyzed in the previous section allowed us to gain some insight about the embeddings generated by our



(a)

(b)

(c)

(d)

(e)

Figure 3: Zachary's karate club network and its node embeddings with node role coloring. (a) Force-directed layout, (b) automorphic embedding, (c) RoleSim embedding, (d) node2vec embedding, and (e) struc2vec embedding.

distance metric in comparison to other state-of-the-art approaches. However, in order to perform a more objective and exhaustive evaluation, we extend this analysis by carrying out a quantitative performance evaluation in this section.

As previously stated, the evaluation of techniques for role discovery is a complicated task due to the lack of available evaluation datasets with role-based ground-truth data. This difficulty extends to the evaluation of techniques for generating embeddings capturing the notion of the node role. The most common solution is indirectly evaluating these embeddings based on their application to other tasks with evaluable or measurable performance metrics. For example, these embeddings can be evaluated by means of a classification problem, where labels related to node roles are predicted using the generated embeddings. However, there is no guarantee that the labels are completely determined by the roles of nodes in the network. Therefore, these experiments only evaluate which techniques capture the driving factors of these labels better, and in which node roles may only partially contribute or not contribute at all. This criticism extends to other tasks previously used to evaluate role-based embeddings, such as link prediction.

In order to overcome these limitations while offering an objective evaluation procedure, we propose a new approach to evaluate techniques for computing role-based embeddings for nodes. Our proposal consists of measuring how well the embeddings produced by these techniques can reconstruct global network structural properties based on

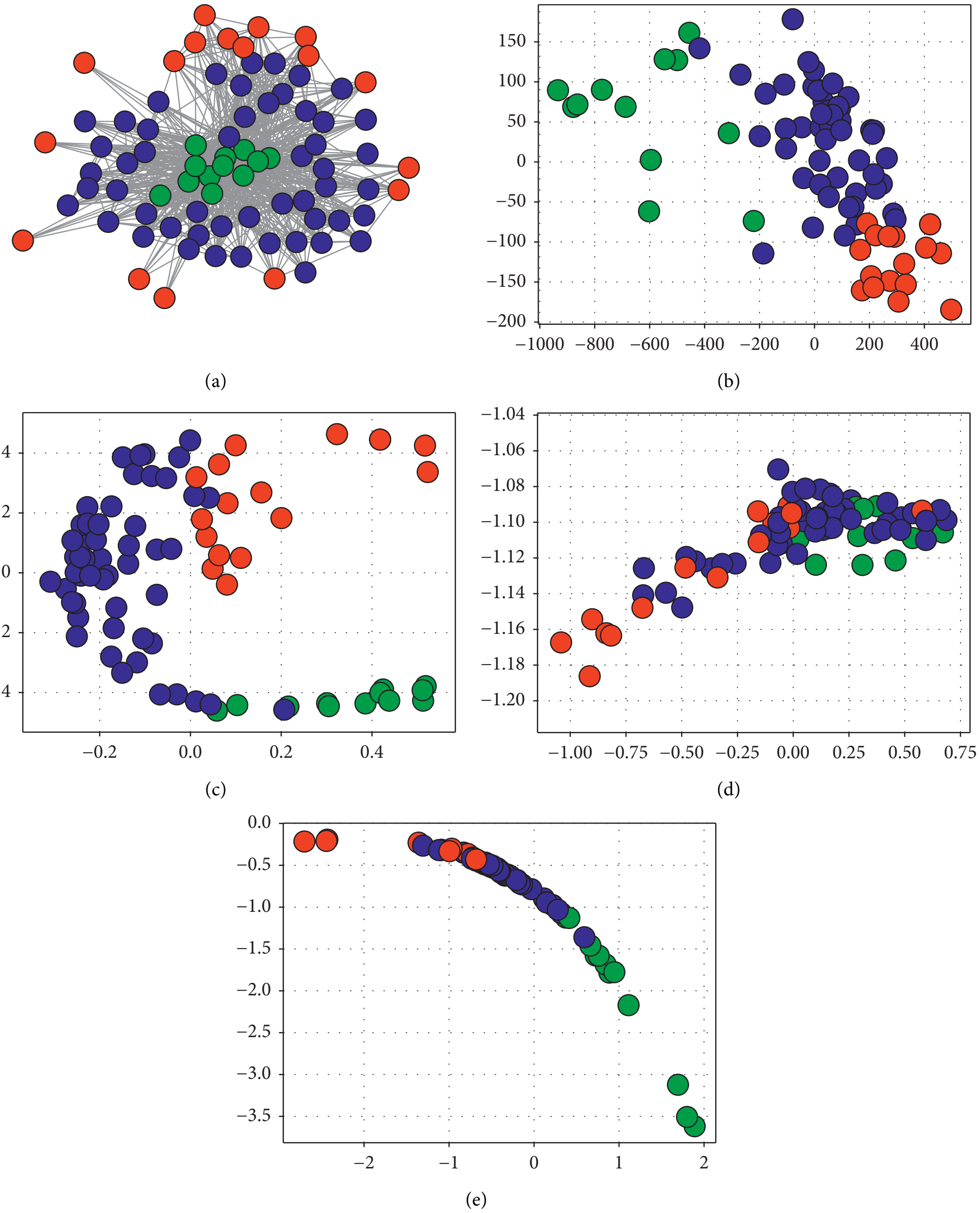


Figure 4: World trade network and its corresponding node embeddings. (a) Force-directed layout, (b) automorphic embedding, (c) RoleSim embedding, (d) node2vec embedding, and (e) struc2vec embedding.

the same arguments we used in Section 4.1. Nodes with a similar role in the network also exhibit similar topological properties. Therefore, evaluating how well these embeddings can be used to reconstruct these topological properties is an indirect way of assessing the quality of these embeddings.

In our experimentation, we considered the network structural properties previously described in Section 4.1. For each method and each property, we trained a linear regression model using the ordinary least squares method. For a given node, these trained linear models take its embedding as input and output the predicted score for the node corresponding to the structural property. We carried out our experimentation using five-dimensional embeddings. We chose this number of dimensions as a reasonable number, enough to capture complex role-related patterns.

In order to carry out an unbiased evaluation, these models were trained using a five-fold crossvalidation, with 80% of the nodes as training set and the remaining 20% as the test set. In addition to crossvalidation, since all the evaluated methods include an element of randomness, we ran five different executions of the evaluation procedure using different random seeds, both for the generation of the embeddings and the crossvalidation procedure. The root mean squared error (RMSE) was used as the evaluation

Table 5: Obtained RMSE for the considered methods, networks, and structural network properties.

	Method	PageRank	Closeness	Betweenness
EU airports	Automorphic	$\mathbf{7.67 \times 10^{-4}}$	$\mathbf{2.34 \times 10^{-2}}$	$\mathbf{7.18 \times 10^{-3}}$
	struc2vec	1.66×10^{-3}	4.15×10^{-2}	8.23×10^{-3}
	node2vec	2.39×10^{-3}	5.58×10^{-2}	9.38×10^{-3}
USA airports	Automorphic	$\mathbf{6.41 \times 10^{-4}}$	$\mathbf{3.59 \times 10^{-2}}$	$\mathbf{7.81 \times 10^{-3}}$
	struc2vec	9.66×10^{-4}	4.69×10^{-2}	7.99×10^{-3}
	node2vec	1.10×10^{-3}	4.43×10^{-2}	8.28×10^{-3}
EU-email	Automorphic	$\mathbf{2.72 \times 10^{-4}}$	5.36×10^{-2}	$\mathbf{2.63 \times 10^{-3}}$
	struc2vec	7.38×10^{-4}	6.45×10^{-2}	3.83×10^{-3}
	node2vec	9.16×10^{-4}	$\mathbf{4.95 \times 10^{-2}}$	4.15×10^{-3}
Univ-email	Automorphic	$\mathbf{2.92 \times 10^{-4}}$	$\mathbf{2.00 \times 10^{-2}}$	$\mathbf{2.63 \times 10^{-3}}$
	struc2vec	5.23×10^{-4}	2.49×10^{-2}	3.73×10^{-3}
	node2vec	6.58×10^{-4}	2.83×10^{-2}	4.18×10^{-3}
Roget	Automorphic	$\mathbf{2.68 \times 10^{-4}}$	3.17×10^{-2}	$\mathbf{2.52 \times 10^{-3}}$
	struc2vec	3.70×10^{-4}	3.62×10^{-2}	3.27×10^{-3}
	node2vec	4.68×10^{-4}	$\mathbf{1.74 \times 10^{-2}}$	3.67×10^{-3}

The lowest RMSE for each dataset and structural network property is marked in bold.

metric. The reported error for a method on a given dataset is the average RMSE obtained in the multiple executions of the five-fold crossvalidation using different seeds.

In this experiment, we included the approximated version of our proposal, node2vec, and struc2vec. RoleSim was left out due to its high computational complexity, which limits its applicability to large networks.

Methods included in the comparison have different parameters that must be adjusted. Due to the high computing requirements for executing each technique multiple times, we limited our experimentation to a restricted set of parameters. For our approximation of the automorphic distance metric, we chose 30 clusters and samples as parameters, since it is a reasonable value considering the size of the studied networks, which are presented below.

For node2vec and struc2vec, we limited our experimentation to different parameter settings included in their original papers. For node2vec, we used 0.50, 1, and 2 as values for the p parameter. The q parameter was also set to 0.50, 1, and 2. In order to avoid running a large number of parameter combinations, which is computationally intensive, we fixed the remaining parameters to the default values used in the original paper: number of walks per source to 10, length of walk per source to 80, and skip-gram window size to 10. For struc2vec, we report the best results using the following parameter combinations. We tested 10 and 20 walks per source with lengths 15 and 80. For the skip-gram model, we used window sizes of 5 and 10. The algorithm was executed with all the optimizations enabled: *OPT1*, *OPT2*, and *OPT3*.

Five different networks were used in our experimentation:

(i) A network, composed of 399 nodes and 5995 links, representing European air-traffic in 2016 [24], built using data from the Statistical Office of the European Union (Eurostat). Nodes representing airports and links represent the existence of commercial flights between pairs of airports.

(ii) Another network, with 1190 nodes and 13599 links, representing air-traffic in 2016 from the United States of America [24]. The network was built using data from the Bureau of Transportation Statistics.

(iii) A communication network, with 1005 nodes and 16706 links, based on mailing data from a large European research institution [58]. Each node represents a person, and each link represents that two nodes shared at least one email.

(iv) Another communication network, composed of 1133 nodes and 5451 links, built using mailing data from the University Rovira i Virgili in Tarragona [59], located in Spain. Each node represents an user, and each link represents that at least one email was sent between the two users.

(v) A network, with 1010 nodes and 3649 links, representing the thesaurus written by Peter Mark Roget [43]. Each node represents a category and each link represents a relation between a pair of categories.

The results obtained for this evaluation procedure are shown in Table 5. The best-performing parameters for struc2vec and node2vec were omitted for the sake of clarity. Note that the absolute error is small in all cases due to the order of magnitude of the scores computed using the included network properties. These results show how the embeddings computed using the approximation of the proposed automorphic distance metric achieve a lower RMSE than the other approaches in 13 out of 15 cases. Only node2vec outperforms our approach in two cases for closeness, where it performs especially well compared to the other approaches.

These results suggest that our proposal can capture node-related information better than other state-of-the-art approaches, as shown by the lower error achieved in the task of reconstructing topological properties.

5. Conclusions

In this paper, we have proposed a novel distance metric for nodes that relaxes the strict concept of automorphic equivalence. To the best of our knowledge, this is the first work to propose a consistent nonnormalized distance metric that captures the concept of automorphic equivalence without approximating it using feature engineering. In addition, we have shown that the proposed distance function is a valid distance metric by proving the required conditions. Furthermore, we have shown how our metric can be exploited to generate node embeddings that capture role information. Finally, we have carried out different experiments in order to show how our proposal can outperform other state-of-the-art techniques in different role-related tasks.

Our proposal creates new opportunities in problems related to role discovery. Future work includes exploiting our distance metrics in problems related to anomaly detection in networks and transfer learning based on roles shared by nodes across different networks.

Data Availability

The network datasets used in our experiments are from publicly available studies, which have been cited in our paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by the Spanish Ministry of Economy and the European Regional Development Fund (FEDER), under grant TIN2012-36951, and the program "Ayudas para contratos predoctorales para la formación de doc 2013," under grant BES-2013-064699. This work was also partially supported by the project "BIGDATAMED: Analisis de datos en Medicina, de las historias clínicas al BIGDATA" with references B-TIC-145-UGR18 and P18-RT-1765.

References

[1] R. A. Rossi and N. K. Ahmed, "Role discovery in networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 4, pp. 1112–1131, 2015.

[2] J. J. Luczkovich, S. P. Borgatti, J. C. Johnson, and M. G. Everett, "Defining and measuring trophic role similarity in food webs using regular equivalence," *Journal of Theoretical Biology*, vol. 220, no. 3, pp. 303–321, 2003.

[3] E. M. Hafner-Burton, M. Kahler, and A. H. Montgomery, "Network analysis for international relations," *International Organization*, vol. 63, no. 3, pp. 559–592, 2009.

[4] P. Holme and M. Huss, "Role-similarity based functional prediction in networked systems: application to the yeast proteome," *Journal of The Royal Society Interface*, vol. 2, no. 4, pp. 327–333, 2005.

[5] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3, pp. 75–174, 2010.

[6] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 80, no. 5, Article ID 056117, 2009.

[7] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, 2012.

[8] R. A. Rossi, B. Gallagher, J. Neville, and K. Henderson, "Modeling dynamic behavior in large evolving graphs," in *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, pp. 667–676, ACM, Rome, Italy, February 2013.

[9] K. Henderson, B. Gallagher, T. Eliassi-Rad et al., "RolX: structural role extraction & mining in large graphs," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1231–1239, Beijing, China, August 2012.

[10] E. P. Xing, W. Fu, and L. Song, "A state-space mixed membership blockmodel for dynamic network tomography," *Annals of Applied Statistics*, vol. 4, no. 2, pp. 535–566, 2010.

[11] R. S. Burt, "Detecting role equivalence," *Social Networks*, vol. 12, no. 1, pp. 83–97, 1990.

[12] H. C. White, S. A. Boorman, and R. L. Breiger, "Social structure from multiple networks. I. Blockmodels of roles and positions," *American Journal of Sociology*, vol. 81, no. 4, pp. 730–780, 1976.

[13] F. Lorrain and H. C. White, "Structural equivalence of individuals in social networks," *Journal of Mathematical Sociology*, vol. 1, no. 1, pp. 49–80, 1971.

[14] L. D. Sailer, "Structural equivalence: meaning and definition, computation and application," *Social Networks*, vol. 1, no. 1, pp. 73–90, 1978.

[15] S. P. Borgatti and M. G. Everett, "Notions of position in social network analysis," *Sociological Methodology*, vol. 22, pp. 1–35, 1992.

[16] N. E. Friedkin and E. C. Johnsen, "Social positions in influence networks," *Social Networks*, vol. 19, no. 3, pp. 209–222, 1997.

[17] P. E. Pattison, "Network models: some comments on papers in this special issue," *Social Networks*, vol. 10, no. 4, pp. 383–411, 1988.

[18] M. G. Everett and S. P. Borgatti, "Regular equivalence: general theory," *Journal of Mathematical Sociology*, vol. 19, no. 1, pp. 29–52, 1994.

[19] P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: a survey," 2017, http://arxiv.org/abs/1705.02801.

[20] J. Liu, Z. He, L. Wei, and Y. Huang, "Content to node: self-translation network embedding," in *Proceedings of the 24th ACM SIGKDD International Conference On Knowledge Discovery & Data Mining*, pp. 1794–1802, London, UK, August 2018.

[21] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: large-scale information network embedding," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 1067–1077, Florence, Italy, May 2015.

[22] V. W. Zheng, S. Cavallari, H. Cai, K. C.-C. Chang, and E. Cambria, "From node embedding to community embedding," 2016, http://arxiv.org/abs/1610.09950.

[23] A. Grover and J. Leskovec, "node2vec: scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864, ACM, San Francisco, CA, USA, August 2016.

[24] L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, "struc2vec: learning node representations from structural identity," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 385–394, Halifax, Canada, August 2017.

[25] G. Jeh and J. Widom, "SimRank: a measure of structural-context similarity," in *Proceedings of the 8th ACM SIGKDD International Conference On Knowledge Discovery And Data Mining*, pp. 538–543, ACM, Edmonton Alberta, Canada, July 2002.

[26] M. R. Hamedani and S.-W. Kim, "SimRank and its variants in academic literature data: measures and evaluation," in *Proceedings of the 31st Annual ACM Symposium on Applied Computing*, pp. 1102–1107, ACM, Pisa, Italy, April 2016.

[27] Z. Lin, M. R. Lyu, and I. King, "PageSim: a novel link-based measure of web page similarity," in *Proceedings of the 15th International Conference on the World Wide Web*, pp. 1019-1020, ACM, Edinburgh, UK, May 2006.

[28] E. A. Leicht, P. Holme, and M. E. Newman, "Vertex similarity in networks," *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 73, no. 2, Article ID 026120, 2006.

[29] R. Jin, V. E. Lee, and L. Li, "Scalable and axiomatic ranking of network role similarity," *ACM Transactions on Knowledge Discovery from Data*, vol. 8, no. 1, 3 pages, 2014.

[30] L. Li, L. Qian, V. E. Lee, M. Leng, M. Chen, and X. Chen, "Fast and accurate computation of role similarity via vertex centrality," in *Proceedings of the International Conference on Web-Age Information Management*, pp. 123–134, Springer International Publishing, Qingdao, China, June 2015.

[31] R. Jin, V. E. Lee, and H. Hong, "Axiomatic ranking of network role similarity," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 922–930, ACM, San Diego, CA, USA, August 2011.

[32] M. Fürer, "Weisfeiler-Lehman refinement requires at least a linear number of iterations," in *Proceedings of the 28th International Colloquium on Automata, Languages, and Programming*, pp. 322–333, Berlin, Germany, July 2001.

[33] B. Weisfeiler and A. Lehman, "A reduction of a graph to a canonical form and an algebra arising during this reduction," *Nauchno-Technicheskaya Informatsia*, vol. 2, no. 9, pp. 12–16, 1968.

[34] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-Lehman graph kernels," *Journal of Machine Learning Research*, vol. 12, pp. 2539–2561, 2011.

[35] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics*, vol. 2, no. 1-2, pp. 83–97, 1955.

[36] J. Travers and S. Milgram, "The small–world problem," *Phychology Today*, vol. 1, pp. 61–67, 1967.

[37] D. J. Watts and S. H. Strogatz, "Collective dynamics of "small-world" networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[38] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: bringing order to the web," Technical report, Stanford University Press, Redwood City, CA, USA, 1999.

[39] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1978.

[40] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[41] J. H. Michael and J. G. Massey, "Modeling the communication network in a sawmill," *Forest Products Journal*, vol. 47, no. 9, 25 pages, 1997.

[42] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, "The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations," *Behavioral Ecology and Sociobiology*, vol. 54, no. 4, pp. 396–405, 2003.

[43] D. E. Knuth, *The Stanford GraphBase-A Platform For Combinatorial Computing*, ACM, New York, NY, USA, 1993.

[44] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 74, no. 3, Article ID 036104, 2006.

[45] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[46] Y. Goldberg and O. Levy, "word2vec explained: deriving Mikolov et al.'s negative-sampling word-embedding method," 2014, http://arxiv.org/abs/1402.3722.

[47] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: online learning of social representations," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 701–710, ACM, New York, NY, USA, August 2014.

[48] M. McPherson, L. Smith-Lovin, and J. M. Cook, "Birds of a feather: homophily in social networks," *Annual Review of Sociology*, vol. 27, no. 1, pp. 415–444, 2001.

[49] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of the International Conference on Learning Representations, ICLR 2013*, Scottsdale, AZ, USA, May 2013.

[50] I. Borg and P. J. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, Springer Science & Business Media, Berlin, Germany, 2005.

[51] F. Wickelmaier, *An Introduction to MDS*, 46 pages, Aalborg University, Aalborg, Denmark, 2003.

[52] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, no. 4, pp. 452–473, 1977.

[53] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Software: Practice and Experience*, vol. 21, no. 11, pp. 1129–1164, 1991.

[54] S. Bhagat, G. Cormode, and S. Muthukrishnan, "Node classification in social networks," *Social Network Data Analytics*, Springer, Berlin, Germany, pp. 115–148, 2011.

[55] W. D. Nooy, A. Mrvar, and V. Batagelj, *Exploratory Social Network Analysis with Pajek*, Cambridge University Press, Cambridge, UK, 2011.

[56] D. A. Smith and D. R. White, "Structure and dynamics of the global Economy: network analysis of international trade 1965–1980," *Social Forces*, vol. 70, no. 4, pp. 857–893, 1992.

[57] D. Chirot and T. D. Hall, "World-system theory," *Annual Review of Sociology*, vol. 8, no. 1, pp. 81–106, 1982.

[58] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graph evolution," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, 2 pages, 2007.

[59] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Physical Review E, Statistical, Nonlinear, and Soft Matter Physics*, vol. 68, no. 6, Article ID 065103, 2003.