Embedding Nodes via their Local Network Topology for Identifying Roles in Networks

Willie Neiswanger
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
willie@cs.cmu.edu

Abstract

This paper proposes a method for characterizing the local network topology in the vicinity of each vertex in a graph. The method provides a Euclidean embedding of each node based on its surrounding network structure. We show how descriptors of the local topologies around vertices can be used to quantify the types of local structure present in a network and identify groups of nodes that have similar roles in relation to their neighbors. Our method for embedding makes use of the Fixed Node Edit Distance (FNED), a distance we define between the local topologies around nodes. We present algorithms for computing the FNED, show how it can be used to embed the local topologies of nodes in Euclidean space, and demonstrate the embedding on multiple publicly available network datasets. Results show the ability of this method to characterize diverse regions of network structure and identify groups of nodes with similar roles.

1 Introduction

The topological structure in the vicinity of a node in a network contains information about the role of the node in relation to its neighbors. For example, in the context of social networks, the local topology around a person includes links to friends, as well as the links between friends or between friends of friends. Networks may contain groups of nodes that have similar local topologies; identifying these groups sheds insight into the types of roles that exist in a network.

Node degree can be viewed as a simple descriptor of local topology. However, it does not capture the potentially complex relations among the immediate neighbors of a node, nor does it incorporate information about the network structure greater than one step away. Additionally, as the degree of a node grows, so does the amount of possible relations among its neighbors. In order to capture the increasing complexity of the local topology, we desire a richer descriptor of the local edge structure that is able to encode the local topology as its complexity scales with node degree.

To this end, we develop a method for embedding nodes in Euclidean space based on local network structure (note that a map from each node into $\mathbb{R}^k$ can be viewed as a k-dimensional descriptor of the local topology). In this paper, we present algorithms to accomplish this embedding through the use of a metric that defines a distance between the subgraphs of a network. The subgraphs capture local network structure around a node, and the metric allows a k-dimensional descriptor to be constructed for each node by finding its distance to a set of other nodes. After defining this metric and providing algorithms for its computation, we demonstrate the method on a number of network datasets and show its ability to find groups of nodes that have similar roles in relation to their surroundings in a graph.

2 Definitions

2.1 k-Step Local Topology of a Node

Let $G = \{V, E\}$ be a graph with a set of vertices V and edges E. We would like to capture the network topology around a given node in the graph. We define the k-step local topology around a node n as follows: let $G' = \{V', E'\}$ be the subgraph of G traversed in k steps of breadth-first search starting from n (where the zeroth step yields $V' = \{n\}$, the first step yields $V' = \{n\} \cup \{\text{all nodes adjacent to } n\}$, and so on). Let $\tilde{E} = E' \cup \{\text{all edges between any two nodes in } V'\}$. Then we define the k-step local topology $T_k(n)$ to be

$$T_k(n) = \{V, \tilde{E}\} \tag{1}$$

Hence $T_k(n)$ is a subgraph of G containing all of the nodes and a subset of the edges from G. We illustrate the k-step local topology in Figure 1, which shows the 1-step and 2-step local topologies around two nodes in a graph.

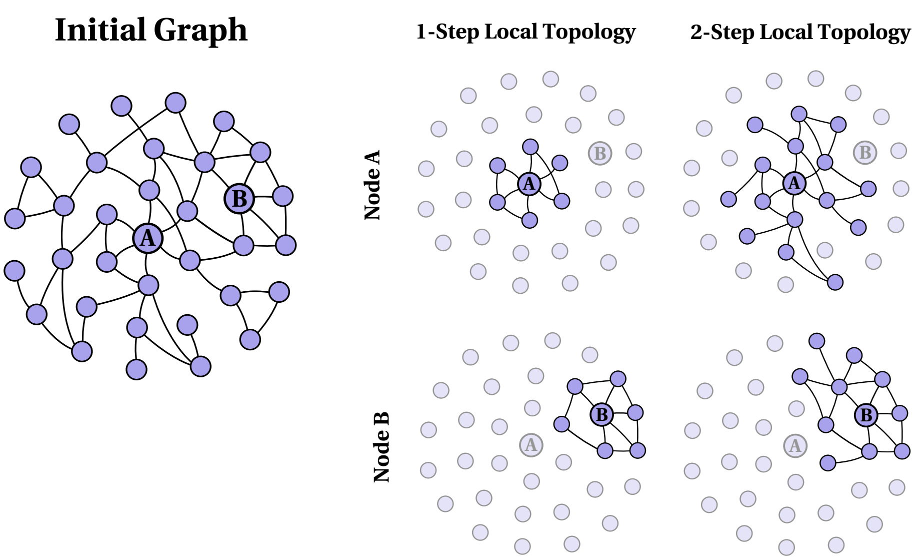

Figure 1: Illustration of the 1-step and 2-step local topologies for two nodes in an undirected graph.

2.2 Fixed Node Edit Distance (FNED)

Our first step towards embedding involves defining a distance between the local topologies around nodes; we describe how these distances are used for the Euclidean embedding in Section 3. The concept of edit distance has been shown to provide an intuitive way to represent distances between abstract structures [6, 5, 1]. We aim to define an edit distance between the local topologies around nodes in a graph. We call this the Fixed Node Edit Distance (FNED), and define it to be the minimum number of edge insertions or deletions to transform the local topology around one node into the local topology around another node.

An equivalent definition of the FNED, which we refer to as the mapping-overlap formulation, allows for easier computation. Intuitively, for a given mapping between the nodes of a pair of local topologies, we can find the number of edges that "do not overlap" (i.e. given the mapping, the number of node-pairs that are adjacent in one of the two local topologies but not adjacent in the other). The mapping that yields the minimum number of non-overlapping edges is equivalent to the FNED defined in terms of minimal edit distance.

Formally, for a graph $G = \{V, E\}$, let Ω_V be the set of bijections from V to itself (i.e. the set of permutations of V). Additionally, let $\Omega_{V,a\to b}$ be Ω_V restricted to bijections where node $a \in V$ is mapped to node $b \in V$. For all vertices $a, b \in V$, we define the FNED between a and b (given the two k-step local topologies $T_k(a)$ and $T_k(b)$), to be

$$\text{FNED}_k(a,b) = \min\Big|_{m \in \Omega_{V,a \to b}} d(T_k(a), T_k(b), m) \tag{2}$$

where

$$\begin{aligned} d(T_k(a),&T_k(b), m) = \\ &\sum_{i=1}^{n} \sum_{j=i}^{n} \Big[A_{T_k(a)}(m(i), m(j)) \oplus A_{T_k(b)}(i,j) \Big] \end{aligned} \tag{3}$$

where A_G denotes the adjacency matrix of graph G. Given adjacency matrices A_{G_1} and A_{G_2}, both of size $n \times n$, we define the XOR operator $\oplus$ to be

$$A_{G_1}(i_1, j_1) \oplus A_{G_2}(i_2, j_2) = \begin{cases} 1 \text{ if } A_{G_1}(i_1, j_1) = A_{G_2}(i_2, j_2) \\ 0 \text{ otherwise} \end{cases} \tag{4}$$

We illustrate the two definitions of the FNED and show an example of the distance between two nodes in Figure 2. In this figure, for the nodes A and B given in Figure 1, we depict the FNED as a minimal sequence of edits, and as an mapping that yields the minimum number of non-overlapping edges.

This paper is focuses only on undirected graphs, through it can be extended to directed and other labelled graphs by modifying the allowable edge edit operations.

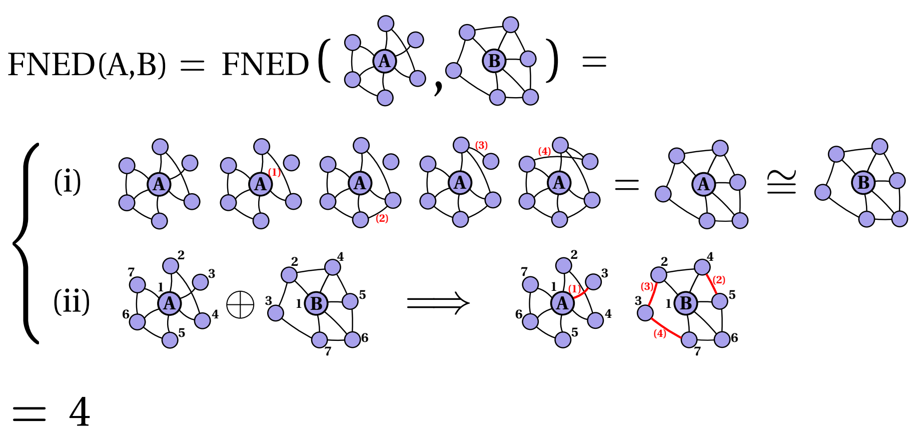

Figure 2: Illustration of the FNED between two local topologies. (i) The FNED viewed as a minimal sequence of edge edits. (ii) The FNED viewed as a minimal set of non-overlapping edges (red edges) over all possible mappings.

3 Algorithms for Embedding

The algorithms in this section describe the process of embedding the nodes of a graph in Euclidean space based on their local topology. Algorithm 1 gives an overview of this process. The embedding into Euclidean space is wholly dependent upon computation of the FNED between the local topologies of the nodes. The algorithms in the following sections provide both exact and approximate methods for computing the FNED, given certain graph types.

Algorithm 1 Node Embedding via Local Topologies

Input:
(i) An undirected graph, $G = \{V, E\}$.
(ii) A local topology step size, k.
(iii) A set of basis nodes, $B \subset V$.
for each $v \in V$ **do**
  for each $b \in B$ **do**
    Set Embedding(v, b) = FNED$_k(v, b)$
  end for
end for
Output: Embedding, the feature matrix where each row is a node, and the set of columns represents the embedding $\in \mathbb{R}^B$.

3.1 Deterministic FNED for Trees

In Algorithm 2 we give a polynomial time algorithm for computing the FNED between nodes in a tree for 2-step local topologies. We will give approximate algorithms for computing the FNED between nodes in an arbitrary graph in Section 3.2, as the FNED is, in general, not possible to compute exactly in polynomial time.

Algorithm 2 Deterministic FNED for Trees

1: **Input:**
(i) An undirected graph, $G = \{V, E\}$ without loops.
(ii) Two nodes: $A, B \in V$.
2: Compute $T_2(A)$ and $T_2(B)$, the local topologies of A and B.
3: $N = \max(|T_2(A)|, |T_2(B)|)$
4: Add $N - \min(|T_2(A)|, |T_2(B)|)$ disconnected "virtual nodes" to the local topology with less nodes.
5: **for** each $a \in T_2(A)$ adjacent to A **do**
6:   **for** each $b \in T_2(B)$ adjacent to B **do**
7:     Find $n_a = \#\{a' | a' \text{ adjacent to } a \text{ and } a' \neq A\}$
8:     Find $n_b = \#\{b' | b' \text{ adjacent to } b \text{ and } b' \neq B\}$
9:     Set CostMatrix$(a, b) = |n_a - n_b|$
10:   **end for**
11: **end for**
12: OptMap = HungarianAlgorithm(CostMatrix)
13: FNED$_k(A, B) = \sum_{i=1}^{|T_2(A)|}$ XOR$(i, \text{OptMap}(i))$
14: **Output:** FNED$_k(A, B)$, the fixed node edit distance between A and B for k-step local topologies.
15: Note 1: HungarianAlgorithm() returns an optimal mapping between nodes adjacent to A and nodes adjacent to B given CostMatrix.
16: Note 2: XOR(i,j) returns $|n_i - n_j|$ (defined on lines 7 and 8) for $i \in T_2(A)$ and $j \in T_2(B)$.

3.2 Approximate FNED for Arbitrary Graphs

Finding an optimal graph matching is not possible in polynomial time for arbitrary graphs. In this section we provide a Markov Chain Monte Carlo (MCMC)-based algorithm to search the combinatorial space of mappings between the nodes of two local topologies. The algorithm follows a directed random walk through the space of permutations of nodes in a local topology in order to minimize the FNED (adhering to the mapping-overlap formulation of the FNED).

3.2.1 MCMC for Computing the FNED

Here, we formulate computation of the FNED between nodes as a stochastic combinatorial optimization problem. In Algorithm 3 we describe an MCMC algorithm similar to Metropolis-Hastings for sampling the FNED between two nodes in an arbitrary graph.

Algorithm 3 MCMC for FNED in Arbitrary Graphs

```
Input:
  (i) An undirected graph, G = {V, E}.
  (ii) A local topology step size, k.
  (iii) Two nodes: A, B ∈ V
  (iv) Maximum sampling iteration, NumIter.
Compute T_k(A) and T_k(B), the k-step local topologies of A and B.
N = max(|T_k(A)|, |T_k(B)|)
Add N − min(|T_k(A)|, |T_k(B)|) disconnected (degree 0) nodes to the local topology with fewer
  nodes.
Initialize a mapping m between nodes in T_k(A) and nodes in T_k(B)
for i = 1 : NumIter do
    m' = m
    Randomly choose 3 nodes ∈ T_k(a), and randomly permute their mapping in m'
    if d(T_k(a), T_k(b), m') < d(T_k(a), T_k(b), m) then
        Set m = m'
        Record d(T_k(a), T_k(b), m)
    else
        if rand() < d(T_k(a), T_k(b), m) / d(T_k(a), T_k(b), m') then
            Set m = m'
            Record d(T_k(a), T_k(b), m)
        end if
    end if
end for
FNED_k(A, B) = the minimum d(T_k(a), T_k(b), m) over m through all iterations.
Output: FNED_k(A, B), the fixed node edit distance between A and B for k-step local topolo-
  gies.
Note 1: d(T_k(a), T_k(b), m) is defined in Section 2.
Note 2: rand() returns a uniformly distributed random number ∈ (0, 1)
```

4 Experiments

4.1 Demonstration of Method on Small Social Networks

We apply our method to two well studied social network datasets. The first shows friendships among members of a karate club at a US university [8], and the second consists of associations among a collection of dolphins [2].

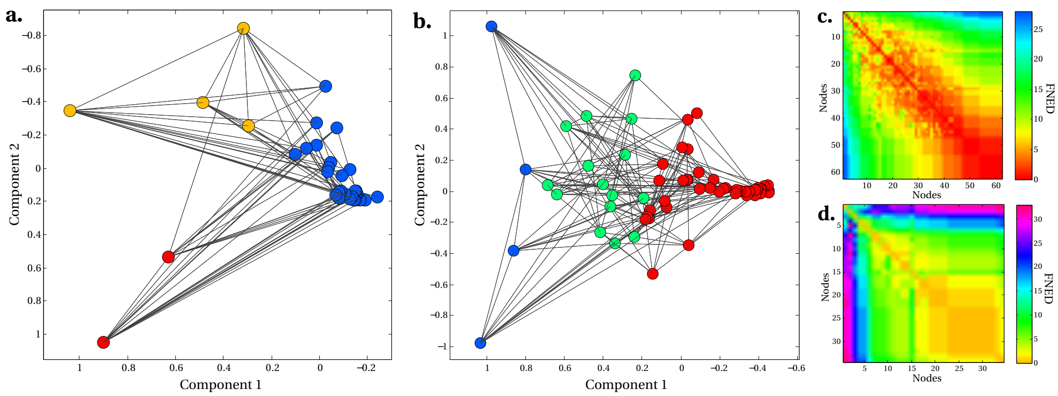

Figure 3: Results of embedding and role-identification for two social networks. (a) The first two MDS components of each embedded node in the karate club social network dataset and (b) dolphin social network dataset. The network edges are shown and node colors represent the results of clustering on the embedded values. (c) A heatmap encoding the Euclidean embedding for nodes in the dolphin social network dataset and (d) karate club social network dataset.

Embedding was performed using the MCMC algorithm to compute the FNED between 1-step local topologies for both datasets. After embedding, the k-means algorithm was carried out to cluster the nodes into groups with similar local topologies. The k-means algorithm was initialized at ten clusters, and converged to three clusters in both datasets. The entities in both networks were partitioned based on their social patterns, with the largest cluster containing a collection of low-degree nodes, and the other two clusters splitting the high-degree nodes into groups based on the connectivity between their adjacent nodes.

To visualize the embedding, multidimensional scaling (MDS) was applied to reduce the dimension of embedded nodes. Figure 3 plots the first two MDS components, where node color denotes the clustering results. In both plots, the node degree tends to decrease when moving from left to right across the x-axis. Nodes with a low degree tend to cluster together and those with a high degree tend to spread further apart, due to the potential for increased topological complexity around higher-degree nodes. Figure 3 also displays the embedding for all nodes in each graph as a heatmap. Heatmaps are useful for visualizing the embedded nodes (the i^{th} row/column represents the i^{th} node's vector embedding) and identifying groups of points with similar local topologies.

4.2 The Topology Among Teams in College Football Conferences

The datset in this experiment consists of a network of college football teams [3]. Edges connect pairs of teams if they are scheduled to play each other. This dataset provides a label for each team representing its conference (out of 12 possible conferences).

In this experiment, instead of clustering the embedded points, we color each according to its conference association. We aim to assess whether teams in a given conference have similar local topologies (we'd expect this to be the case, as there exist per-conference policies regarding the number of games played against in-conference and out-of-conference opponents).

Embedding was performed using the MCMC algorithm to compute the FNED between 1-step local topologies. Figure 4 shows the first two MDS components of the Euclidean embedding. This figure also shows the correspondence between each node and its conference label. We find that the

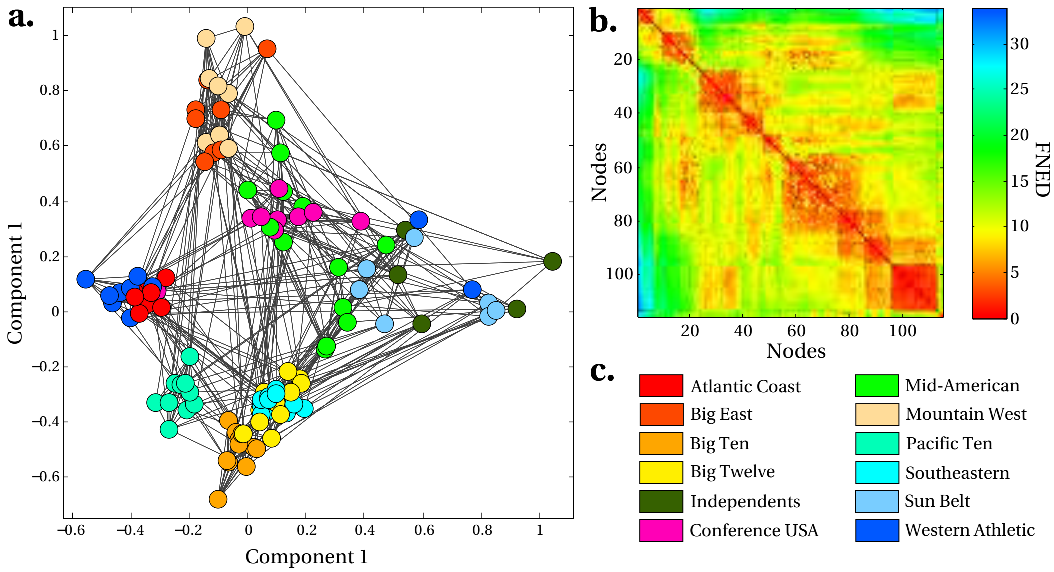

Figure 4: A network of college football teams. (a) The first two MDS components of each embedded node in this network, with edges shown, and node colors corresponding with the each team's conference (as listed in (c)). Teams belonging to the same conference are embedded similarly. (c) A heatmap encoding the Euclidean embedding of nodes in this network, where clusters of nodes with similar local topologies are visible along the diagonal.

embedding often places football teams from the same conference at similar points in space. This is reflected in the heatmap displaying the embedding of all nodes, where we see clear clusters along the diagonal. Figure 4 allows us to see which conferences have teams with very similar (e.g. Pacific Ten) or more varied (e.g. Mid-American) local topologies, which conferences are similar to others in terms of the local topologies of their teams, and in a couple cases, which teams have a local topology distinct from the others in their conference.

4.3 Role Discovery in Larger Networks and Comparisons with Node Degree

In this experiment, we hope to demonstrate the ability of our method to aid in the discovery of groups of nodes with similar roles in larger networks, and explicitly compare our descriptor of the local topology around a node with the node's degree. We apply our method two two networks: a neural network of the nematode *C. elegans* [4, 7], and a coauthorship network of network scientists [3].

Embedding was performed using the MCMC algorithm to compute the FNED between 1-step local topologies. Figure 5 shows the first two MDS components of the Euclidean embedding for both datasets, as well as the heatmaps displaying the the embedding of all nodes. In both heatmaps (especially for the network scientists heatmap) it is possible to see groups of nodes with similar local topologies. Clustering is performed in both datasets using the k-means algorithm. This algorithm was initialized at fifteen clusters, and converged to four clusters in the neural network dataset and to five clusters in the network scientists dataset. The color of each node corresponds with its cluster assignment. For the network author dataset, to provide some context for the embedded nodes, we display the names of the authors in the cluster with the highest average degree.

We'd like to compare our descriptor of local network structure with the node's degree, which we can view as a simple, one-dimensional descriptor of the local topology around a node. In Figure 5, we plot the first MDS component of the Euclidean embedding of each node against its degree. These plots give some illustration of the relationship between our descriptors of a node's local topology and its degree. They show how a one-dimensional representation of the embedding of a node's local topology correlates strongly with its degree, but can vary greatly as the node degree becomes large;

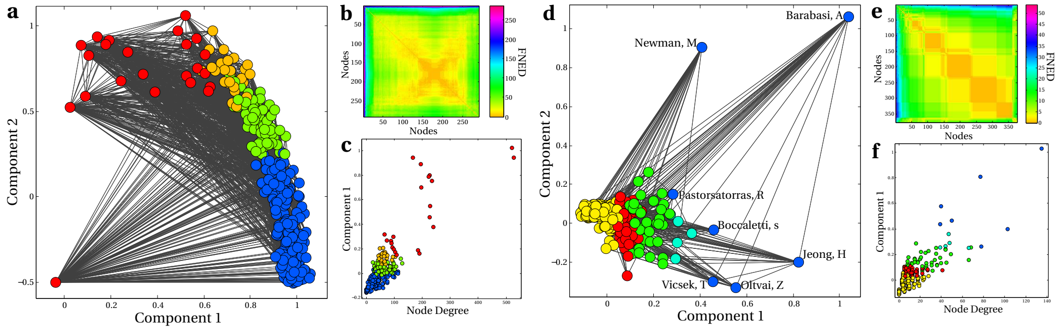

Figure 5: (a) The first two MDS components of each embedded node for a neural network of *C. elegans* and (d) a coauthorship network of network scientists, where the network edges are shown and node colors correspond to clustering results. (b) A heatmap encoding the Euclidean embedding of nodes in the neural network and (e) the network scientists network, where clusters of nodes with similar local topologies are visible along the diagonal. (c) A plot of the first MDS component of each embedded node vs its degree for the neural network and (f) the network scientists network, where our descriptor of local topology can be seen to vary as the degree becomes large.

it also gives evidence for the idea that a large amount of information may be gleaned from rich local network structure in vicinity of a node.

5 Conclusion

We have introduced a method for representing the local topology around a node in Euclidean space by defining the k-step local topology of a node and fixed node edit distance (FNED) between a pair of nodes. We have provided algorithms for both exact and approximate computation of the FNED for nodes in different types of graphs, and have demonstrated the embedding on five publically available datsets. Our demonstrations have shown that this Euclidean embedding of a node's local network structure can be used for discovering groups of nodes that have similar relations to the surrounding network.

References

[1] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis Applications*, 13:113–129, 2010. ISSN 1433-7541.

[2] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[3] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74:036104, Sep 2006.

[4] N. Pujol, E.M. Link, L.X. Liu, C.L. Kurz, G. Alloing, M.W. Tan, K.P. Ray, R. Solari, C.D. Johnson, and J.J. Ewbank. A reverse genetic analysis of components of the toll signaling pathway in¡ i¿ caenorhabditis elegans¡/i¿. *Current Biology*, 11(11):809–821, 2001.

[5] D. C. Reis, P. B. Golgher, A. S. Silva, and A. F. Laender. Automatic web news extraction using tree edit distance. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 502–511, New York, NY, USA, 2004. ACM. ISBN 1-58113-844-X.

[6] Kaspar Riesen and Horst Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Image Vision Comput.*, 27(7):950–959, jun 2009. ISSN 0262-8856.

[7] DJ Watts and SH Strogatz. Collective dynamics of'small-world'networks. *Nature*, 393(6684):440–442, 1998.

[8] W.W. Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473, 1977.