Embedding Nodes via their Local Network Topology for use in Machine Learning

Willie Neiswanger

Abstract

This paper proposes a method for characterizing the local network topology in the vicinity of each node in a graph. Our method provides a Euclidean embedding of each node based on its local topology. We show how features denoting the local topologies of vertices can be used beneficially in machine learning tasks involving datasets with relations between elements. Our method for embedding makes use of the Fixed Node Edit Distance (FNED), a distance metric between the local topologies of nodes, which we define in this paper. We give multiple algorithms for computing the FNED, show how it is used for embedding the local topologies of nodes, and demonstrate the embedding on multiple publicly available network datasets.

1 Definitions

1.1 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

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

We illustrate the k-step local topology around nodes in a graph in Figure 1.

1.2 Fixed Node Edit Distance (FNED)

Our first step towards embedding involves defining a distance between the local topologies of nodes; we describe how the distances between nodes can be used for Euclidean embedding in Section 2. The concept of edit distance has been shown to provide an intuitive way to represent distances between abstract structures [5, 4, 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 (between nodes) 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 previously (proof omitted).

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 k-step local topologies denoted $T_k(a)$ and $T_k(b)$, respectively), 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

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

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.

This project is currently concerned only with undirected graphs, with future plans to extend it to directed and other labelled graphs.

2 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 topologies.

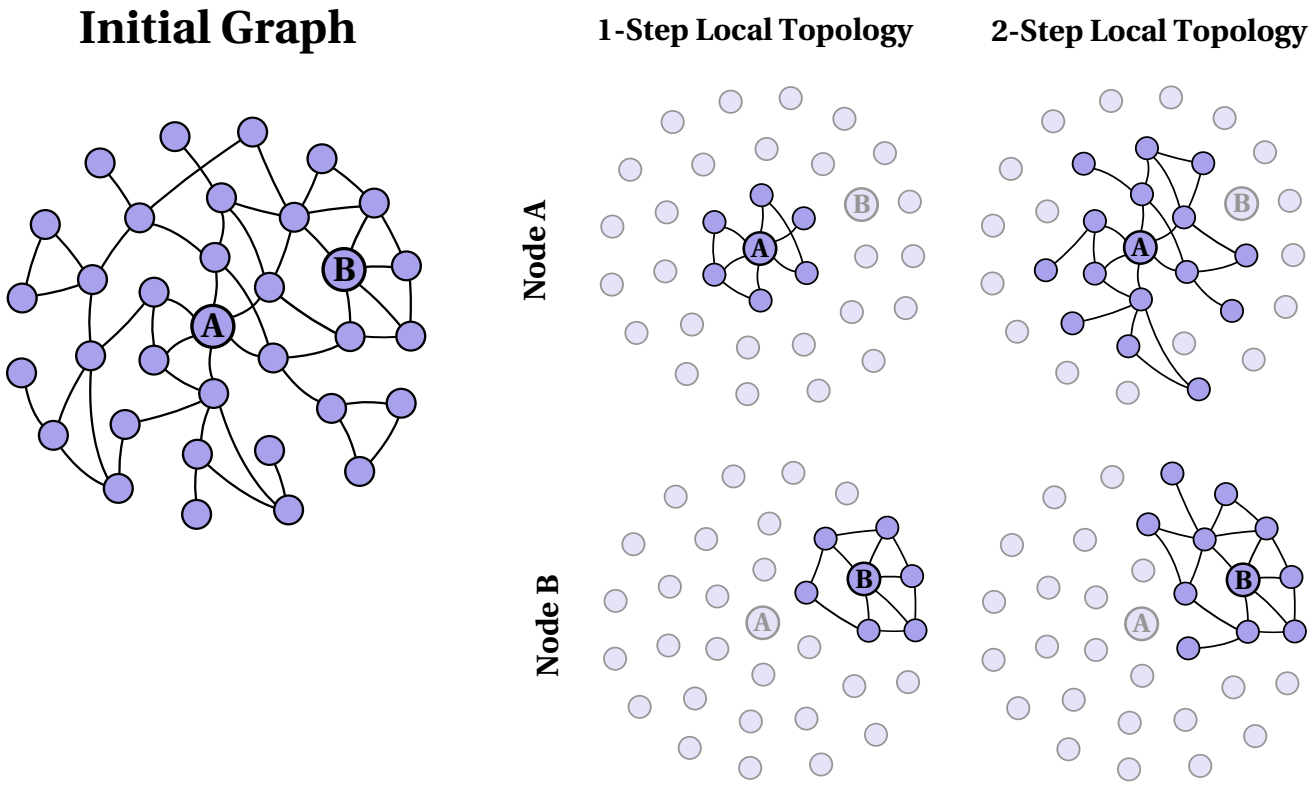

Figure 1: Illustration showing the 1-step and 2-step local topologies of two nodes in an undirected graph.

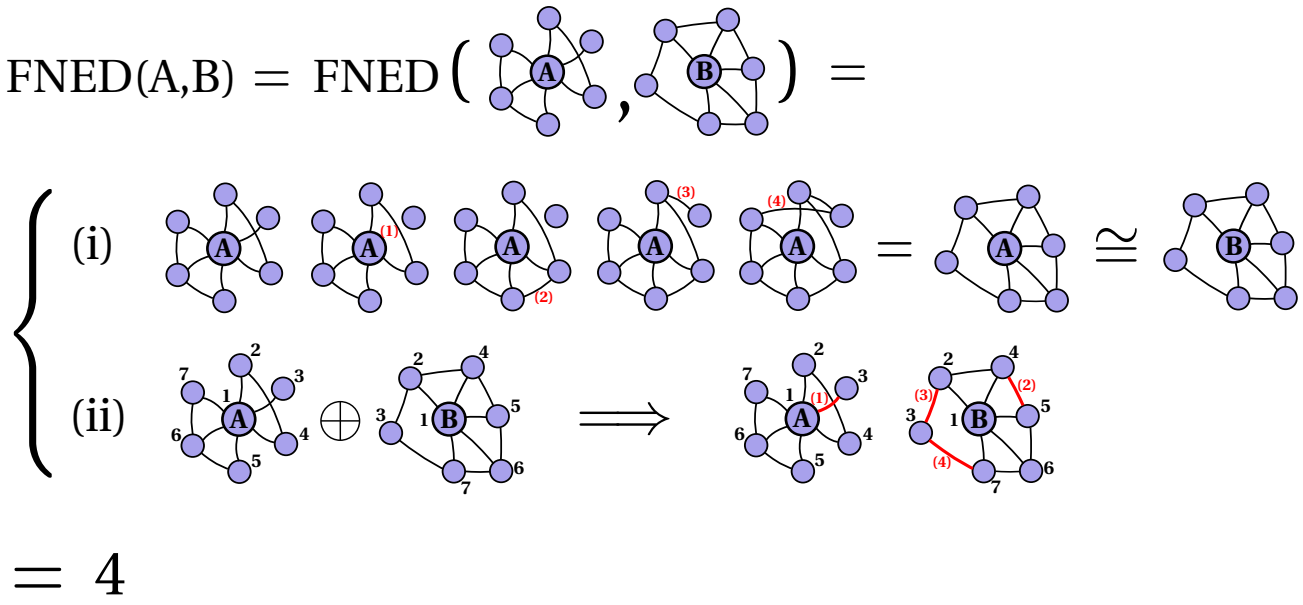

Figure 2: Illustration of the FNED between two local topologies. (i) shows the FNED viewed as a sequence of edge edits, while (ii) shows an equivalent definition of the FNED (the mapping-overlap formulation) as a collection of edges that "do not overlap" for a given mapping.

2.1 Overview for Embedding Nodes

Algorithm 1 gives an overview of the process of embedding in Euclidean space each node in a graph, based on its local topology.

Algorithm 1 Node Embedding via Local Topologies

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

The embedding into Euclidean space is wholly dependent upon computation of the FNED between the local topologies of the nodes. The following algorithms provide different methods for computing the FNED.

2.2 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.

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 $n_a = \#\{a' | a'$ adjacent to a and $a' \neq A\}$
8: Find $n_b = \#\{b' | b'$ adjacent to b and $b' \neq B\}$
9: Set CostMatrix$(a, b) = |n_a - n_b|$
10: **end for**
11: **end for**
12: OptMap = HungarianAlgorithm(CostMatrix)
13: $\text{FNED}_k(A, B) = \sum_{i=1}^{|T_2(A)|} \text{XOR}(i, \text{OptMap}(i))$
14: **Output:** $\text{FNED}_k(A, B)$, 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 $j \in T_2(B)$.

2.3 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.

2.3.1 MCMC for FNED Computation

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

1: **Input:**
   (i) An undirected graph, $G = \{V, E\}$.
   (ii) A local topology step size, k.
   (iii) Two nodes: $A, B \in V$
   (iv) Maximum sampling iteration, NumIter.
2: Compute $T_k(A)$ and $T_k(B)$, the k-step local topologies of A and B.
3: $N = \max(|T_k(A)|, |T_k(B)|)$
4: Add $N - \min(|T_k(A)|, |T_k(B)|)$ disconnected (degree 0) nodes to the local topology with fewer nodes.
5: Initialize a mapping m between nodes in $T_k(A)$ and nodes in $T_k(B)$
6: **for** $i = 1 : \text{NumIter}$ **do**
7: $m' = m$
8: Randomly choose 3 nodes $\in T_k(a)$, and randomly permute their mapping in m'
9: **if** $d(T_k(a), T_k(b), m') < d(T_k(a), T_k(b), m)$ **then**
10: Set $m = m'$
11: Record $d(T_k(a), T_k(b), m)$
12: **else**
13: **if** rand() $< \frac{d(T_k(a), T_k(b), m)}{d(T_k(a), T_k(b), m')}$ **then**
14: Set $m = m'$
15: Record $d(T_k(a), T_k(b), m)$
16: **end if**
17: **end if**
18: **end for**
19: $\text{FNED}_k(A, B)$ = the minimum $d(T_k(a), T_k(b), m)$ over m through all iterations.
20: **Output:** $\text{FNED}_k(A, B)$, the fixed node edit distance between A and B for k-step local topologies.
21: Note 1: $d(T_k(a), T_k(b), m)$ is defined in Section 1.
22: Note 2: rand() returns a uniformly distributed random number $\in (0, 1)$

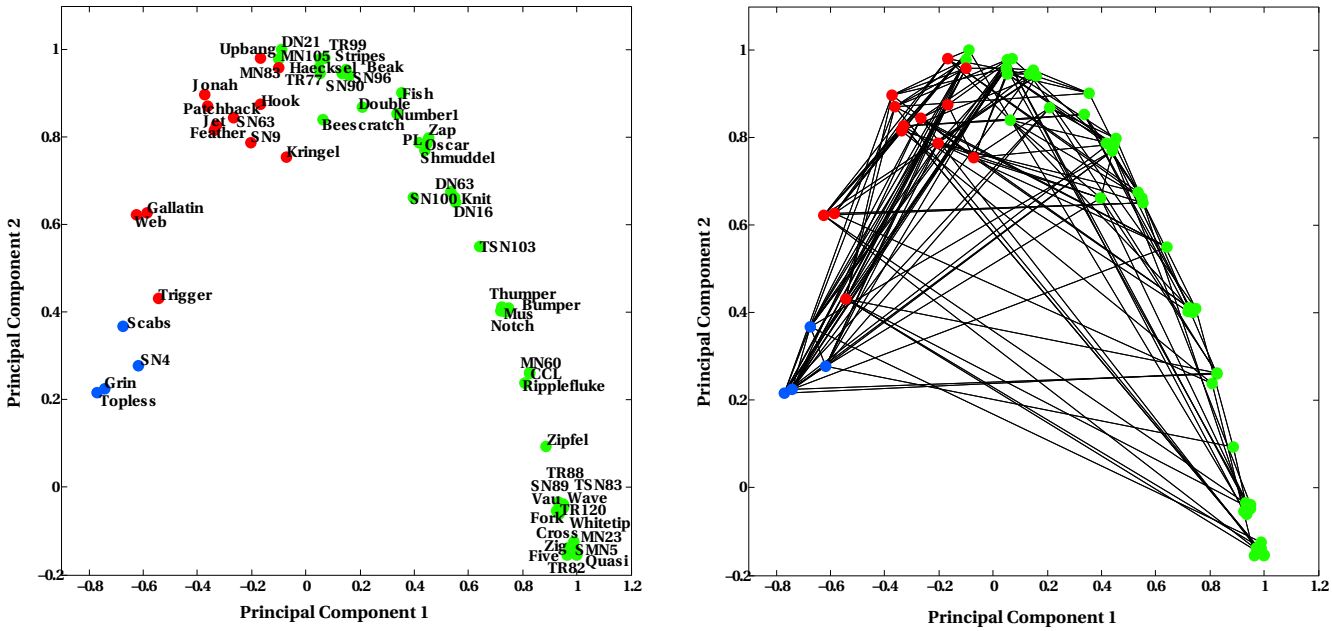

Figure 3: A dolphin social network is shown. The axes denote the first two principal components of the Euclidean embedding of the nodes (i.e. dolphins). Clustering the dolphins (using the k-means algorithm) yields three main types of local topologies in this social network, which we can view as clusters of dolphin social behavior. The dolphin's names are shown, as are the edges in the network, and the marker color in both plots denotes each dolphin's cluster assignment.

3 Experiments

3.1 Graph Clustering with Local Topologies in a Dolphin Social Network

The dataset in this experiment consists of a social network between a collection of 62 dolphins [2]. An edge was assigned between two dolphins if researchers spotted them traveling in the same group.

Embedding was performed using the MCMC algorithm to compute the FNED between 1-step local topologies. After embedding, the k-means algorithm was carried out to perform clustering of the dolphins based on their local topologies. The k-means algorithm was initialized at 10 clusters, but returned three clusters as a result. The embedding seems to separate dolphins by their social patterns, and the three clusters might be interpreted as high sociability, medium sociability, and low sociability. Note that the method of node embedding presented in this paper could be viewed as a richer version of node degree, as it yields a high dimensional description of topological structure around a node, while degree yields a very local, 1-dimensional description. Figure 3 plots the first two principal components of the node embedding, with marker color denoting the k-means cluster assignment (note that k-means was performed on the embedded feature vector, not on the principal components). The second plot on this figure shows the dolphin social network graph. Visually, one can see the number of edges decreasing when moving from left to right along the curve.

3.2 Words in Dickens' David Copperfield with Adjective-Noun Labels

The dataset in this experiment consists of the 60 most common nouns and adjectives in Charles Dickens' David Copperfield [3]. Edges are assigned between words if they appear adjacent to each other at any point in the text.

In this experiment, we hope to show that the embedding method described in this paper yields features that could be used to benefit a supervised machine learning problem; if so, Euclidean local topology features could be added to the features of datasets that are equipped with information about relationships between data elements. In particular, we hope to see some sort of differentiation between the embedding of adjectives and the embedding of the nouns in this dataset.

Embedding was performed using the MCMC algorithm to compute the FNED between 1-step local topologies. Figure 4 plots the first two dimensions (and first three dimensions on the right) of the embedding feature vectors. A few of the words, along with the graph edges, are shown in the two plots. Adjectives are colored red and nouns are colored blue. From this embedding, we can see that the adjectives and nouns are slightly differentiated in the embedding space. Additionally, a group of adjectives are isolated from the rest of the words.

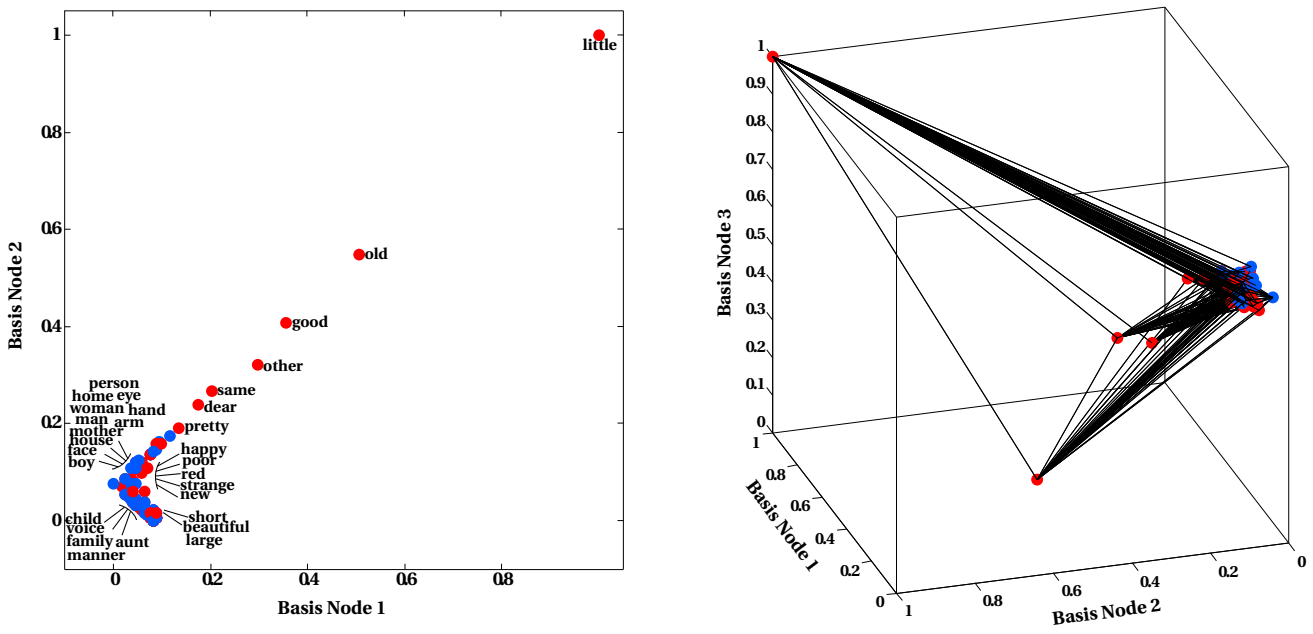

Figure 4: The 60 most common adjectives and nouns in Dickens' David Copperfield. Red markers denote adjectives and blue markers denote nouns. Words of different types tend to be embedded in different positions in space.

3.3 College Football Team Matches with League Labels

The datset in this experiment consists of the "matchups" between 115 college football teams [3] in a given season. Edges are assigned between two teams if they play each other during the season. Additionally, this dataset provides a label for each team to the league in which it resides (out of 12 leagues).

Similar to the previous experiment, we hope to show that the embedding method described in this paper yields features that could be used to benefit a supervised machine learning problem; in particular, we hope to see an embedding that places teams with similar matchup topologies near each other in space. We would guess that teams residing in the same league have similar matchup topologies.

Embedding was performed using the MCMC algorithm to compute the FNED between 1-step local topologies. Figure 5 shows the first two principal components of the Euclidean embedding. This figure also shows school name for each team and the matchup graph edges. In both plots, color denotes league assignment. We find that the embedding often places football teams from the same league at similar points in space. We also find a number of other points of interest, such as places where schools from similar geographic locations are embedded similarly (even for similar geographic schools in different leagues).

4 Conclusion

We have introduced a method for representing the local topology around a node in Euclidean space by defining the k-step local topology and fixed node edit distance between nodes. Additionally, we have provided two algorithms for computing the FNED between nodes of different types of graphs, and have demonstrated this embedding on three publically available datsets. Our demonstrations have shown that the Euclidean local topology features can be used alone to perform graph clustering of nodes into groups with similar local topologies, or can provide additional features for datasets equipped with relationships between their elements, in order to benefit a supervised learning task.

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] D. C. Reis, P. B. Golgher, A. S. Silva, and A. F. Laender. Automatic web news extraction using tree edit distance. In

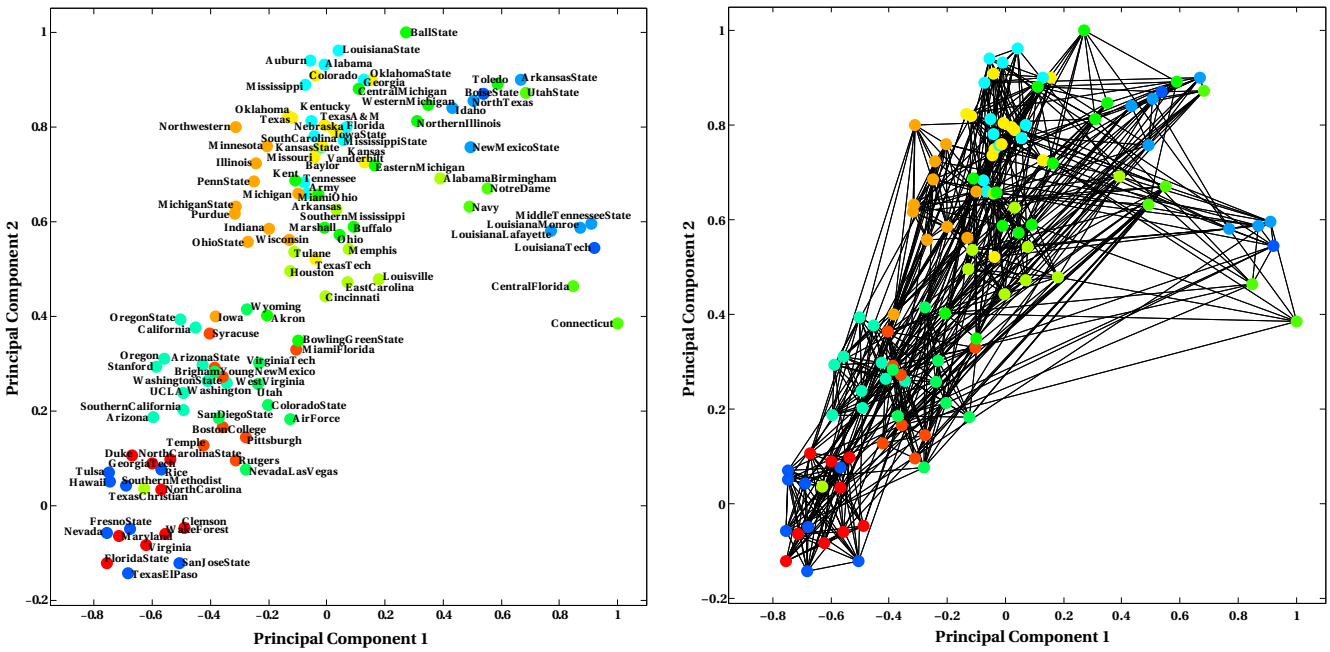

Figure 5: A network of college football team matchups during a given season is shown. Markers are colored based on the league (out of 12) in which the associated team resides. Teams belonging to the same league tend to be embedded similarly, as do teams from similar geographic locations.

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.

[5] 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.