Graph Clustering using Graph Entropy Complexity Traces

Lu Bai, Edwin R. Hancock* and Lin Han
Department of Computer Science, University of York, York YO10 5DG, UK
{lu,erh,hanlin}@cs.york.ac.uk

Peng Ren†
China University of Petroleum(Huadong), China
pengren@upc.edu.cn

Abstract

In this paper, we aim to present a principled approach to the problem of depth-based complexity characterisation of graphs. Our idea is to decompose graphs into substructures of increasing size, and then to measure the complexity of these substructures using Shannon entropy or von-Neumann entropy. We commence by identifying the dominant vertex in a graph. From the dominant vertex, we construct subgraphs of increasing K layers, so-called semidiameter subgraphs. We then measure how the entropy varies with increasing K layer semidiameter subgraphs. We construct a vector of subgraph entropies for each graph, a depth-based complexity trace, and then perform graph clustering in the principal components space of the vectors. We explore our approach on both synthetic data and datasets from the domain of bioinformatics.

1. Introduction

Graph based relational representations have proven to be more powerful than the simple vectorial feature representations. Compared to vectorial representations, a major drawback with graph representations is the lack of available standard machine learning algorithms for tasks such as clustering and classification. One way to overcome this problem is to embed the graph data into a vector space. However, there are two problems encountered when attempts are made to abstract feature vectors from a set of graphs. First, as the vectors can represent graphs of different size, then the vectors may be different length. The second problem is that information residing in the edges of the graph is discarded. In order to overcome these problems, several successful solutions have been developed. These include a) embedding graph into vector space using the dissimilarity embedding [1], b) representing graph structure using permutation invariant polynomials computed from the eigenvectors of the Laplacian matrix using algebraic graph theory [6] and c) computing permutation-invariant graph features via the Ihara zeta function [4].

Recently depth-based representations of undirected graph have proved to be powerful techniques not only for gauging the topological structure of graphs, but also for measuring their intrinsic complexity [2]. One approach is to gauge information content flow through the layers of a graph, and to use this as a structural signature. This approach allows a complexity trace to be defined which gauges how the complexity of the graph varies as a function of depth. Unfortunately, to construct such a trace requires a measure of intrinsic structural complexity, and this requires burdensome computations. To overcome this limitation, we focus on developing an efficient depth-based signature, that can both capture fine structure and can be evaluated relatively efficiently. To construct a depth-based complexity trace of a graph G, we commence by identifying the dominant vertex and deriving the semidiameter subgraphs of G. Then we construct a entropy complexity trace of G by measuring how the entropy varies on the hierarchy based semidiameter subgraphs of increasing K layers. The K-th element of the complexity trace vector is computed using the Shannon entropy or von Neumann entropy of the remaining semidiameter subgraphs. We empirically demonstrate that our entropy complexity trace can easily scale to large graphs.

This paper is organized as follows. Section 2 gives the definition of dominant vertex and semidiameter sub-
graphs. We show how to construct complexity trace for individual graph using graph entropy. Section 3 provides experimental evaluation. Finally, Section 4 provides conclusions.

2 Graph Complexity Trace

In this section, we investigate how to construct a complexity level characterisation of graph structure using graph entropy. The idea is as follows. We commence by using Dijkstra’s algorithm to identify the dominant vertex in a graph. From the shortest path matrix and the dominant vertex, we construct semidiameter subgraphs of increasing $K$ layers. We then measure how the subgraph entropy varies as the layers of semidiameter subgraphs increase.

2.1 Dominant Vertex

Suppose a graph $G(V, E)$ is size $n = |V|$. Using the Dijkstra algorithm, the first step in the search for the dominant vertex is to generate shortest path matrix $S_G$. Each element $S_G(v_i, v_j)$ of the matrix represents the shortest path between vertices $v_i$ and $v_j$. The average shortest path vector $S_V$ of graph $G(V, E)$ is a vector with the same vertex sequence in the adjacency matrix $A$, and each element $S_V(i) = \frac{\sum_{j=1}^{n} S_G(v_i, v_j)}{n}$ represents the average shortest path of vertex $v_i$ to the remaining vertices. The shortest path covariance vector $C_V$ of $G(V, E)$ is a vector with the same vertex sequence as the adjacency matrix $A$, and each element is defined as:

$$C_V(i) = \sum_{j=1}^{n} [S_G(v_i, v_j) - S_V(i)]^2 \quad (1)$$

Then the vertex $v_i$ with the smallest value of $C_V(i)$ in the covariance shortest path vector $C_V$, is the dominant vertex $v_D$ of $G(V, E)$. The dominant vertex can thus be viewed as the centroid vertex of $G(V, E)$. Compared to the remaining vertices, the property of the dominant vertex will be presented in the next subsection in terms of semidiameter subgraphs.

2.2 The $K$ Layer semidiameter Subgraphs

Consider the graph $G(V, E)$ and its shortest path matrix $S_G$, the set

$$S_K(v_i, G) := \{v \in V | d(v, v_i) \leq K, K \geq 1\} \quad (2)$$

is the set of vertices in the $K$-semidiameter sphere around vertex $v_i$ of $G(V, E)$. Then the subgraph $G_K^v(V_K^v, E_K^v)$ ($K \geq 1$), which possesses the same set of vertices in the $K$-semidiameter sphere and the same corresponding connection relationships of these vertices in $G(V, E)$ with adjacency matrix $A$, is called the $K$ layer semidiameter subgraph of $G(V, E)$ rooted from vertex $v_i$. Furthermore, it should be noted that the dominant vertex and the semidiameter subgraphs rooted from the dominant vertex possess the following characteristics: a) the vertices distribution around the dominant vertex is more symmetric than that around the remainders, and b) the dimension of the final vectorial description is equal to the greatest length of the shortest path from the dominant vertex to the remainder.

2.3 Steady State Random Walk

We use the state vector of the steady state random walk on the graph to calculate the Shannon entropy of (sub)graph. For the graph $G(V, E)$ the adjacency matrix $A$ has elements

$$A(u, v) = \begin{cases} 1 & \text{if}(u, v) \in E; \\ 0 & \text{otherwise}. \quad (3) \end{cases}$$

The degree matrix $D$ of $G(V, E)$ is a diagonal matrix whose elements are given by the degrees of the vertices, i.e. $D(u, u) = d_u = \sum_{v, v \in V} A(u, v)$. For vertex $u \in V$ the probability of a steady state random walk on $G(V, E)$ visiting vertex $u$ is

$$P_G(u) = \frac{d_u}{\sum_{v \in V} d_v} \quad (4)$$

2.4 Pattern Vectors from Graph Entropy

For a graph $G(V, E)$, the full set of semidiameter subgraphs is defined as:

$$G_{v_D}^D = \{G_{v_D}^{v_D}, ..., G_{v_D}^{v_{v_D}}, G_{v_D}^{v_{v_D}} \} \quad (5)$$

where $v_D$ is the dominant vertex of $G(V, E)$ and $L$ is the longest shortest path from the dominant vertex $v_D$ to the remaining vertices in $G(V, E)$. Suppose we have probability distributions resulting from the steady state random walk on each layer semidiameter subgraph rooted from the dominant vertex $v_D$ denoted by $P_{v_D}^{v_D}, ..., P_{v_D}^{v_{v_D}}$. We construct the depth-based complexity trace vector of $G(V, E)$ as

$$C_S(G) = [H_S(P_{v_D}^{v_D}), ..., H_S(P_{v_D}^{v_{v_D}}), ..., H_S(P_{v_D}^{v_{v_{v_D}}})] \quad (6)$$

where $H_S(P_{v_D}^{v_D})$ is the Shannon entropy of the $K$ layer semidiameter subgraph $G_K^v(V_K^v, E_K^v)$ associated with the probability distribution $P_{v_D}^{v_D}$ given by:

$$H_S(P_{v_D}^{v_D}) = - \sum_{q=1}^{|V_{v_D}^v|} P_{v_D}^{v_D}(q) \log P_{v_D}^{v_D}(q) \quad (7)$$
We also use the approximated von Neumann entropy [3] of the $K$ layer semidiameter subgraph $G_K^{v_D}(V_K^{v_D}, E_K^{v_D})$ to construct the vector. The computation of the approximated von Neumann entropy can be rendered quadratic in the number of the vertices. Suppose the degree matrix of $G_K^{v_D}(V_K^{v_D}, E_K^{v_D})$ is a diagonal matrix $D_K$ whose elements are given by

$$D_K(u_K, v_K) = d_{uk} = \sum_{v_K \in V_K^{v_D}} A_K^{v_D}(u_K, v_K) \quad (8)$$

Based on the definition in [3], the approximated von Neumann entropy of the $K$ layer semidiameter subgraph is defined as

$$H_V(G_K^{v_D}) = \frac{|V_K^{v_D}|}{4} - \sum_{(u_K, v_K) \in E_K^{v_D}} \frac{1}{4 d_{u_K} d_{v_K}} \quad (9)$$

Then the complexity trace vector associated with von Neumann entropy can also be written as

$$C_V(G) = [H_V(G_1^{v_D}), ..., H_V(G_2^{v_D}), ..., H_V(G_L^{v_D})] \quad (10)$$

### 2.5 Graphs of Different Size

We need to be able to compare graphs using our complexity trace vectors of uniform length. This is achieved by padding out the dimensions of the vectors. Consider two different size graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ with dominant vertices $v_{D1}$ and $v_{D2}$ respectively. Suppose that the greatest lengths of the shortest paths from the dominant vertices $v_{D1}$ and $v_{D2}$ are $L_1$ and $L_2$, respectively, where $L_1 > L_2$. Then the dimensions of the vectors are $L_1$ and $L_2$ respectively. We add $L_1 - L_2$ vertices from any vertex on the $L_2$-semidiameter sphere for $v_{D2}$ to extend the sphere to the $L_1$-semidiameter sphere for $v_{D1}$ so that the dimensions of the two vectors are both $L_1$. We assume that the added vertices in $G_2(V_2, E_2)$ have no connecting edges. The padding value of the complexity trace vector of $G_2(V_2, E_2)$ is calculated using the entropy of the $L_2$ layer semidiameter subgraph (i.e. $G_2$ itself) of $G_2(V_2, E_2)$.

### 3. Experiments

#### 3.1 Interior Complexity Evaluation

We commence by illustrating how the complexity trace depends on depth in the graph, and demonstrate that this can be used to distinguish different objects. The evaluation utilizes graphs extracted from images of a box, a house and a cup, taken respectively from the ALOI, CMU and COIL databases. For each object we use 18 images captured from different viewpoints. The graphs are the Delaunay triangulations of feature points extracted from the different images. For each graph, we locate the dominant vertex and construct the semidiameter subgraphs. The interior complexity values are computed using the Shannon or von Neumann entropy defined in (7) and (9). Fig.1 shows the sets of complexity histograms of raw complexity traces (18 per object) for each object in turn. The main features to note are that the traces from the same object are similar, whereas those from different objects are dissimilar.

### 3.2 Stability Evaluation on Synthetic Data

Next, we explore the relationship between the graph edit distance and the depth-based entropy complexity trace vectors resulting from our methods. The evaluation utilizes three randomly generated seed graphs. The three seed graphs have 444 vertices, 210 vertices and 337 vertices respectively. For each seed graph, we apply random edit operation of vertex fraction deletion to simulate the effects of noise. The feature distance of the original seed graph $G_o$ and its noise corrupted counterpart $G_e$ is defined as their Euclidean distance given by

$$d_{G_o, G_e} = \sqrt{(C(G_o) - C(G_e))^T (C(G_o) - C(G_e))}$$

The experimental results are shown in Fig.2. The x-axis shows the 1% to 35% of vertices randomly deleted, and the y-axis shows the value of the Euclidean distance $d_{G_o, G_e}$ between the original seed graph $G_o$ and its noise corrupted counterpart $G_e$. It is clear that when less than 20% vertices deleted, the fluctuation is moderate and when less than 5% are deleted the effect is small. This means that our vectorial description is robust even when the dominant vertex undergoes relatively large perturbations even if the dominant vertex is itself noise. As a whole, there is an approximately linear relationship between the graph edit distance and the Euclidean distance.
Datasets: We employed the MUTAG, NCI109 and D&D datasets in our experiments. MUTAG consists of 188 mutagenic aromatic and heteroaromatic nitro compounds. NCI109 contains 4127 chemical compounds with a balance sub-set. Finally, D&D contains 1178 protein structures. In each case the graphs contained in the datasets have been transformed into unweighted graphs. Details of the datasets used can be found in [5]. The maximum number of vertices, average number of vertices and the numbers of testing graphs in these datasets are 28, 17.93 and 188, 111, 29.68 and 4127, and 5748 284.32 and 1178 respectively.

Experimental setup: For each dataset, we construct the complexity trace vectors \( C_S(G) \) and \( C_V(G) \) for each graph using the proposed methods, and then pad the vectors so that they reside in a uniform dimensioned vector space. We also compared the proposed methods with two fast alternative methods reported in literature. These are a) pattern vectors from coefficients of the Ihara zeta function (CIZF) [4] and b) pattern vectors from algebraic graph theory associated (PVAG) [6]. We performed 10-fold cross-validation of C-Support Vector Machine (C-SVM), using 9 samples for training and 1 sample for testing. All parameters of the C-SVM classifier were optimized on the training dataset. We report the average prediction accuracy and runtime in Table 1. The runtime was measured for Matlab R2011a running on a 3.2GHz Intel 4-Core processor.

Results: Table 1 shows that our methods possess efficient computation and good performance on this real-world application. In terms of the runtime and graph size, our methods can efficiently compute graph complexity trace even for graphs with thousands of vertices. While CIZF and PVAG prove computationally burdensome, and can not finish in one day. The runtime and classification performance of CIZF and PVAG is competitive to our methods on MUTAG and NCI109 which contain small size graphs, but is much slower and renders limited use on the D&D dataset which contains large graphs. Our methods outperform CIZF and PVAG on runtime and classification accuracy for datasets of large graphs. The complexity trace computed using the Shannon entropy is a little more efficient but gives lower performance than that computed using the von Neumann entropy.

4. Conclusion

In this paper, we have shown how the graphs can be characterised using the graph entropy complexity traces. Our characterisation commences from the dominant vertex and its semidiameter subgraphs with increasing \( K \) layers. We compute the entropies of the semidiameter subgraphs rooted from the dominant vertex. We embed sets of graphs into feature space for clustering. Here we use 10-fold cross-validation of C-SVM to assign the graphs to classes. Experiments on a number of standard bioinformatics databases reveal that our method is both effective and outperforms alternatives.

References