Graph Complexity from the Jensen-Shannon Divergence

Lu Bai and Edwin R. Hancock*

Department of Computer Science
University of York, UK
Deramore Lane, Heslington, York, YO10 5GH, UK

Abstract. In this paper we aim to characterize graphs in terms of structural complexities. Our idea is to decompose a graph into substructures of increasing layers, and then to measure the dissimilarity of these substructures using Jensen-Shannon divergence. We commence by identifying a centroid vertex by computing the minimum variance of its shortest path lengths. From the centroid vertex, a family of centroid expansion subgraphs of the graph with increasing layers are constructed. We then compute the depth-based complexity trace of a graph by measuring how the Jensen-Shannon divergence varies with increasing layers of the subgraphs. The required Shannon or von Neumann entropies are computed on the condensed subgraph family of the graph. We perform graph clustering in the principal components space of the complexity trace vector. Experiments on graph datasets abstracted from bioinformatics and image data demonstrate effectiveness and efficiency of the graphs complexity traces.

1 Introduction

Graph based relational representations have proven to be both powerful and flexible in pattern recognition. Compared to vector based pattern recognition, a major drawback with graph representations is the lack of a natural correspondence order. This limits the direct application of standard machine learning algorithms for problems such as graph clustering. One way to overcome this problem is to embed the graph data into a vector space, where standard machine learning techniques can be deployed. There have been several successful solutions which include a) embedding graph into vector space using the dissimilarity embedding [5], b) representing graph structure using permutation invariant polynomials computed from the eigenvectors of the Laplacian matrix using algebraic graph theory [10], and c) computing permutation-invariant graph features via the Ihara zeta function [7]. The limitations of the existing methods is that they usually depend on the graph topology or size, and as a result they tend to be computationally burdensome or can not be efficiently computed in an algebraic manner.

* Edwin R. Hancock is supported by a Royal Society Wolfson Research Merit Award.

G.L. Gimel' farb et al. (Eds.): SSPR & SPR 2012, LNCS 7626, pp. 79–88, 2012.

To overcome the limitations of existing methods, we propose a novel framework for characterizating graphs based on computing complexity traces. Depth-based representations of undirected graph structures have proved powerful for characterizing their topological structure in terms of intrinsic complexity [1,2]. One approach is to gauge information content flow through K layer subgraphs of a graph (e.g.subgraphs around a vertex having a maximum topology distance or minimal path length K) of increasing size and to use the flow 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 [2]. Unfortunately, to construct such a trace requires a measure of the intrinsic structural complexity, and this requires burdensome computations. In this paper we focus on developing an efficient depth-based signature, that can both capture fine structure and can be evaluated relatively efficiently. To compute a complexity trace of a graph G, we identify the centroid vertex v_C in G by selecting the vertex with minimum variance of shortest path lengths. Based on v_C, we derive a family of expansion subgraphs from v_C with in increasing layer size K. Then we construct a complexity trace of G by measuring how the dissimilarity between the K layer subgraph and G varies on the expansion subgraphs with the increasing layer K. To compute the proposed depth-based complexity trace efficiently, we turn to the Jensen-Shannon divergence as the dissimilarity measure. This is a nonextensive information theoretic measure derived from the mutual information between probability distributions over different structures. Here the required entropies of the Jensen-Shannon divergence are computed using the Shannon entropy or von Neumann entropy on the (sub)graphs. We empirically demonstrate that our Jensen-Shannon complexity trace can easily scale to large graphs. The performance of our framework is competitive to the state of the art methods in the literature.

2 Centroid Expansion Subgraphs

In this section, we introduce a set of subgraphs which we refer to as centroid expansion subgraphs of a given graph. We first describe how to identify the centroid vertex for a graph and explain how to extract the centroid expansion subgraphs from the graph with regard to the centroid vertex. Then we describe how to compute entropies on these centroid expansion subgraphs.

2.1 Centroid Vertex

The shortest path for a pair of vertices v_i and v_j in an undirected graph $G(V, E)$ can be obtained by using Dijkstra algorithm. We refer to the matrix S_G whose elements $S_G(i, j)$ represents the shortest path length between vertices v_i and v_j as shortest path matrix for $G(V, E)$. The average-shortest-path vector S_V for $G(V, E)$ is a vector with the same vertex sequence as S_G, with each element $S_V(i) = \sum_{j=1}^{|V|} S_G(i, j)/|V|$ representing the average shortest path length from

vertex v_i to the remaining vertices. We then locate the centroid vertex v_i for $G(V,E)$ as follows

$$\hat{i} = \arg\min_i \sum_{j=1}^{|V|} [S_G(i,j) - S_V(i)]^2. \tag{1}$$

The centroid vertex v_i of $G(V,E)$ is located through selecting a vertex with a minimum variance of shortest path lengths out of all vertices in $G(V,E)$. Therefore, the shortest paths starting from the centroid vertex v_i form a *steady* path set that exhibits less length variability than those path sets starting from other vertices. For a graph $G(V,E)$ with the centroid vertex v_C, the K-layer centroid expansion subgraph $\mathcal{G}_K(\mathcal{V}_K;\mathcal{E}_K)$ is

$$\begin{cases} \mathcal{V}_K := \{u \in V | S(v_C,u) \leq K, K \geq 1\}; \\ \mathcal{E}_K := \{\{v,u\} \subseteq V_K | \{v,u\} \in E\}. \end{cases} \tag{2}$$

The number of centroid expansion subgraphs is equal to the greatest length of the shortest path from the centroid vertex to the other vertices of the graph.

2.2 Entropies on K-Layer Centroid Expansion Subgraphs

The definition of steady state random walks and entropy on a subgraph is similar to that for a graph. Given the K-layer centroid expansion subgraph $\mathcal{G}_K(\mathcal{V}_K;\mathcal{E}_K)$ of a graph $G(V,E)$, the adjacency matrix A_K for $\mathcal{G}_K(\mathcal{V}_K;\mathcal{E}_K)$ has elements

$$A_K(i,j) = \begin{cases} 1 \text{ if} (v_i,v_j) \in \mathcal{E}_K; \\ 0 \text{ otherwise.} \end{cases} \tag{3}$$

The vertex degree matrix of $\mathcal{G}_K(\mathcal{V}_K;\mathcal{E}_K)$ is a diagonal matrix D_K whose elements are given by $D_K(v_i,v_i) = d_K(i) = \sum_{v_i,v_j \in \mathcal{V}_K} A_K(i,j)$. From the matrixes D_K and A_K we can construct the Laplacian matrix $L_K = D_K - A_K$. The normalized Laplacian matrix is given by $\hat{L}_K = D_K^{-1/2} L_K D_K^{-1/2}$. The spectral decomposition of the normalized Laplacian matrix is $\hat{L}_K = \hat{\Phi}_K \hat{\Lambda}_K \hat{\Phi}_K^T$ where $\hat{\Lambda}_K = diag(\hat{\lambda}_{K_1}, \hat{\lambda}_{K_2}, ..., \hat{\lambda}_{K_{|\mathcal{V}_K|}})$ is a diagonal matrix with the ordered eigenvalues as elements $(0 = \hat{\lambda}_{K_1} < \hat{\lambda}_{K_2} < ... < \hat{\lambda}_{K_{|\mathcal{V}_K|}})$ and $\hat{\Phi}_K = (\hat{\phi}_{K_1} | \hat{\phi}_{K_2} | ... | \hat{\phi}_{K_{|\mathcal{V}_K|}})$ is a matrix with the corresponding ordered orthonormal eigenvectors as columns. The normalized Laplacian matrix is positive semi-definite and so has all eigenvalues non-negative. The number of zero eigenvalues is the number of connected components in $\mathcal{G}_K(\mathcal{V}_K;\mathcal{E}_K)$. In [12], the von Neumann entropy of $\mathcal{G}_K(\mathcal{V}_K;\mathcal{E}_K)$ associated with the normalized Laplacian eigenspectrum is defined as $H_{VN} = -\sum_{i=1}^{|\mathcal{V}_K|} \frac{\hat{\lambda}_{K_i}}{2} \log \frac{\hat{\lambda}_{K_i}}{2}$. Since the computation of the von Neumann entropy requires cubic number of vertices operations, Han et al. [3] have shown how the computation can be rendered quadratic in the number of the vertices. By approximating the von Neumann entropy by its quadratic counterpart, the approximated von Neumann entropy for $\mathcal{G}_K(\mathcal{V}_K;\mathcal{E}_K)$ is given by

$$H_{VN}(\mathcal{G}_K) = \frac{|\mathcal{V}_K|}{4} - \sum_{(v_i,v_j) \in \mathcal{E}_K} \frac{1}{4\ d_K(i) d_K(j)} \tag{4}$$

Furthermore, the probability of a steady state random walk on $\mathcal{G}_K(\mathcal{V}_K;\mathcal{E}_K)$ visiting vertex v_i is $P_K(i) = d_K(i)/\sum_{v_j\in\mathcal{V}_K} d_K(j)$. The Shannon entropy of $\mathcal{G}_K(\mathcal{V}_K;\mathcal{E}_K)$ with the probability distribution $P\{\mathcal{G}_K\} = P_K$ is then given by

$$H_S(P\{\mathcal{G}_K\}) = H_S(P_K) = -\sum_{i=1}^{|\mathcal{V}_K|} P_K(i)\log P_K(i). \tag{5}$$

3 Jensen-Shannon Complexity Traces of Graphs

In this section, we investigate how to use the Jensen-Shannon divergence as a means of constructing a depth-based complexity trace of graph-structure.

3.1 Jensen-Shannon Divergence Measure

The Jensen-Shannon divergence is a nonextensive mutual information measure. It is defined on probability distributions over structured data [4]. The Jensen-Shannon divergence $JSD(P_m, P_n)$ between probability distributions P_m and P_n is given by:

$$JSD(P_m, P_n) = H_S(\frac{P_m + P_n}{2}) - \frac{H_S(P_m) + H_S(P_n)}{2} \tag{6}$$

where $H_S(P_m)$ is the Shannon entropy for the probability distribution P_m.

3.2 Composite Structure of Subgraphs

Before we use the Jensen-Shannon divergence as a means of constructing a complexity trace of a graph, we required a composite structure graph of a pair of (sub)graphs. For a pair of subgraphs $\mathcal{G}_K(\mathcal{V}_K,\mathcal{E}_K)$ and $\mathcal{G}_{K'}(\mathcal{V}_{K'},\mathcal{E}_{K'})$, their composite structure graph $\mathcal{G}_K \oplus \mathcal{G}_{K'}$ has vertex and edge sets $\mathcal{V}_K \oplus \mathcal{V}_{K'}$ and $\mathcal{E}_K \oplus \mathcal{E}_{K'}$ respectively. The most common algorithms to create a composite structure graph of two initial (sub)graphs are formed by taking graph product and graph union. For reason of the efficient computation here we take the (sub)graph union. To construct an union graph $\mathcal{G}_U(\mathcal{V}_U,\mathcal{E}_U)$ of $\mathcal{G}_K(\mathcal{V}_K,\mathcal{E}_K)$ and $\mathcal{G}_{K'}(\mathcal{V}_{K'},\mathcal{E}_{K'})$, we perform pairwise correspondence matching. Details of the construction are outside the scope of this paper. Our approach follows that of Han et.al's work in [11].

3.3 Complexity Characterisation of Graph Structure

We define a depth-based Jensen-Shannon complexity trace for a graph. For a graph $G(V,E)$ the full set of its centroid expansion subgraphs is $G_C^{v_C} = \{\mathcal{G}_1, ..., \mathcal{G}_K, ..., \mathcal{G}_L\}$ where v_C is the centroid vertex of G, L is the greatest length of shortest paths from the centroid vertex v_C to the remaining vertices in $G(V,E)$, and $\mathcal{G}_K$ is the K-layer centroid expansion subgraph of $G(V,E)$. The essentiality of the L layer subgraph is the graph $G(V,E)$ itself. Suppose we have probability distributions resulting from

steady state random walks on each of the K layer centroid expansion subgraph $\mathcal{G}_K$ denoted by $P\{\mathcal{G}_1\}, ..., P\{\mathcal{G}_K\}, ..., P\{\mathcal{G}_L\}$. The complexity trace is computed as

$$CT = [JSD(P\{\mathcal{G}_1\}, P\{\mathcal{G}_L\}), ..., JSD(P\{\mathcal{G}_K\}, P\{\mathcal{G}_L\}), ..., JSD(P\{\mathcal{G}_L\}, P\{\mathcal{G}_L\})]^T \tag{7}$$

where $JSD(P\{\mathcal{G}_K\}, P\{\mathcal{G}_L\})$ is the Jensen-Shannon divergence between the K layer centroid expansion subgraph and the L layer centroid expansion subgraph (i.e. graph $G(V, E)$). This complexity trace encapsulates an mutual information based interior dissimilarity transformation between the graph $G(V, E)$ and its K, which is from 1 to L, layer centroid expansion subgraphs with their steady state random walk probability distributions. The Jensen-Shannon divergence $JSD(P\{\mathcal{G}_K\}, P\{\mathcal{G}_L\})$ is defined as:

$$JSD(P\{\mathcal{G}_K\}, P\{\mathcal{G}_L\}) = H_S(\frac{P\{\mathcal{G}_K\} \oplus P\{\mathcal{G}_L\}}{2}) - \frac{H_S(P\{\mathcal{G}_K\}) + H_S(P\{\mathcal{G}_L\})}{2} \tag{8}$$

where $\frac{P\{\mathcal{G}_K\} \oplus P\{\mathcal{G}_L\}}{2}$ represents the probability distribution of the steady state random walk over the union graph $\mathcal{G}_U(\mathcal{V}_U, \mathcal{E}_U)$ of $\mathcal{G}_K(\mathcal{V}_K, \mathcal{E}_K)$ and $\mathcal{G}_L(\mathcal{V}_L, \mathcal{E}_L)$. As the L layer expansion subgraph $\mathcal{G}_L(\mathcal{V}_L, \mathcal{E}_L)$ contains the full structure of the K layer expansion subgraph $\mathcal{G}_K(\mathcal{V}_K, \mathcal{E}_K)$, using the graph union mentioned in Section 3.3, $\mathcal{G}_U(\mathcal{V}_U, \mathcal{E}_U)$ can be represented by $\mathcal{G}_L(\mathcal{V}_L, \mathcal{E}_L)$. As a result (8) can be rewritten as:

$$JSD(P\{\mathcal{G}_K\}, P\{\mathcal{G}_L\}) = \frac{H_S(P\{\mathcal{G}_L\}) - H_S(P\{\mathcal{G}_K\})}{2} \tag{9}$$

Since we also use the von Neumann entropy in (4) to construct the complexity trace CT, then CT in (7) can also be written as

$$CT = [JSD(\mathcal{G}_1, \mathcal{G}_L), ..., JSD(\mathcal{G}_K, \mathcal{G}_L), ..., JSD(\mathcal{G}_L, \mathcal{G}_L)]^T \tag{10}$$

where $JSD(\mathcal{G}_K, \mathcal{G}_L)$ is given by

$$JSD(\mathcal{G}_K, \mathcal{G}_L) = \frac{H_{VN}(\mathcal{G}_L) - H_{VN}(\mathcal{G}_K)}{2} \tag{11}$$

3.4 Graphs of Different Size

The L layer expansion subgraph is the undirected graph itself, and the dimension of a Jensen-Shannon complexity trace vector is thus equal to greatest layer L. However, the complexity trace vectors for graphs of different sizes may exhibit various lengths. To compare these graphs by using complexity trace vectors, we need to make vector lengths uniform. This is achieved by padding out the dimensions of the complexity trace vectors. Hence, for complexity trace vectors CT_m and CT_n of two graphs G_m and G_n with dimensions L_m and L_n respectively, where $L_m > L_n$, we use the L_n-th element value of CT_n as the added padding value for the extended $L_n + 1$-th to L_m-th elements of CT_n. Since the L_n-th element Jensen-Shannon divergence value is 0, the padding values are 0.

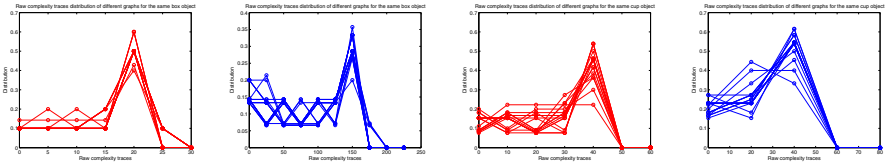
3.5 Computational Complexity Evaluation

The computational complexity of proposed complexity trace is governed by four computational step. Consider a sample graph $G(V, E)$ with size s and highest shortest path length L for the centroid vertex. The Dijkstra shortest path calculation requires $O(s^2)$ operations. The processing of centroid expansion subgraphs requires $O(Ls^2)$ operations. Since the L layer centroid expansion subgraph possesses the full structures of any K layer centroid expansion subgraphs, the union graph is the L layer centroid expansion subgraph. As a result the union graph construction approximately requires $O(s)$ operations. The Jensen-Shannon divergence calculation approximately requires $O(\sqrt[3]{s^2})$ operations. The L is approximated equal to $\sqrt[3]{s}$.

4 Experimental Evaluation

4.1 Interior Complexity Evaluation

We commence by illustrating how the representational power of the proposed complexity traces of graphs, and demonstrate that these can be used to distinguish different objects. The evaluation utilizes graphs extracted from images of a box and a house, taken respectively from the ALOI and CMU 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 identify the centroid vertex and construct centroid expansion subgraphs. The interior complexity values are computed using (9) or (11). Figs.1(a)(b) and (c)(d) show the sets of complexity histograms of complexity traces using Shannon or von Neumann entropy (18 per object) for each object in turn respectively. The main features to note are that the distributions from the same object are similar to each other, whereas those from different objects are dissimilar.



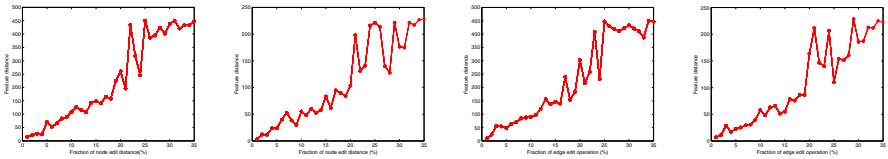
(a) (b) (c) (d)

Fig. 1. Complexity histograms of complexity traces of different graphs

4.2 Stability Evaluation on Centroid Vertex

To evaluate the stability of our proposed complexity trace from the centroid vertex, we explore the relationship between graph edit distance and the pattern vectors resulting from our complexity trace vectors of graphs. The evaluation utilizes two randomly generated seed graphs. The two seed graphs have 500 vertices and

300 vertices respectively. For each seed graph, we first identify its centroid vertex as the original centroid vertex, then we apply random edit operations of edges fraction addition 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 $d_{G_O,G_E} = \sqrt{(CT_O - CT_E)^T (CT_O - CT_E)}$ where CT_O and CT_E are complexity traces of G_O and G_E from the same centroid vertex, i.e.the original centroid vertex. The experimental results are shown in Fig.2. Fig.2(a)(b) and (c)(d) show the feature distance between pattern vectors using Shannon or von Neumann entropy for the two seed graphs and their edited graphs respectively. In each subfigure, the x-axis shows the 1% to 35% of edges randomly added, and the y-axis shows the value of the Euclidean distance d_{G_O,G_E} between G_O and G_E. From Fig.2 it is clear that when less than 5% are added the fluctuation is small, and when around 20% are added the fluctuation becomes moderate. This implies that the proposed complexity trace from the centroid vertex is robust even when the seed graph structures undergo relatively large perturbations. As a whole, there's an approximately linear relationship between the graph edit distance and the Euclidean distance. This implies that the proposed method possesses the ability to distinguish graphs under controlled structural error.



(a) (b) (c) (d)

Fig. 2. Distance distribution between feature vectors versus graph edit operation

4.3 Real-World Datasets

We compare our proposed complexity trace method with several state of the art methods. The methods for comparison include 1) Jensen-Shannon graph kernel (JSGK) [6], 2) von-Neumann thermodynamic depth complexity (VNTD) [2,3], 3) information functionals f^V (FV) (e=1) and f^P (FP) (e=1) [8], and 4) Ihara coefficients for graphs (CIZF) [7]. We use three standard graph based datasets abstracted from bioinformatics datasets [9,2] for experimental evaluation. For the FV and FP, we set the parameters α as 2, and c_k and b_k as $\rho - k + 1$ [8].

MUTAG: The MUTAG benchmark is based on graphs representing 188 chemical compounds, and aims to predict whether each compound possesses mutagenicity. The maximum and average number of vertices are 28 and 17.93 respectively. As the vertices and edges of each compound are labeled with a real number, we transform these graphs into unweighted graphs.

PPIs: The PPIs dataset consists of protein-protein interaction networks (PPIs). The graphs describe the interaction relationships between histidine kinase in different species of bacteria. Histidine kinase is a key protein in the development of

signal transduction. If two proteins have direct (physical) or indirect (functional) association, they are connected by an edge. There are 219 PPIs in this dataset and they are collected from 5 different kinds of bacteria. We select *Proteobacteria*40 PPIs and *Acidobacteria*46 PPIs as the second group test graphs. The maximum, minimum and average number of vertices of selected graphs are 238, 6 and 109.60 respectively.

D&D: The D&D dataset contains 1178 protein structures. Each protein is represented by a graph, in which the nodes are amino acids and two nodes are connected by an edge if they are less than 6 Angstroms apart. The prediction task is to classify the protein structures into enzymes and non-enzymes. The maximum and average number of virtices are 5748 and 284.32 respectively.

ENZYMES: The ENZYMES dataset is a dataset based on graphs representing protein tertiary structures consisting of 600 enzymes from the BRENDA enzyme database. In this case the task is to correctly assign each enzyme to one of the 6 EC top-level classes. The maximum and average number of vertices are 126 and 32.63 respectively.

4.4 Performance Comparison

We evaluate the performance of our proposed Jensen-Shannon complexity trace using Shannon (JSCTS) or von Neumann (JSCTV) entropy on the mentioned standard datasets and compare them with several alternative state of the art graph based methods. We perform 10-fold cross-validation associated with SMO-Support Vector Machine Classification to evaluate the performance of our method and the alternatives, using nine samples for training and one for testing. All parameters of the SVMs were optimized. The codes of our previous work in [6] and the other methods were also re-optimized. We report the average prediction accuracies and runtime of each method in Table 1(-:infeasible runtime; =: over computing), the runtime were measured under Matlab R2011a running on a ThinkPad T61p with an Intel 2.2GHz 2-Core processor and 2GB RAM.

Table 1. Experimental Comparison on Bioinformatics Datasets

Datasets	**JSCTS**	**JSCTV**	JSGK	VNTD	FV	FP	CIZF
MUTAG	85.63	82.44	87.76	83.51	84.57	85.63	80.85
PPIs	76.74	77.90	69.85	67.44	70.93	70.93	70.93
Enzymes	29.00	32.16	27.05	30.50	24.17	23.33	32.00
D&D	75.32	76.15	78.00	−	=	=	− =
Datasets	**JSCTS**	**JSCTV**	JSGK	VNTD	FV	FP	CIZF
MUTAG	1”	1”	2”	19′21”	1”	1”	1”
PPIs	1”	1”	2”	52′27”	1”	1”	55”
Enzymes	1”	1”	19”	4h37′	1”	1”	11”
D&D	42”	44”	14′59”	−	=	=	− =

In terms of the runtime and graph size, our method can efficiently compute graph complexity traces even for graphs with thousands of vertices, while VNTD and CIZF prove computationally burdensome or can not be finished in one day on D&D dataset. Our method outperforms all the alternatives on classification accuracy, only the JSGK is competitive to ours on D&D and MUTAG datasets. Our method outperforms all the alternatives on runtime for datasets of large graphs. Compare to depth-base complexity measures VNTD, FV and FP, our depth-based Jensen-Shannon complexity trace using Shannon or von Neumann entropy outperforms all of them on classification and runtime.

5 Conclusion

In this paper, we have shown how to construct a depth-based Jensen-Shannon complexity trace for a graph. Our method is based on the graph decomposition and Jensen-Shannon divergence. For a graph, we have identified a centroid vertex by computing the minimum variance of its shortest path lengths, and thus obtained a family of expansion subgraphs with increasing layers. The proposed complexity trace of a graph has been constructed by measuring how the Jensen-Shannon divergence varies with increasing layers of the subgraphs. We use the Shannon entropy or von Neumann entropy to calculate the required entropies in the Jensen-Shannon divergence. Experiments on graph datasets abstracted from bioinformatics demonstrate effectiveness and efficiency of the proposed complexity trace.

References

1. Crutchfield, J.P., Shalizi, C.R.: Thermodynamic depth of causal states: Objective complexity via minimal representations. Physical Review E 59, 275–283 (1999)
2. Escolano, F., Hancock, E.R., Lozano, M.A.: Heat diffusion: Thermodynamic depth complexity of networks. Physical Review E 85, 206–236 (2012)
3. Han, L., Escolano, F., Hancock, E.R.: Graph characterizations from von Neumann entropy. To appear in Pattern Recognition Letter (2012)
4. Martins, A.F.T., Smith, N.A., Xing, E.P., Aguiar, P.M.Q., Figueiredo, M.A.T.: Nonextensive information theoretic kernels on measures. Journal of Machine Learning Research 10, 935–975 (2009)
5. Bunke, H., Riesen, K.: Improving vector space embedding of graphs through feature selection algorithms. Pattern Recognition 44, 1928–1940 (2011)
6. Bai, L., Hancock, E.R.: Graph Clustering Using the Jensen-Shannon Kernel. In: Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W. (eds.) CAIP 2011, Part I. LNCS, vol. 6854, pp. 394–401. Springer, Heidelberg (2011)
7. Ren, P., Wilson, R.C., Hancock, E.R.: Graph Characterization via Ihara Coefficients. IEEE Transactions on Neural Networks 22, 233–245 (2011)
8. Dehmer, M., Mowshowitz, A.: A history of graph entropy measures. Proceedings of Information Sciences 181, 57–78 (2011)
9. Shervashidze, N., Schweitzer, P., Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-Lehman graph kernels. Journal of Machine Learning Research 1, 1–48 (2010)

10. Wilson, R.C., Hancock, E.R., Luo, B.: Pattern vectors from algebraic graph theory. IEEE Transactions Pattern Analysis and Machine Intelligence 27, 1112–1124 (2005)
11. Han, L., Hancock, E.R., Wilson, R.C.: Learning Generative Graph Prototypes Using Simplified von Neumann Entropy. In: Jiang, X., Ferrer, M., Torsello, A. (eds.) GbRPR 2011. LNCS, vol. 6658, pp. 42–51. Springer, Heidelberg (2011)
12. Passerini, F., Severini, S.: Quantifying complexity in networks: the von Neumann entropy. International Journal of Agent Technologies and Systems 1, 58–67 (2009)