Elsevier Editorial System(tm) for Pattern Recognition
Manuscript Draft

Manuscript Number: PR-D-14-00925R1

Title: Fast Depth-Based Subgraph Kernels for Unattributed Graphs

Article Type: Full Length Article

Keywords: Depth-based representations

entropy

graph kernels

the Jensen-Shannon divergence

graph isomorphism tests

Corresponding Author: Mr. Lu Bai,

Corresponding Author's Institution: University of York

First Author: Lu Bai

Order of Authors: Lu Bai; Edwin Hancock

Abstract: In this paper, we investigate two fast subgraph kernels based on a depth-based representation of graph-structure. Both methods gauge
depth information through a family of K-layer expansion subgraphs rooted at a vertex \cite{Escolano-PhyRev2012}. The first method commences by computing a
centroid-based complexity trace for each graph, using a depth-based representation rooted at the centroid vertex that has minimum shortest path length variance to the remaining vertices \cite{BAI2014PatternRecognition}.
This subgraph kernel is computed by measuring the Jensen-Shannon divergence
between centroid-based complexity entropy traces. The second method, on the other hand, computes a depth-based
representation around each vertex in turn. The corresponding subgraph kernel is computed using isomorphisms tests to compare the depth-based representation rooted at each vertex in turn. For graphs with n vertices, the time complexities for the two new kernels are $O(n^2)$ and $O(n^3)$ respectively, in contrast to $O(n^6)$ for the classic G\"{a}rtner graph kernel \cite{DBLP:conf/colt/GartnerFW03}. Key to achieving this efficiency is that we compute the required Shannon entropy of the random walk for our kernels with $O(n^2)$ operations. This computational strategy enables our subgraph kernels to easily scale up to graphs of reasonably large sizes and thus overcome the size limits arising in state-of-the-art graph kernels. Experiments on standard bioinformatics and computer vision graph datasets demonstrate the effectiveness and efficiency of our new subgraph kernels.

Fast Depth-Based Subgraph Kernels for Unattributed Graphs

Lu Bai (1), Edwin R. Hancock (2)

(1) School of Information, Central University of Finance and Economics, Beijing, China.

(2) Department of Computer Science, The University of York, York YO10 5DD, UK
{lu,erh}@cs.york.ac.uk

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

***Highlights (for review)**

We investigate two fast subgraph kernels based on a depth-based representation.

Our kernels gauge
depth information rooted at a vertex for a graph.

The time complexities for the two kernels are O(n^2) and O(n^3) respectively.

We evaluate the performance of our subgraph kernels on standard graph datasets.

We demonstrate the effectiveness of the proposed subgraph kernels.

Acknowledgement

Dear reviewers

Thank you very much for your constructive suggestions. These have helped us to improve the paper. Below we describe our responses to each of the suggestions.

Best wishes

Authors

Revisions

Reviewer # 3

The reviewer would like to make sure that this paper will be a significant step forward comparing to the previous contributions and has some suggestions.

Comment 1: More diverse data sets should be included in the evaluation. NC I1, NC I109 PPIs, PTC (MR) that were used in "Attributed Graph kernels using the Jensen-Tsallis q-Differences" should be added. Authors are also requested to include all sets that they used previously to validate methods in their past publications.

Our revision: We have included some other datasets used in our previous paper in the revised manuscript, e.g., the NCI1, NCI109, PPIs, COIL5, Shock, CATH2, PTC(MR) and GatorBait dataset. See details in Section 5.1 and 5.2.

Comment 2: Along with the expanded set of data for testing, more methods should be added for comparison. The reviewer is interested in the inclusion of JT1 and JT2 or even more methods that authors recently published. The reason is that the results for JT1 and JT2 that listed in Table 2 of the paper mentioned above yield accuracies of 85.1% and 85.5% for MUTAG data set, whereas the JSSK and ISK proposed in this manuscript yield lower, namely: 83.77% and 84.66% accuracies for the same data set. The accuracy for JT1 and JT2 are also above 85 for NC I1 and NC I109 data sets, but these data sets were not included in this manuscript. Since accuracy measures for JSSK and ISK are lower than 83% for the other data sets listed in Tab.2, the reviewer does not understand why JSSK and ISK are proposed as new and perhaps more accurate methods that those previously proposed by the authors.

Our revision: We have used our previous Jensen-Tsallis kernel (ECML-PKDD, 2014) and the quantum Jensen-Shannon kernel (Pattern Recognition, 2015) for comparisons. The evaluation results can be found in Table 2 and Table 3. Moreover, discussion of the new evaluation can be found in Section 5.2-"Discussions and Analysis".

Comment 3: After all methods are tested using an expanded dataset; a statistical analysis should be performed to single out the best method. Right now, the authors use the average accuracy as a

performance measure. However, the average accuracy itself may not be sufficient since multiple methods are tested on multiple sets of data. In addition, the performance of each method has a spread. Since the average accuracy of JSSK is not always higher than the accuracy of ISK (Tab.2 in the manuscript) a statistical analysis is necessary.

Our revision: We not only provide the average classification accuracy (through 10-fold cross validation) but also provide the standard error of each kernel on each dataset. See details in Table 2. Moreover, through Table 2, we observe that different kernels perform differently on different datasets. To demonstrate the best kernel over all datasets, for each kernel, we also compute the average classification accuracy (associated with standard error) from the accuracies over all datasets. Details can be found in Section 5.2-" Statistical Analysis". We demonstrate that our ISK kernel is the best kernel in terms of either the classification accuracy and the performance stability (based on the standard error) over all datasets.

Reviewer # 4

The paper is well-written and provides a strong contribution. However, several issues should be addressed in a revision:

Comment 1: The kernel captures structural properties and is focused on unattributed graphs. A possible extension to attributed graphs is mentioned in the conclusions. To better embed the contribution into the literature, the attributed case should be discussed in the introduction together with available kernels / dissimilarity measures.

Our revision: We add some new contents of R-convolution kernels on attributed graphs, see details in the second paragraph of Section 1.1.

Comment 2: Citations are missing for the bioinformatics graph databases (MUTAG, D&D, ENZYMES). Furthermore, stateof-the-art classification results should be indicated for the bioinformatics as well as the computer vision databases in order to better evaluate the impact of the proposed kernels.

Our revision: We have fixed the citation problem for the bioinformatics graph datasets.

Comment 3: In Section 4.4 it is stated that the ISK is equivalent to the all subgraph kernel. However, ISK considers special types of subgraphs (centred around a vertex) and it seems possible that the same subgraph is counted several times in equation 22 (centred around different vertices). On the other hand, the all subgraph kernel considers arbitrary subgraphs and counts each subgraph pair only once. Please clarify.

Our revision: We have improve the description of the relationship between the two kernels more precise. For an instance, we replace the sentence "The entropic isomorphism kernel is equivalent to the all subgraph kernel" as " We show the equivalence between the entropic isomorphism kernel and the all subgraph kernel. " (see details in the second paragraph of Section 4.4). Moreover, in the last paragraph, we also emphasize the difference between the two kernels.

Comment 4: Minor issues:

- page 3: "alternative graph kernels that Specifically from the R-convolution framework" (specifically)
- equation 3: "P_G(V)" => "P_G(v)"
- equations 7 and 9: "(u,v) subset N^K" => "(u,v) subset N^K times N^K"
- below equation 13: "P=(p_1,…,p_M)" and "Q=(q_1,…,q_M)" seem to have a wrong cardinality M
- page 21: "the average of the geodesic distances to the all other points" (to all other)
- page 24: "Moreover, the efficiency of the JSSK kernel is also slower" => ISK kernel

Our revision: We have fixed the mentioned typos in the manuscript.

Fast Depth-Based Subgraph Kernels for Unattributed Graphs

Lu Bai[1*], Edwin R. Hancock[2**]

[1] *School of Information, Central University of Finance and Economics, Beijing, China.*
[2] *the Department of Computer Science, University of York, York, UK.*

Abstract

In this paper, we investigate two fast subgraph kernels based on a depth-based representation of graph-structure. Both methods gauge depth information through a family of K-layer expansion subgraphs rooted at a vertex [1]. The first method commences by computing a centroid-based complexity trace for each graph, using a depth-based representation rooted at the centroid vertex that has minimum shortest path length variance to the remaining vertices [2]. This subgraph kernel is computed by measuring the Jensen-Shannon divergence between centroid-based complexity entropy traces. The second method, on the other hand, computes a depth-based representation around each vertex in turn. The corresponding subgraph kernel is computed using isomorphisms tests to compare the depth-based representation rooted at each vertex in turn. For graphs with n vertices, the time complexities for the two new kernels are $O(n^2)$ and $O(n^3)$ respectively, in contrast to $O(n^6)$ for the classic Gärtner graph kernel [3]. Key to achieving this efficiency is that we compute the required Shannon entropy of the random walk for our kernels with $O(n^2)$ operations. This computational strategy enables our subgraph kernels to easily scale up to graphs of reasonably large sizes and thus overcome the size limits arising in state-of-the-art graph kernels. Experiments on standard bioinformatics and computer vision graph datasets demonstrate the effectiveness and efficiency of our new subgraph kernels.

Keywords: Depth-based representations, entropy, graph kernels, the Jensen-Shannon divergence, graph isomorphism tests.

*Email address: lu@cs.york.ac.uk and bailu69@hotmail.com.
**Edwin R. Hancock is supported by a Royal Society Wolfson Research Merit Award.

1. Introduction

There has recently been an increasing interest in learning and mining data using graph structures. Application include a) view-based object recognition [4], b) bioinformatics [5, 6] (e.g., classifying proteins into different families, classifying tissue samples), and c) social networks (e.g., classifying users based on their feeds on Twitter, Facebook, etc.). One challenge arising in classifying graphs is how to convert the discrete graph structures into numeric features or efficiently compute similarities between graphs for classification. One way to address this problem is to use graph kernels.

1.1. Graph Kernels

Graph kernels can characterize graph features in an explicit high dimensional space and thus have the capability of preserving graph structures. A number of graph kernels have been defined in the literature. Generally speaking, most existing graph kernels are usually formulated in terms of instances of the R-convolution kernel family developed by Haussler [5]. R-convolution is a generic way for defining graph kernels based on comparing all pairs of decomposed subgraphs. Specifically, all available graph decompositions can be used to define a kernel, e.g., the graph kernel based on comparing all pairs of decomposed a) walks, b) paths and c) restricted subgraph or subtree structures. With this scenario, Kashima et al. [7] have proposed a random walk kernel by comparing pairs of isomorphic random walks in a pair of graphs. The main drawback of the random walk kernel is the notorious tottering problem. This occurs when a random walk on a graph moves in one direction and then immediately returns to the starting position through the same vertices and edges possibly multiple times. To overcome this shortcoming, Borgwardt et al. [8] have proposed a shortest path kernel by counting the numbers of pairwise shortest paths having the same length in a pair of graphs. Aziz et al. [9] have defined a backtrackless kernel using the cycles identified by the Ihara zeta function [10] in a pair of graphs. The method overcomes the tottering problem using backtrackless substructures, i.e., the shortest paths or cycles in graphs. Unfortunately, shortest paths and cycles are structurally simple, and reflect limited topology information. Moreover, the computational efficiency of the two kernels also tends to be burdensome for graphs of large sizes, e.g., a graph having more than one thousand vertices.

To address the problem of inefficiency, Shervashidze et al. [5] have developed a fast subtree kernel by comparing pairs of subtrees identified by the Weisfeiler-Lehman (WL) algorithm. Unfortunately, like the random walk kernel, the WL isomorphism based subtree kernel also suffers from tottering. This is because the subtrees identified by the WL algorithm may also include several copies of the same pairwise vertices connected by the same edge. Furthermore, Costa and Grave [11] have defined a neighborhood subgraph pairwise distance kernel by counting the number of pairwise isomorphic neighborhood subgraphs. Both the WL subtree and neighborhood subgraph kernels can be computed in polynomial time. Some alternative graph kernels that specifically from the R-convolution framework include a) the segmentation graph kernel developed by Harchaoui and Bach [12], b) the point cloud kernel developed by Bach [13], c) the subgraph matching kernel developed by Kriege and Mutzel [14], and d) the (hyper)graph kernel based on directed subtree isomorphism tests developed and described in our previous work [15]. Moreover, it is important to note that, some of the aforementioned R-convolution kernels can accommodate attributed graphs too (i.e., these kernels can accommodate the attributed information residing on the vertices or edges). They can thus capture more characteristics that encapsulate label information on the vertices and edges [14]. Examples include the WL subtree kernel [5], the shortest path kernel [8], the random walk kernel [7], the subgraph matching kernel [14], and the (hyper)graph kernel [15].

One significant drawback of R-convolution kernels is that they compromise to use substructures of limited size, which only roughly capture topological arrangements of a graph. Though this strategy avoids the notorious inefficiency of R-convolution kernels when using large substructures, the limited size can only reflect restricted topological characteristics of a graph. Moreover, some R-convolution kernels still require significant computational overheads for large graphs (e.g., graphs having thousands of vertices).

An alternative way to construct a kernel is to measure the mutual information between pairs of graphs using the classical Jensen-Shannon divergence. In probability theory, the Jensen-Shannon divergence is a dissimilarity measure between probability distributions in terms of the nonextensive entropy difference associated with the probability distributions [16]. It is not only symmetric but also always well defined and bounded. In our previous work [4], we have used the classical Jensen-Shannon divergence to define a Jensen-Shannon kernel for graphs. Here, the

Jensen-Shannon divergence between a pair of graphs is defined in terms of the entropy difference between the entropy of a composite graph structure and that of the individual graphs. Unlike the R-convolution kernels, the entropy associated with a probability distribution of an individual graph can be computed without decomposing the graph into substructures. Therefore, the computation of the Jensen-Shannon graph kernel between a pair of graphs avoids burdensome (dis)similarity measurements involved in comparing all substructure pairs. Unfortunately, the existing Jensen-Shannon graph kernel can only capture the global similarity between a pair of graphs, and cannot distinguish the basis of the interior topological information. Furthermore, the required entropy that must be calculated for the composition of a pair of graphs is obtained from the product graph. The vertex number of the product graph is the multiple of the vertex numbers of the pair of graphs being compared. As a result, the entropy difference is dominated by that of the product graph when the graphs being compared are large.

To overcome the shortcomings of existing graph kernels, in this paper we aim to develop novel and fast subgraph kernels. Our new kernels are based on a rapidly computed depth-based graph representation.

1.2. Depth-Based Representations

Depth-based representations have been widely used for characterizing undirected graphs [17]. One approach to computing a depth-based representation for a graph is based on an information content flow through a family of K-layer expansion subgraphs [1]. These subgraphs can be located from a vertex and have a maximum topology distance K from the vertex to the remaining vertices. Following this approach, Escolano et al. [1] have shown how to compute the thermodynamic based depth complexity for a graph. This is done by measuring the heat flow complexities of expansion subgraphs around the vertices of the graph. Unfortunately, the heat flow complexity measure for a (sub)graph having n vertices requires time complexity $O(n^5)$. As a result, the thermodynamic depth complexity measure cannot be efficiently computed. To overcome this shortcoming, Bai and Hancock [2, 18, 19] have developed a centroid-based complexity trace from a centroid vertex that has the minimum variance of shortest path lengths to the remaining vertices. This depth-based representation is computed around the centroid vertex, and decomposes a graph into a family of

K-layer centroid expansion subgraphs that has a greatest shortest path length K rooted from the centroid vertex. The resulting complexity trace vector is computed by measuring the entropies of the expansion subgraphs. The centroid based method can be computed efficiently. The reason for this is that the entropy based complexity measures are computed on a small set of expansion subgraphs rooted at the centroid vertex, and can be computed in polynomial time.

Unfortunately, the centroid-based complexity trace may generate information loss for a graph structure. This is because the complexity trace vector of a graph can be viewed as an embedding vector, embedding a graph into a vector tends to approximate the structural correlations into a low dimensional space. One way to overcome the problem is to kernelize the embedding vectors (i.e., the complexity trace vectors) of graphs as a kernel function that represents graph structure in a high dimensional space and thus better preserves graph structure. Furthermore, since the centroid vertex is identified through a global analysis of the shortest path length distribution, the centroid expansion subgraphs provide a fine representation of graph structure. As a result, the centroid-based complexity trace and its required centroid expansion subgraphs offer us a potential way of defining a subgraph kernel. Unfortunately, the subgraphs of increasing layer size K tend to be the global graph (i.e., the largest layer subgraph is the graph itself), and straightforwardly measuring the (dis)similarity between wholes graphs usually requires burdensome computations.

1.3. Contributions

The aim of this paper is to develop fast subgraph kernels, that can not only be efficiently computed for large graphs but can also capture rich topological arrangement information contained within graphs. To this end, we investigate how to kernelize a depth-based representation of graphs. The contributions of this paper are twofold.

First, we develop a new depth-based subgraph kernel, namely the Jensen-Shannon subgraph kernel. This is done by measuring the Jensen-Shannon divergence between depth-based representations rooted at the centroid vertices [2]. To this end, we commence by computing the centroid-based complexity trace developed in our previous work and described in [2, 18, 19]. The advantage of using the complexity trace to characterize graphs is that it not only reflects dominant depth complexity information around the centroid vertex for a graph but also represents the graph in a high

dimensional space. This is because the centroid-based complexity trace for a graph encapsulates information flow from the centroid vertex to the global graph using entropy measures. By contrast, existing entropy measures [21, 22, 23] or the depth complexity measures [17, 1] only provide us with an uni-valued complexity measure for a graph. They thus reflect limited graph characteristics. With a pair of graphs and their centroid-based complexity traces to hand, the Jensen-Shannon subgraph kernel can be computed by measuring the Jensen-Shannon divergence measure developed in [4, 20] between the complexity traces, i.e., we compute the divergence between the entropies for each pair of K-layer centroid expansion subgraphs derived from the centroid vertices. Furthermore, to overcome the afore mentioned problem arising in our previous Jensen-Shannon divergence measure (i.e., the product graph of large size may dominant the kernel value), we propose to compute the divergence for a pair of (sub)graphs based on the entropy difference between the original (sub)graphs and a disjoint union formed by the (sub)graphs (i.e., a composite structure of (sub)graphs). In other word, we use the disjoint union as the composite structure, instead of the product graph. For a pair of (sub)graphs, the size of their disjoint union is only the sum of their sizes. We thus overcome the shortcoming of dominating kernel value using the product graph of large size. Compared to our previous centroid-based complexity traces [2, 18, 19] and the Jensen-Shannon kernel [4, 20] for graphs, our new subgraph kernel has the following advantages. a) Compared to the original centroid-based complexity traces that embed graphs into a vector space, the new kernel is computed by kernelising the complexity trace vectors using the Jensen-Shannon divergence. The new kernel can characterize graphs in a higher dimensional space and thus better preserves graph structures. b) Compared to the Jensen-Shannon graph kernel, the new kernel computes the Jensen-Shannon divergence between each pair of K-layer centroid expansion subgraphs including the global graphs, i.e., the largest layer subgraphs are the global graphs themselves. By contrast, the Jensen-Shannon graph kernel only computes the divergence measure between the whole graphs. As a result, the new kernel overcomes the restriction of only capturing similarity on whole graphs that arises in the Jensen-Shannon graph kernel [4].

Second, we develop another new depth-based subgraph kernel, namely the entropic isomorphism kernel, by entropically measuring the isomorphisms between the depth-based representations for all vertices. Specifically, we compute a depth-based representation around each vertex, by

computing the entropies on the expansion subgraphs that are derived from that vertex. We compute the resulting kernel by performing entropy-based isomorphism tests between pairwise expansion subgraphs for a pair of graphs. Unlike our Jensen-Shannon subgraph kernel, the entropic isomorphism kernel computed from the depth-based representations reflects the depth information from any vertex. By contrast, the Jensen-Shannon subgraph kernel only reflects the depth information from the centroid vertex.

Both our Jensen-Shannon subgraph kernel and the entropic isomorphism kernel can be efficiently computed. One key reason for the efficiency is that the required Shannon entropy for the depth-based representation only requires computational complexity that is quadratic in vertex number (see details in Section 2.1). Furthermore, unlike the existing R-convolution kernels that only reflect restricted topological characteristics, our new kernels also capture rich depth-based topological arrangement information. We also demonstrate the relationship between the new entropic isomorphism kernel and the all subgraph kernel. Thus, we give a theoretical reason for the effectiveness of the new kernel. Finally, we empirically demonstrate the effectiveness and efficiency for both the Jensen-Shannon subgraph kernel and the entropic isomorphism kernel on standard graph datasets abstracted from computer vision and bioinformatics databases.

The remainder of this paper is organized as follows. Section 2 describes two depth-based representations, namely the centroid depth-based complexity trace and the h-layer depth-based representation. Section 3 defines the Jensen-Shannon subgraph kernel. Section 4 shows how the all depth-based subgraph kernel is constructed. Section 5 provides our experimental evaluation. Finally, Section 6 concludes our work.

2. Depth-Based Representations of Graphs

In this section, we introduce some preliminary concepts that will be used for developing the work presented in this paper. To this end, we commence by introducing a fast Shannon entropy measure for a graph associated with the steady state random walk. Second, we review how to compute a depth-based representation for a graph from the centroid vertex, i.e., the centroid-based complexity trace developed in [2], by measuring the Shannon entropy on a family of centroid expansion subgraphs derived from the centroid vertex. Compared to existing depth-based complexity

measures [1, 17], the centroid-based complexity trace reflects richer complexity information in a high dimensional space. However, it only reflects the depth complexity information from the centroid vertex. To address this problem, we finally develop an alternative depth-based representation for a graph from each vertex, namely the h-layer depth-based representation.

2.1. The Random Walk Shannon Graph Entropy

We commence by reviewing the fast Shannon entropy of a (sub)graph that has been developed in [2], using the steady state random walk. The entropy will be used to compute the depth-based representation. Assume an undirected graph $G(V, E)$, where V is the set of vertices and $E \subseteq V \times V$ is the set of undirected edges. The neighbourhood $\mathcal{N}(v)$ of a vertex $v \in V$ is the set of vertices to which v is connected by an edge, and is defined as

$$\mathcal{N}(v) = \{u|(v, u) \in E\}. \tag{1}$$

The degree matrix of $G(V, E)$ is a diagonal matrix D with elements

$$D(v, v) = d(v) = |\mathcal{N}(v)|, \tag{2}$$

where $d(v)$ is the degree of vertex v. The probability of a steady state random walk visiting the vertex v in $G(V, E)$ is

$$P_G(v) = \frac{D(v, v)}{\sum_{u \in V} D(u, u)} \tag{3}$$

The Shannon entropy of $G(V, E)$ associated with the steady state random walk is

$$H_S(G) = -\sum_{v \in V} P_G(v) \log P_G(v). \tag{4}$$

Time Complexity: For a graph $G(V, E)$ having n vertices, the Shannon entropy $H_S(G)$ requires time complexity $O(n^2)$. This is because the degree matrix D of $G(V, E)$ can be computed by visiting all pairs of vertices. Thus the entropy $H_S(G)$ can be directly computed by visiting all the n^2 pairs of vertices.

The Shannon entropy associated with the steady state random walk allows us to efficiently capture characteristics of graphs, and can hence be used to develop a new fast entropy based similarity

measure for graphs. By contrast, both the Shannon entropy associated with information functionals developed by Dehmer in [21] and the von Neumann entropy developed by Anand et al. in [22] and Passerini and Severini in [23] require time complexity $O(n^3)$, since they require the spectrum decomposition of $G(V,E)$. Furthermore, from Eq.(4), we observe that for the Shannon entropy H_S, vertices with large degree will dominate the entropy value. Thus, the proposed Shannon entropy H_S is suited to characterizing graphs possessing a group of highly interconnected vertices, i.e., a dominant cluster.

2.2. The Centroid-Based Complexity Trace of A Graph

In this subsection, we review how to compute a centroid-based complexity trace for a graph developed in [2]. We commence by identifying the centroid vertex of a graph. Given an undirected graph $G(V,E)$, the shortest path matrix S_G can be computed by using Dijkstra's algorithm [24]. Each element $S_G(v,u)$ of S_G represents the shortest path length between vertices v and u. The average-shortest-path vector S_V for $G(V,E)$ is a vector with the same vertex order as S_G and has element

$$S_V(v) = \frac{1}{|V|}\sum_{u\in V} S_G(v,u), \tag{5}$$

which represents the average shortest path length from vertex v to the remaining vertices. The centroid vertex $\hat{v}_C$ for $G(V,E)$ is the vertex that has the minimum variance of shortest path lengths to the remaining vertices, and its vertex-index is

$$\hat{v}_C = \arg\min_{v} \sum_{u\in V} [S_G(v,u) - S_V(v)]^2. \tag{6}$$

Let $N_{\hat{v}_C}^K$ be a subset of V satisfying $N_{\hat{v}_C}^K = \{u \in V \mid S_G(\hat{v}_C,u) \leq K\}$. For $G(V,E)$ with the centroid vertex $\hat{v}_C$, the K-layer centroid expansion subgraph $\mathcal{G}_K(\mathcal{V}_K;\mathcal{E}_K)$ is

$$\begin{cases} \mathcal{V}_K &= \{u \in N_{\hat{v}_C}^K\}; \\ \mathcal{E}_K &= \{(u,v) \subset N_{\hat{v}_C}^K \times N_{\hat{v}_C}^K \mid (u,v) \in E\}. \end{cases} \tag{7}$$

The number of centroid expansion subgraphs is equal to the greatest length L of the shortest paths from the centroid vertex to the remaining vertices of the graph $G(V,E)$. The L-layer expansion

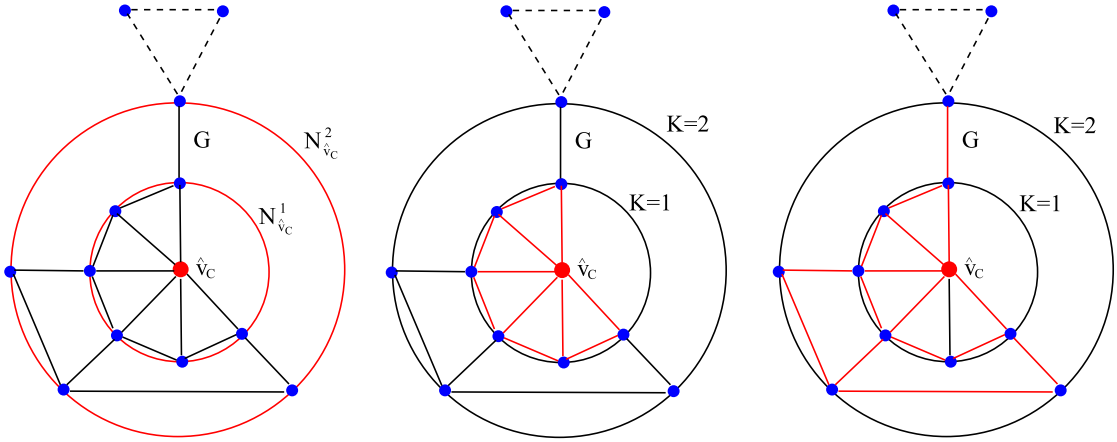


Figure 1: The left-most figure shows the determination of K-layer centroid expansion subgraphs for a graph $G(V,E)$ which hold $|N^1_{\hat{v}_C}| = 6$ and $|N^2_{\hat{v}_C}| = 10$ vertices. While the middle and the right-most figure show the corresponding 1-layer and 2-layer subgraphs regarding the centroid vertex $\hat{v}_C$, and are depicted by red-colored edges. In this example, the vertices of different K-layer subgraphs regarding the centroid vertex $\hat{v}_C$ are calculated by Eq.(6), and pairwise vertices possess the same connection information in the original graph $G(V,E)$.

subgraph is the graph $G(V,E)$ itself. An example of the generation of a K-layer subgraph for a graph $G(V,E)$ is shown in Fig.1.

Definition 2.1 (Centroid-based complexity trace): Let the family of centroid expansion subgraphs for $G(V,E)$ be $\{\mathcal{G}_1, \cdots, \mathcal{G}_K, \cdots, \mathcal{G}_L\}$. We measure the entropies of the subgraphs and establish the centroid-based complexity trace D_C for $G(V,E)$ as

$$DB_C(G) = \{H_S(\mathcal{G}_1), \cdots, H_S(\mathcal{G}_K), \cdots, H_S(\mathcal{G}_L)\}, \tag{8}$$

where $\cdots, H_S(\mathcal{G}_K)$ is the Shannon entropy associated with the steady state random walk on the K-layer centroid expansion subgraph $\mathcal{G}_K$. □

For a graph $G(V,E)$ having n vertices, computing the centroid depth-based complexity trace $D_C(G)$ of $G(V,E)$ requires time complexity $O(Ln^2)$. This follows the definition in Eq.(7). For a graph $G(V,E)$, the Dijkstra's algorithm requires time complexity $O(n^2)$. Computing the Shannon entropies of the L K-layer centroid expansion subgraphs requires time complexity $O(Ln^2)$. Hence, the overall time complexity is $O(Ln^2)$.

2.3. The h-Layer Depth-Based Representation of A Graph

In this subsection, we develop the centroid-based complexity trace further by defining a h-layer depth-based representation around each vertex for a graph (i.e., a depth-based complexity trace around each vertex). Unlike the centroid-based complexity trace that only reflects the depth complexity information from the centroid vertex, for all vertices the h-layer depth-based representations reflect the depth complexity information from any vertex.

For an undirected graph $G(V,E)$ and its shortest path matrix S_G, let N_v^K be a subset of V satisfying $N_v^K = \{u \in V \mid S_G(v,u) \leq K\}$. For $G(V,E)$, the K-layer expansion subgraph $\mathcal{G}_v^K(\mathcal{V}_v^K;\mathcal{E}_v^K)$ around the vertex v is

$$\begin{cases} \mathcal{V}_v^K &= \{u \in N_v^K\}; \\ \mathcal{E}_v^K &= \{(u,v) \subset N_v^K \times N_v^K \mid (u,v) \in E\}. \end{cases} \tag{9}$$

Let L_{max} be the greatest length of the shortest paths from v to the remaining vertices of $G(V,E)$. If $L_v \geq L_{max}$, the L_v-layer expansion subgraph is $G(V,E)$ itself.

Definition 2.2 (h-layer depth-based representation): For a graph $G(V,E)$ and a vertex $v \in V$, the h-layer depth-based representation around the vertex v of $G(V,E)$ is a h dimensional vector

$$DB_G^h(v) = [H_S(\mathcal{G}_v^1), \cdots, H_S(\mathcal{G}_v^K), \cdots, H_S(\mathcal{G}_v^h)]^T \tag{10}$$

where h ($h \leq L_v$) is the length of the shortest paths from the vertex v to the remaining vertices in $G(V,E)$, $\mathcal{G}_v^K(\mathcal{V}_v^K;\mathcal{E}_v^K)$ ($K \leq h$) is the K-layer expansion subgraph of $G(V,E)$ around the vertex v, and $H_S(\mathcal{G}_v^K)$ is the Shannon entropy of $\mathcal{G}_v^K$ defined in Eq.(4). □

For a graph $G(V,E)$ having n vertices, computing the h-layer depth-based representation $D_G^h(v)$ of $G(V,E)$ around all vertices $v \in V$ requires time complexity $O(hn^3)$. This follows the definition in Eq.(9). For a graph $G(V,E)$, the Dijkstra algorithm requires time complexity $O(n^2)$. Computing the Shannon entropies of the h K-layer expansion subgraphs, which are derived from v, requires time complexity $O(hn^2)$. Hence, the overall time complexity of computing the h-layer depth-based representations for n vertices is $O(hn^3)$.

3. A Jensen-Shannon Subgraph Kernel

In this section, we develop a fast subgraph kernel using the Jensen-Shannon divergence. We commence by showing how to compute the Jensen-Shannon divergence for (sub)graphs. For a pair of graphs, we develop the new subgraph kernel by measuring the Jensen-Shannon divergence between the subgraph entropies from their centroid-based complexity traces.

3.1. A Composite Entropy of Graphs Through The Disjoint Union Graph

To compute the Jensen-Shannon divergence between a pair of graphs, we require a composite structure for the graphs. In our previous work [4], we have used the product union to construct the composite graph. Unfortunately, constructing a product graph is computationally burdensome. Furthermore, the number of vertices for a product graph can be large. Thus, the product graph will dominate the computation of the Jensen-Shannon divergence. To overcome this problem, we propose to use a different strategy for constructing a composite structure $G_p \oplus G_q$ for a pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$. We turn to the disjoint union for constructing our composite structure. According to [25], the disjoint union graph of $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ is

$$G_{DU} = G_p \cup G_q = \{V_p \cup V_q, E_p \cup E_q\}. \tag{11}$$

Through Eq.(11), we observe that the size of the disjoint union graph G_{DU} for $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ is $|V_p| + |V_q|$. By contrast, the size of the product graph for $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ is $|V_p||V_q|$. In other word, the size of the disjoint union graph for a pair of graphs is much smaller than their product graph.

Let graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ be the connected components of the disjoint union graph $G_{DU}(V_{DU}, E_{DU})$, then we compute the relative sizes of the connected components as

$$\rho_p = \frac{|V(G_p)|}{|V(G_{DU})|} = \frac{|V(G_p)|}{(|V(G_p)| + |V(G_q)|)}.$$

and

$$\rho_q = \frac{|V(G_q)|}{|V(G_{DU})|} = \frac{|V(G_q)|}{(|V(G_p)| + |V(G_q)|)}.$$

The entropy (i.e., the composite entropy) [26] of G_{DU} is then

$$H_S(G_{DU}) = \rho_p H_S(G_p) + \rho_q H_S(G_q). \tag{12}$$

Here the entropy function H_S is the Shannon entropy $H_S(\cdot)$ defined in Eq.(4).

3.2. A Jensen-Shannon Divergence on Graphs

The classical Jensen-Shannon divergence is a nonextensive mutual information dissimilarity measure defined on probability distributions. Assume $M_+^1(\chi)$ is a set of probability distributions where χ is a set provided with some $\sigma - algebra$ of measurable subsets, the Jensen-Shannon divergence $D_{JS} : M_+^1(\chi) \times M_+^1(\chi) \rightarrow R$ between the probability distributions P and Q, is negative definite (**nd**) with the following function [23]:

$$\begin{aligned} D_{JS}(P,Q) &= \frac{1}{2}D_{KL}(P||M) + \frac{1}{2}D_{KL}(Q||M) \\ &= \frac{1}{2}\int_{\chi} \ln(\frac{dP}{dM})dP + \frac{1}{2}\int_{\chi} \ln(\frac{dQ}{dM})dQ, \end{aligned} \tag{13}$$

where $M = \frac{P+Q}{2}$ and $D_{KL}(P||M) = \int_{\chi} \ln(\frac{dP}{dM})dP$ is the Kullback-Leibler divergence between P and M. If χ is countable, i.e., $P = (p_1, p_2, \ldots, p_N)$ and $Q = (q_1, q_2, \ldots, q_N)$ are two discrete probability distributions, a more general definition is

$$D_{JS}(P,Q) = H_S(\frac{P+Q}{2}) - \frac{H_S(P) + H_S(Q)}{2}, \tag{14}$$

where $H_S(P) = \sum_{m=1}^{M} p_m \log p_m$ is a Shannon entropy of the probability distribution P. We defined a Jensen-Shannon divergence measure for a pair of graphs. Given a pair of graphs $G_p(V_p, E_q)$ and $G_q(V_q, E_q)$, the Jensen-Shannon divergence for them is

$$D_{JS}(G_p, G_q) = H_S(G_p \oplus G_q) - \frac{H_S(G_p) + H_S(G_q)}{2}. \tag{15}$$

where $G_p \oplus G_q$ is the composite structure formed by the graphs $G_p(V_p, E_q)$ and $G_q(V_q, E_q)$. Here we use the disjoint union defined in Sec.3.1 as the composite structure, and the entropy function $H_S(\cdot)$ is the Shannon entropy associated with the steady state random walk defined in Eq.(4).

With the Jesnen-Shannon divergence for graphs defined in Eq.(14) to hand, we define a Jensen-Shannon diffusion graph kernel k_{JS}: $G_p \times G_q \rightarrow R$ with the kernel value

$$k_{JS}(G_p, G_q) = \exp(-\lambda D_{JS}(G_p, G_q)). \tag{16}$$

where λ is a decay factor and satisfies $0 < \lambda \leq 1$. Note that, unlike the Jensen-Shannon divergence which is a dissimilarity measure, the Jensen-Shannon diffusion kernel is an information theoretic similarity measure of a pair of graphs.

Lemma 3.1. *The Jensen-Shannon diffusion kernel defined in Eq.(16) is positive definite (**pd**).*

Proof. This follows the definition in [27]. If a similarity or dissimilarity measure $s_G(G_p, G_q)$ between a pair of graphs G_p and G_q is symmetrical, then a diffusion kernel $k_s = \exp(\lambda s_G(G_p, G_q))$ or $k_s = \exp(-\lambda s_G(G_p, G_q))$ associated with the (dis)similarity measure $s_G(G_p, G_q)$ is **pd**. ■

Note that, a positive definite graph kernel is often called a *valid kernel*. Clearly, imposing a graph kernel to be positive definite restricts the broad class of similarity-based graph kernels into a small group of valid kernels. However, it has been observed that the property of positive definiteness is crucial for the definition of kernel machines and turns out to implicate a considerable number of theoretical merits associated with graph kernels [28].

For a pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ each of which has n vertices, computing the Jensen-Shannon diffusion kernel $k_{JS}(G_p, G_q)$ in Eq.(16) requires $O(n^2)$ operations. This is because both $H_S(G_p)$ and $H_S(G_p)$ require time complexity $O(n^2)$. The disjoint union graph entropy $H_S(G_{DU})$ can be directly computed based on $H_S(G_p)$ and $H_S(G_p)$ according to Eq.(12). As a result, the Jensen-Shannon diffusion kernel $k_{JS}(G_p, G_q)$ requires time complexity $O(n^2)$.

3.3. The Jensen-Shannon Subgraph Kernel

In this subsection, we develop a fast Jensen-Shannon subgraph kernel ($k_{\mathcal{JS}}$) as an information theoretic decomposition kernel. The proposed kernel $k_{\mathcal{JS}}$ is defined by kernelizing the centroid-based graph complexity traces. This is done by measuring the information content similarities for the K-layer subgraphs using the Jensen-Shannon divergence. For a graph $G(V, E)$, we commence by identifying the centroid vertex $\hat{v}_C$ using Eq.(6). Based on $\hat{v}_C$ we construct the K-layer centroid expansion subgraph $\mathcal{G}_K$ of $G(V, E)$ using Eq.(7). As we increase K from 1 to the greatest shortest path length L with respect to the centroid vertex $\hat{v}_C$, we obtain a family of centroid expansion subgraphs $\{\mathcal{G}_1, \cdots, \mathcal{G}_K, \cdots, \mathcal{G}_L\}$. We then measure the entropies of the subgraphs and establish the depth-based representation $DB_C(G)$ of $G(V, E)$ as $DB_C(G) = \{H_S(\mathcal{G}_1), \cdots, H_S(\mathcal{G}_K), \cdots, H_S(\mathcal{G}_L)\}$. For a pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$, we com-

pute a similarity measure between their depth-based representations $DB_C(G_p)$ and $DB_C(G_q)$ as follows

$$s(DB_C(G_p), DB_C(G_q)) = \sum_{K=1}^{L} s_H(H(\mathcal{G}_{p;K}), H(\mathcal{G}_{q;K})). \tag{17}$$

where $s_H(H(\mathcal{G}_{p;K}), H(\mathcal{G}_{q;K}))$ is an entropy-based similarity measure for the K-layer subgraphs $\mathcal{G}_{p;K}$ and $\mathcal{G}_{q;K}$ of $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$. By using the Jensen-Shannon diffusion kernel $k_{JS}(\cdot,\cdot)$ in Eq.(16) as the entropy-based similarity measure $s_H(\cdot,\cdot)$ in Eq.(17), the similarity between the depth-based representations $DB_C(G_p)$ and $DB_C(G_q)$ is formulated as the sum of the diffusion kernel measures for all the pairs of K-layer subgraphs of $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$.

Definition 3.1 (Jensen-Shannon subgraph kernel): Consider a pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$. The Jensen-Shannon subgraph kernel $k_{\mathcal{JS}}(G_p, G_q)$ is defined as

$$k_{\mathcal{JS}}(G_p, G_q) = s(DB_C(G_p), DB_C(G_q)) = \sum_{K=1}^{L} k_{JS}(\mathcal{G}_{p;K}, \mathcal{G}_{q;K}). \tag{18}$$

where $\mathcal{G}_{p;K}(\mathcal{V}_{p;K}, \mathcal{E}_{p;K})$ and $\mathcal{G}_{q;K}(\mathcal{V}_{q;K}, \mathcal{E}_{q;K})$ are the K-layer centroid expansion subgraphs of $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ rooted at their corresponding centroid vertices $\hat{v}_{p;C}$ and $\hat{v}_{q;C}$, respectively, and $k_{JS}(\mathcal{G}_{p;K}, \mathcal{G}_{q;K})$ is the Jensen-Shannon diffusion kernel between $\mathcal{G}_{p;K}(\mathcal{V}_{p;K}, \mathcal{E}_{p;K})$ and $\mathcal{G}_{q;K}(\mathcal{V}_{q;K}, \mathcal{E}_{q;K})$. According to Eq.(12), Eq.(15) and Eq.(16), $k_{JS}(\mathcal{G}_{p;K}, \mathcal{G}_{q;K})$ is

$$k_{JS}(\mathcal{G}_{p;K}, \mathcal{G}_{q;K}) = \exp\{\frac{2|\mathcal{V}_{p;K}| - |\mathcal{V}_{q;K}|}{2|\mathcal{V}_{p;K}| + 2|\mathcal{V}_{q;K}|}\lambda H(\mathcal{G}_{p;K}) + \frac{2|\mathcal{V}_{q;K}| - |\mathcal{V}_{p;K}|}{2|\mathcal{V}_{p;K}| + 2|\mathcal{V}_{q;K}|}\lambda H(\mathcal{G}_{q;K})\}. \tag{19}$$

Lemma 3.2. *The Jensen-Shannon subgraph kernel $k_{\mathcal{JS}}$ is **pd**.*

Proof. For all $\{c_1, \cdots, c_N\} \subseteq R$ and any N graphs $\{G_1, \cdots, G_N\}$ we have the following expression

$$\begin{aligned}\sum_{i,j=1}^{N} c_i c_j k_{\mathcal{JS}}(G_i, G_j) &= \sum_{i,j=1}^{N} c_i c_j \{\sum_{K=1}^{L} k_{JS}(\mathcal{G}_{i;K}, \mathcal{G}_{j;K})\} \\ &= \sum_{i,j=1}^{N} c_i c_j k_{JS}(\mathcal{G}_{i;1}, \mathcal{G}_{j;1}) + , ..., + \sum_{i,j=1}^{N} c_i c_j k_{JS}(\mathcal{G}_{i;L}, \mathcal{G}_{j;L}).\end{aligned}$$

Here for all $\{c_1, ..., c_N\} \subseteq R$ and any choice of the N subgraphs $\{\mathcal{G}_{1;K}, \cdots, \mathcal{G}_{N;K}\}$ which are the K-layer centroid expansion subgraphs of the N graphs $\{G_1, \cdots, G_N\}$, we have

$$\sum_{i,j=1}^{N} c_i c_j k_{JS}(\mathcal{G}_{i;K}, \mathcal{G}_{j;K}) \geq 0,$$

since k_{JS} is **pd** (Lemma 3.1). Therefore, we have

$$\sum_{i,j=1}^{N} c_i c_j k_{\mathcal{JS}}(G_i, G_j) \geq 0,$$

and the proposed Jensen-Shannon subgraph kernel is also **pd**. ■

Note that, for a pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ with different sizes, the longest layers of their expansion subgraphs could be different. Suppose that $\hat{v}_{C;p}$ and $\hat{v}_{C;q}$ are the centroid vertices of $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$, and the lengths of the greatest shortest paths from the centroid vertices $\hat{v}_{C;p}$ and $\hat{v}_{C;q}$ are L_p and L_q, respectively, where $L_p > L_q$. In practical computations, to balance the layer difference between the largest centroid expansion subgraphs of the two graphs, we use the graph $G_q(V_q, E_q)$ as the $(L_q + 1)$-layer to L_p-layer expansion subgraphs of $G_q(V_q, E_q)$. As a result, for a set of graphs $\{G_1, \ldots, G_s, \ldots, G_l, \ldots, G_N\}$ in which G_l has the greatest shortest path from the centroid vertex, we use each graph G_s itself as the $(L_s + 1)$-layer to L_l-layer expansion subgraphs.

3.4. Analysis of Computational Complexity

For a pair of graphs each of which has n vertices and L layer expansion subgraphs, the proposed Jensen-Shannon subgraph kernel requires time complexity $O(n^2)$. This is because computing the centroid-based representations requires time complexity $O(Ln^2)$. Computing the Jensen-Shannon diffusion kernel between the centroid depth-based representations requires time complexity $O(L)$. L usually tends to be $\sqrt[3]{n}$. As a result, the overall time complexity is $O(n^2)$. This indicates that for a pair of graphs the time complexity of the Jensen-Shannon subgraph kernel tends to be quadratic in the vertex number of the larger graph. Thus, the new subgraph kernel can be computed in a polynomial time.

3.5. Discussion

We make three observations regarding the Jensen-Shannon subgraph kernel. First our Jensen-Shannon subgraph kernel is equivalent to the similarity measure between depth-based representations of graphs. Since a depth-based representation of a graph $G(V, E)$ exhibits high dimensional depth-based entropy complexity characteristics via the centroid expansion subgraphs $\{\mathcal{G}_1, \cdots, \mathcal{G}_K, \cdots, \mathcal{G}_L\}$.

Our subgraph kernel $k_{\mathcal{JS}}$ captures richer complexity based information than that obtained from straightforwardly applying the Jensen-Shannon diffusion kernel to the original graphs. Second, the Jensen-Shannon subgraph kernel only compares pairs of subgraphs with the same layer size K. This avoids enumerating all pairs of subgraphs and renders the computation efficient. Third, for a pair of graphs, the Jensen-Shannon subgraph kernel can also efficiently measure the similarity of their L-layer subgraphs (i.e., the two graphs themselves). Hence, our Jensen-Shannon subgraph kernel overcomes the subgraph size restriction which commonly arises in existing R-convolution graph kernels.

4. An Entropic Isomorphism Kernel

In this section, we develop an entropic graph isomorphism kernel. We commence by defining an entropy-based isomorphism test for a pair of K-layer expansion subgraphs. Then we develop the new subgraph kernel by measuring the similarity measure between h-layer depth-based representations for a pair of graphs using the new isomorphism test.

4.1. An Entropy-Based Isomorphism Test

For a graph $G(V,E)$ and its vertices v and u, $\mathcal{G}_v^K(\mathcal{V}_v^K;\mathcal{E}_v^K)$ and $\mathcal{G}_u^K(\mathcal{V}_u^K;\mathcal{E}_u^K)$ are the corresponding K-layer expansion subgraphs around v and u defined by Eq.(9). We perform the following isomorphism test on $\mathcal{G}_v^K(\mathcal{V}_v^K;\mathcal{E}_v^K)$ and $\mathcal{G}_u^K(\mathcal{V}_u^K;\mathcal{E}_u^K)$ as

$$I(\mathcal{G}_v^K,\mathcal{G}_u^K)=\begin{cases}1 & \text{if } H_S(\mathcal{G}_v^K)=H_S(\mathcal{G}_u^K),\\ & |\mathcal{V}_v^K|=|\mathcal{V}_u^K|,\ |\mathcal{E}_v^K|=|\mathcal{E}_u^K|,\\ & \text{and } l_v=l_u=K,\\ 0 & \text{otherwise.}\end{cases} \tag{20}$$

where if $I(\mathcal{G}_v^K,\mathcal{G}_u^K)=1$, then $\mathcal{G}_v^K\simeq\mathcal{G}_u^K$ (i.e., $\mathcal{G}_v^K$ and $\mathcal{G}_u^K$ are isomorphic). Here, l_v and l_u are respectively the longest shortest path lengths of $\mathcal{G}_v^K$ and $\mathcal{G}_u^K$ from the vertices v and u.

For a pair of graphs each of which has n vertices, the proposed entropy-based isomorphism test requests time complexity $O(n^2)$. Because the test relies on computing the Shannon entropy associated with the steady state random walk, it has time complexity $O(n^2)$. This indicates that our entropy-based isomorphism test for a pair of graphs can be performed in a polynomial time.

4.2. The Entropic Isomorphism Kernel

In this subsection, we develop an entropic isomorphism kernel (k^h_{ISK}) using the entropy-based isomorphism test between the K-layer expansion subgraphs. We commence by developing a similarity measure between a pair of h-layer depth-based representations. For a vertex v_p of a graph $G_p(V_p, E_p)$ and a vertex v_q of a graph $G_q(V_q, E_q)$, we compute their h-layer depth-based representations $DB^h_{G_p}(v_p) = \{H_S(\mathcal{G}^1_{v;p}), \cdots, H_S(\mathcal{G}^K_{v;p}), \cdots, H_S(\mathcal{G}^h_{v;p})\}$ and $DB^h_{G_q}(v_q) = \{H_S(\mathcal{G}^1_{v;q}), \cdots, H_S(\mathcal{G}^K_{v;q}), \cdots, H_S(\mathcal{G}^h_{v;q})\}$ respectively. We compute the similarity measure between the h-layer depth-based representations $DB^h_{G_p}(v_p)$ and $DB^h_{G_q}(v_q)$ as

$$s_I(DB^h_{G_p}(v_p), DB^h_{G_q}(v_q)) = \sum_{K=1}^{h} I(\mathcal{G}^K_{v;p}, \mathcal{G}^K_{v;q}), \tag{21}$$

where $I(\mathcal{G}^K_{v;p}, \mathcal{G}^K_{v;q})$ is the entropy-based isomorphism test defined in Eq.(20).

Definition 4.1 (Entropic Isomorphism kernel): Consider $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ as a pair of sample graphs. The entropic isomorphism kernel k^h_{ISK} using the h-layer depth-based representations of graphs is defined as

$$\begin{aligned} k^h_{ISK}(G_p, G_q) &= \sum_{v_p \in V_p} \sum_{v_q \in V_q} s_I(D^h_{G_p}(v_p), D^h_{G_q}(v_q)) \\ &= \sum_{v_p \in V_p} \sum_{v_q \in V_q} \sum_{K=1}^{h} I(\mathcal{G}^K_{v;p}, \mathcal{G}^K_{v;q}). \end{aligned} \tag{22}$$

Intuitively, the entropic isomorphism kernel $k^{(h)}_{ISK}$ is **pd**, because it counts the number of isomorphic expansion subgraphs between each pair of h-layer depth-based representations. In other words, $k^{(h)}_{ISK}$ can be seen as an example of the classical R-convolution graph kernels by counting the number of isomorphic expansion subgraph pairs.

4.3. Analysis of Computational Complexity

For a pair of graphs each of which has n vertices, the entropic isomorphism kernel requires time complexity $O(n^3)$. This is because computing the h-layer depth-based representations for the graphs over all vertices requires time complexity $O(hn^3)$. Measuring the isomorphism based similarity between their h-layer depth-based representations requires time complexity $O(hn^2)$.

The layer h usually tends to be much smaller than n. As a result, the time complexity is $O(n^3)$. This indicates that the entropic isomorphism kernel can also be computed in a polynomial time, though this kernel may require more time complexity than the Jensen-Shannon subgraph kernel.

4.4. Relationship with the All Subgraph Kernel

In this subsection, we explore the relationship between our entropic isomorphism kernel and the classical all subgraph kernel. Let $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ be two graphs, the all subgraph kernel [3] is defined as

$$k_{subgraph}(G_p, G_q) = \sum_{S_p \sqsubseteq G_p} \sum_{S_q \sqsubseteq G_q} \delta(S_p, S_q), \tag{23}$$

where

$$\delta(S_p, S_q) = \begin{cases} 1 & \text{if } S_p \simeq S_q, \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

Here, δ is the Dirac kernel, that is, it is 1 if the arguments are equal and 0 otherwise (i.e., it is 1 if a pair of subgraphs are isomorphic and 0 otherwise).

We show the equivalence between the entropic isomorphism kernel k^h_{ISK} and the all subgraph kernel $k_{subgraph}$. To this end, we consider a pair of graphs as $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$. Let $\mathbb{G}^K_{v_p}$ and $\mathbb{G}^K_{v_q}$ be the expansion subgraph sets which contain all the K-layer ($1 \leq K \leq h$) expansion subgraphs around the vertices $v_p \in V_p$ and $v_p \in V_q$ respectively. Based on the definition in Eq.(20), for any pair of subgraphs $S_p(\mathcal{V}_p; \mathcal{E}_p)$ and $S_q(\mathcal{V}_q; \mathcal{E}_q)$ we rewrite Eq.(24) as

$$\delta^K_{(v,u)}(S_p, S_q) = \begin{cases} 1 & \text{if } H_S(S_p) = H_S(S_q), \\ & S_p \in \mathbb{G}^K_{v_p},\ S_q \in \mathbb{G}^K_{v_q}, \\ & |\mathcal{V}_p| = |\mathcal{V}_q|,\ |\mathcal{E}_p| = |\mathcal{E}_q|, \\ & \text{and } l_{v_p} = l_{v_q} = K, \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

where l_{v_p} and l_{v_q} are the longest shortest path lengths of S_p and S_q from the vertices v_p and v_q respectively, and $H_S(S_p)$ and $H_S(S_q)$ are the Shannon entropies associated with the steady state

random walks on S_p and S_q respectively. Associated with Eq.(25), the kernel k^h_{ISK} can be re-defined by re-writing Eq.(23) as

$$\begin{aligned} k^h_{ISK}(G_p, G_q) &= k_{subgraph}(G_p, G_q) \\ &= \sum_{K=1}^{h} \sum_{v_p \in V_p} \sum_{v_q \in V_q} \sum_{S_p \in \mathbb{G}^K_{v_p}} \sum_{S_q \in \mathbb{G}^K_{v_q}} \delta^K_{(v_p, v_q)}(S_p, S_q). \end{aligned} \tag{26}$$

Through Eq.(23) and Eq.(26), we observe that both the kernels $k_{subgraph}$ and k^h_{ISK} need to identify all pairs of isomorphic subgraphs. Moreover, for the kernels $k_{subgraph}$ and k^h_{ISK} each isomorphic subgraph pair adds an unit value to the kernel value. Thus, both the entropic isomorphism kernel and the all subgraph kernel count the number of isomorphic subgraph pairs and thus have equivalence.

Furthermore, comparing the entropic isomorphism kernel and the all subgraph kernel, we also observe three differences that conclude the advantage of the entropic isomorphism kernel. a) First, for the entropic isomorphism kernel we can efficiently measure the isomorphism for a pair of subgraphs of large size. The reason for this is that the computational complexity of the Shannon entropy associated with the steady state random walk is quadratic in the (sub)graph size. While for the all subgraph kernel, measuring the isomorphism between a pair of subgraphs usually requires burdensome computation. b) Second, the entropic isomorphism kernel overcomes the NP-hard problem of measuring all pairs of subgraphs that arise in the all subgraph kernel. c) Third, for the entropic isomorphism kernel, only the pair of expansion subgraphs having the same shortest paths of greatest lengths around their rooted vertices are evaluated for measuring isomorphism. In other words, only a pair of expansion subgraphs having the same layers around the rooted vertices can be evaluated. On the other hand, the all subgraph kernel roughly or arbitrarily evaluates a pair of subgraphs and counts the isomorphic subgraph pairs. Hence, the entropic isomorphism kernel also encapsulates location information between pairs of subgraphs, and this is ignored by the all subgraph kernel.

5. Experimental Results

In this section, we empirically evaluate the performance of our new subgraph kernels. Our experimental evaluation consists of two parts. First, we test our subgraph kernels on the graph classification problem using standard graph datasets. These graphs are abstracted from bioinformatics and computer vision databases. Moreover, we also compare our new subgraph kernels with several state-of-the-art methods. Second, we evaluate the computational efficiency of our new subgraph kernels.

5.1. Graph Datasets

We demonstrate the performance of our new subgraph kernels on six standard graph based datasets abstracted from problems formulated by bioinformatics and computer vision. These datasets include: MUTAG, D&D, ENZYMES, BAR31, BSPHERE31, GEOD31, CATH2, NCI1, NCI109, COIL5, Shock, PPIs, GATORBait and PTC(MR). More details concerning the datasets are shown in Table.1.

MUTAG: The MUTAG dataset consists of graphs representing 188 chemical compounds, and here the goal is to predict whether each compound possesses mutagenicity [33]. The maximum, minimum and average number of vertices are 28, 10 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.

D&D: The D&D dataset contains 1178 protein structures [34]. Each protein is represented by a graph, in which the vertices are amino acids and two vertices 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, minimum and average number of vertices are 5748, 30 and 284.32 respectively.

ENZYMES: The ENZYMES dataset consists of graphs representing protein tertiary structures, and contains 600 enzymes from the BRENDA enzyme database [37]. In this case, the task is to correctly assign each enzyme to one of the 6 EC top-level classes. The maximum, minimum and average number of vertices are 126, 2 and 32.63 respectively.

BAR31, BSPHERE31 and GEOD31: The SHREC 3D Shape database consists of 15 classes and 20 individuals per class, that is 300 shapes [29]. This is a standard benchmark in 3D shape recognition. From the SHREC 3D Shape database, we establish three graph datasets named BAR31, BSPHERE31 and GEOD31 datasets through three mapping functions. These functions are a) ERG barycenter: distance from the center of mass/barycenter, b) ERG bsphere: distance from the center of the sphere that circumscribes the object, and c) ERG integral geodesic: the average of the geodesic distances to all other points. Details of the three mapping function can be found in [29]. The number of maximum, minimum and average vertices for the three datasets are a) 220, 41 and 95.42 (for BAR31), b) 227, 43 and 99.83 (for BSPHERE31), and c) 380, 29 and 57.42 (for GEOD31), respectively.

CATH2: The CATH2 dataset has proteins in the same class (i.e., Mixed Alpha-Beta), architecture (i.e., Alpha-Beta Barrel), and topology (i.e., TIM Barrel), but in different homology classes (i.e., Aldolase vs. Glycosidases) [2]. The CATH2 dataset is harder to classify, since the proteins in the same topology class are structurally similar. The protein graphs are 10 times larger in size than chemical compounds, with $200 - 300$ vertices. There is 190 testing graphs in the CATH2 dataset.

NCI1 and NCI109: The NCI1 and NCI109 datasets consist of graphs representing two balanced subsets of datasets of chemical compounds screened for activity against non-small cell lung cancer and ovarian cancer cell lines respectively [35, 36]. There are 4110 and 4127 graphs in NCI1 and NCI109 respectively.

COIL5: We establish a COIL5 dataset from the COIL database. The COIL image database consists of images of 100 3D objects. We use the images for the first five objects. For each object we employ 72 images captured from different viewpoints. For each image we first extract corner points using the Harris detector, and then establish Delaunay graphs based on the corner points as vertices. As a result, in the dataset there are 5 classes of graphs, and each class has 72 testing graphs. The number of maximum, minimum and average vertices for the dataset are 241, 72 and 144.90 respectively.

Shock: The Shock dataset consists of graphs from the Shock 2D shape database. Each graph is a skeletal-based representation of the differential structure of the boundary of a 2D shape. There are 150 graphs divided into 10 classes. Each class contains 15 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 (i.e., a) *Aquifex*4 and *thermotoga*4 PPIs from *Aquifex aelicus* and *Thermotoga maritima*, b) *Gram-Positive*52 PPIs from *Staphylococcus aureus*, c) *Cyanobacteria*73 PPIs from *Anabaena variabilis*, d) *Proteobacteria*40 PPIs from *Acidovorax avenae*, and e) *Acidobacteria*46 PPIs). Note that, unlike the experiment in [38] that only uses the *Proteobacteria*40 and the *Acidobacteria*46 PPIs as the testing graphs, we use all the PPIs as the testing graphs in this paper. As a result, the experimental results for some kernels are different on the PPIs dataset.

GatorBait: GatorBait has 100 shapes representing fishes from 30 different classes [29]. We have extracted Delaunay graphs from their shape quantization (Canny algorithm followed by contour decimation). Since the classes are associated to fish genus and not to species, we find high intra-class variability in many cases. Therefore, the database, though having only 100 samples, plays a challenging role in testing graph classification. The number of maximum, minimum and average vertices for the dataset are 545, 239 and 348.70.

PTC: The PTC (The Predictive Toxicology Challenge) dataset records the carcinogenicity of several hundred chemical compounds for male rats (MR), female rats (FR), male mice (MM) and female mice (FM) [11]. These graphs are very small (i.e., $20 - 30$ vertices), and sparse (i.e., $25 - 40$ edges. We select the graphs of male rats (MR) for evaluation. There are 344 test graphs in the MR class.

5.2. Experiments on Graph Datasets

We evaluate the performance of our proposed Jensen-Shannon subgraph kernel (JSSK) and the entropic isomorphism kernel (ISK) on several standard graph datasets, and then compare them with several alternative state of the art graph kernels. The graph kernels used for comparison include: 1) the backtraceless random walk kernel using the Ihara zeta function based cycles (BRWK) [9], 2) the Weisfeiler-Lehman subtree kernel (WL) [5], 3) the shortest path graph kernel (SPGK)

Table 1: Information of the Graph-based Datasets

Datasets	MUTAG	D&D	ENZYMES	BAR31	BSPHERE31	GEOD31	CATH2
Max # vertices	28	5748	126	220	227	380	568
Min # vertices	10	30	2	41	43	29	143
Mean # vertices	17.93	284.3	32.63	95.42	99.83	57.42	308.03
# graphs	188	1178	600	300	300	300	190
# classes	2	2	6	15	15	15	2

Datasets	NCI1	NCI109	COIL5	Shock	PPIs	GatorBait	PTC
Max # vertices	111	111	241	33	218	545	109
Min # vertices	3	4	72	4	3	239	2
Mean # vertices	29.87	29.68	144.90	13.16	109.63	348.70	25.60
# graphs	4110	4127	360	150	219	100	344
# classes	2	2	5	10	5	30	2

[8], 4) the graphlet count graph kernels with graphlet of size 3 (GCGK) [30], 5) the unaligned quantum Jensen-Shannon kernel (UQJS) [38], and 6) the attributed graph kernel from the Jensen-Tsallis q-differences associated with $q = 2$ (JTQK) [39]. For our ISK kernel, we set the largest value of h as 10, i.e., at most 10 expansion subgraphs around a vertex are considered. For the WL kernel and JTQK kernel, we set the largest iteration of the required vertex label strengthening methods (i.e., the WL algorithm for the WL subtree kernel and the tree-index method for the JTQK kernel) as 10.

For each kernel, we compute the kernel matrix on each graph dataset. We perform 10-fold cross-validation using the C-Support Vector Machine (C-SVM) Classification, and compute the classification accuracies, using LIBSVM. We use nine samples for training and one for testing. The C-SVMs classification was performed with its parameters optimized on each dataset. We report the average classification accuracies and standard errors from the 10-fold cross-validation for each kernel in Table.2. Furthermore, we also report the runtime of computing the kernel matrices for each kernel in Table.3. Here, the runtime was measured under Matlab R2011a running on a 2.5GHz Intel 2-Core processor (i.e., i5-3210m). Finally, note that, the JTQK, WL and SPGK kernels are able to accommodate attributed graphs. In our experiments, we use the vertex degree (not the original vertex labels) as the vertex label for the JTQK, WL and SPGK kernels. Thus, the experimental results for these kernels on some datasets (i.e., the MUTAG, NCI1, NCI109,

Table 2: Classification Accuracy (In $\%$ $\pm$ Standard Error) Comparisons

Datasets	MUTAG	D&D	ENZYMES	BAR31	BSPHERE31	GEOD31	CATH2
JSSK	$83.77 \pm .74$	**76.32** $\pm .46$	$24.38 \pm .55$	$52.76 \pm .47$	$43.33 \pm .40$	32.03 ± 1.02	$75.42 \pm .76$
ISK	**84.66** $\pm .56$	$75.32 \pm .35$	**41.80** $\pm .43$	**62.80** $\pm .47$	**52.50** ± 47	$39.76 \pm .43$	$67.55 \pm .67$
JTQK	$83.22 \pm .87$	$74.35 \pm .23$	$39.38 \pm .76$	$60.56 \pm .35$	$46.93 \pm .61$	**40.10** $\pm .46$	$68.70 \pm .69$
UQJS	$82.72 \pm .44$	$--$	$36.58 \pm .46$	$30.80 \pm .61$	$24.80 \pm .61$	$23.73 \pm .66$	$71.11 \pm .88$
BRWK	$77.50 \pm .75$	$--$	$20.56 \pm .35$	$--$	$--$	$--$	$--$
WL	$82.05 \pm .57$	$73.52 \pm .20$	$38.41 \pm .45$	$58.53 \pm .53$	$42.10 \pm .68$	$38.20 \pm .68$	$67.36 \pm .63$
SPGK	$83.38 \pm .81$	$--$	$28.55 \pm .42$	$55.73 \pm .44$	$48.20 \pm .76$	$38.40 \pm .65$	**81.89** $\pm .63$
GCGK	$82.04 \pm .39$	$74.70 \pm .30$	$24.87 \pm .22$	$22.96 \pm .65$	$17.10 \pm .60$	$15.30 \pm .68$	73.68 ± 1.09

Datasets	NCI1	NI109	COIL5	Shock	PPIs	GatorBait	PTC
JSSK	$64.86 \pm .24$	$65.72 \pm .26$	$67.75 \pm .67$	$37.66 \pm .80$	$45.04 \pm .80$	$9.20 \pm .65$	$56.94 \pm .43$
ISK	$76.21 \pm .25$	$76.42 \pm .24$	$38.30 \pm .56$	$39.86 \pm .68$	$79.47 \pm .32$	**11.40** $\pm .52$	**60.26** $\pm .42$
JTQK	**81.23** $\pm .25$	**81.40** $\pm .26$	$30.86 \pm .66$	$37.73 \pm .72$	**88.47** $\pm .47$	$9.60 \pm .87$	$57.47 \pm .41$
UQJS	$69.09 \pm .20$	$70.17 \pm .23$	**70.11** $\pm .61$	**40.60** $\pm .92$	$65.61 \pm .77$	$9.00 \pm .89$	$56.70 \pm .49$
BRWK	$60.34 \pm .17$	$59.89 \pm .15$	$14.63 \pm .21$	$0.33 \pm .37$	$--$	$--$	$53.97 \pm .31$
WL	$80.68 \pm .27$	$80.72 \pm .29$	33.16 ± 1.01	36.40 ± 1.00	$88.09 \pm .41$	$10.10 \pm .61$	$56.85 \pm .52$
SPGK	$74.21 \pm .30$	$73.89 \pm .28$	$69.66 \pm .52$	$37.88 \pm .93$	$59.04 \pm .44$	$9.00 \pm .75$	$55.52 \pm .46$
GCGK	$63.72 \pm .12$	$62.33 \pm .13$	$66.41 \pm .63$	$26.93 \pm .63$	$46.61 \pm .47$	$8.40 \pm .83$	$55.41 \pm .59$

D&D, ENZYMES, PTC datasets that have original label information residing on vertices) may be different from those reported in [5, 8, 39].

Experimental Results: a) On the MUTAG dataset, the accuracy of our ISK kernel exceeds the alternative kernels. The accuracy of our JSSK kernel is competitive to that of the ISK kernel, but exceeds other kernels. b) On the D&D dataset, the accuracy of our JSSK kernel exceeds the alternative kernels. The accuracy of our ISK kernel is competitive to that of the JSSK and WL kernels, and exceeds other kernels. The SPGK and BRWK kernels cannot complete the required computations on the D&D dataset, because the graphs in the dataset are very large (e.g., some graphs have more than thousands vertices). c) On the ENZYMES, BAR31, BSPHERE31 and GEOD31 datasets, the accuracies of our ISK kernel obviously exceed those of the remaining kernels. The classification accuracies of our JSSK kernel are lower than those of the ISK, SPGK and WL kernels, and exceed other kernels. The BRWK cannot finish the computation on the BAR31, BSPHERE31 and GEOD31 datasets. This is because the BRWK kernel relies on the cycle structures of the graphs. The graphs in these datasets are very sparse. As a result, the

Table 3: CPU Runtime Comparisons

Datasets	MUTAG	D&D	ENZYMES	BAR31	BSPHERE31	GEOD31	CATH2
JSSK	1”	45”	1”	1”	1”	1”	4”
ISK	15”	$3h28$”	$3'30$”	$3'50$”	$3'10$”	$2'40$”	$9'51$”
JTQK	3”	$23h39'$	30”	$1'22$”	$1'35$”	$1'17$”	$39'14$”
UQJS	20”	$> 1day$	$4'23$”	$10'30$”	$13'48$”	$8'49$”	$1h14'$
BRWK	1”	$> 1day$	13”	$--$	$--$	$--$	$> 1day$
WL	3”	$7'43$”	21”	30”	25”	15”	53”
SPGK	1”	$> 1day$	2”	11”	14”	11”	$4'13$”
GCGK	1”	$1'17$”	2”	2”	2”	2”	8”

Datasets	NCI1	NCI109	COIL5	Shock	PPIs	GatorBait	PTC
JSSK	52”	53”	3”	1”	2”	3”	4”
ISK	$2h19'$	$2h20$”	$9'55$”	6”	$1'40$”	7”	59”
JTQK	$10'50$”	$10'55$”	$7'19$”	3”	$1'43$”	$29'31$”	8”
UQJS	$2h55'$	$2h55'$	$18'20$”	14”	$3'24$”	$20'53$”	$1'46$”
BRWK	$6'49$”	$6'49$”	$16'46$”	8”	$> 1day$	$> 1day$	29”
WL	$2'31$”	$2'37$”	$1'5$”	3”	20”	33”	9”
SPGK	16”	16”	31”	1”	22”	$2'25$”	1”
GCGK	5”	5”	4”	1”	4”	3”	1”

BRWK cannot capture any cycle in these datasets. d) On the CATH2 dataset, the accuracy of the SPGK kernel exceeds that of the remaining kernels. The accuracy of our JSSK kernel exceeds that of any kernel, excluding the SPGK kernel. Moreover, the accuracy of the ISK kernel exceeds that of all the remaining kernels with the exception of the JSSK, JTQK, UQJS and SPGK kernels. e) Overall, on the NCI1, NCI109, and PPIs datasets, the accuracies of the ISK kernel are only lower than those of the JTQK and WL kernels, but outperform the remaining kernels. On the other hand, the JSSK kernel only outperforms the GCGK and BRWK kernels. f) On the GatorBait and PTC datasets, the accuracies of our ISK kernel exceed those of all the alternative kernels. On the other hand, the accuracies of our JSSK kernel exceed or are competitive to those of all the alternative kernels. g) On the Shock dataset, the accuracy of our ISK kernel is only a little lower than that of the UQJS kernel, and exceeds that of all the remaining kernels. The accuracy of our JSSK kernel exceeds or is competitive to that of all alternative kernels. h) Finally, on the COIL5 dataset, the accuracy of our JSSK kernel exceeds or is competitive to that of all the alternative kernels, while the accuracy of our ISK kernel only exceeds that of the JTQK, WL and BRWK kernels, and is

lower than that of all the remaining kernels.

Discussion and Analysis: In terms of the runtime, it is clear that our JSSK kernel is the fastest kernel. It can efficiently finish the computation on all datasets. The reasons for this efficiency are twofold. First, for the JSSK kernel the required centroid depth-based representation (i.e., the centroid-based complexity trace) of a graph only encapsulates a small number of centroid expansion subgraphs. In other words, the JSSK kernel only measures limited number of subgraphs. Second, the associated Jensen-Shannon divergence measure between a pair of centroid expansion subgraphs only requires computation of quadratic vertex number, even a pair of large global graphs being compared. On the other hand, for our ISK kernel the efficiency is slower than that of the JSSK kernel. The reason for this is that the ISK kernel considers the depth-based representations (i.e., the h-layer depth-based representation around each vertex) rooted from all the vertices. By contrast, the JSSK kernel only considers the depth-based representation derived from the centroid vertex. As a result, the ISK kernel needs to measure more expansion subgraphs. Moreover, the efficiency of the ISK kernel is also slower than the JTQK, UQJS, SPGK, WL and GCGK kernels on some datasets, but it can still finish the computation in polynomial time. Unlike some kernels (i.e., the SPGK and BRWK kernels) which can only finish the computation on datasets having small graphs, our ISK kernel can also finish the computation on datasets having large graphs in polynomial time. The reason for this is that the required Shannon entropy associated with the steady state random walk for the ISK kernel can be efficiently computed (i.e., the computation is quadratic vertex number). As a result, the required h-layer depth-based representations and the entropy-based isomorphism test can be efficiently computed and measured, respectively.

In terms of the classification accuracies, our ISK kernel outperforms all the alternative kernels on the MUTAG, ENZYMES, BAR31, BSPHERE31, GatorBait and PTC datasets. On the other hand, on the PPIs, NCI1 and NCI109 datasets, the accuracies of our ISK kernel are only lower than those of the JTQK and WL kernels, but are higher than those of other kernels. On the Shock dataset, the accuracy of our ISK kernel is only a little lower than that of the UQJS kernel, but is higher than that of other kernels. On the D&D dataset, the accuracy of our ISK kernel is only a little lower than that of our JSSK kernel, but is higher than that of other kernels. The reason for this effectiveness are fourfold. First, comparing to the JSSK kernel our ISK considers the depth-based

representations around all the vertices, while our JSSK kernel only considers the depth-based representation around the centroid vertex. As a result, the ISK kernel can capture more depth-based features of a graph than the JSSK kernel. Second, comparing to the JTQK and WL kernels, our ISK kernel overcomes the notorious tottering problem that arises in the JTQK and WL kernels that require vertex label strengthening methods. This is because a subtree identified by the Weisfeiler-Lehman algorithm for the WL kernel and the tree-index method for JTQK kernels may encapsulate several same pairs of vertices connected by the same edges. By contrast, any expansion subgraph used for the ISK kernel does not contain any repeated topology structure. Third, comparing to the GCGK, SPGK and BRWK kernels our ISK kernel overcomes the simple substructure problem that arises in the GCGK, SPGK and BRWK kernels. The sizes of the substructures used in the GCGK, SPGK and BRWK kernels (i.e., the graphlet structures, shortest paths and cycles) are very small (i.e., these substructures are structurally simple) and only reflect restrict topology information of graphs. By contrast, the h-layer depth-based representation used in the ISK kernel tends to lead a vertex to the global graph. As a result, the ISK kernel can reflect richer topology information of graphs. Fourth, compared with the UQJS kernel, which only reflects global similarity information between a pair of graphs, our ISK kernel can reflect richer interior topological information relying on the h-layer depth-based representation. Overall, the performance of the JTQK and WL kernels is competitive to that our ISK kernel. However, we also observe that all the three kernels do not perform well on the COIL and CATH2 datasets. We observe that the vertex degrees in the graphs used for testing are quite similar, when compared to the degree distributions for graphs from the alternative datasets. This indicates that our ISK kernel, together with the JTQK and WL kernels are not suitable for graphs having similar vertex degrees.

Comparisons with the Jensen-Shannon Diffusion Kernel: To take our study one step further, we evaluate the performance of the Jensen-Shannon diffusion kernel (JSDK), that are integrated in our JSSK kernel, on the graph datasets. The classification accuracies (including the standard errors) and the CPU runtime are reported in Table.4 and Table.5 respectively. Through Table.5, we observe that the runtime of the diffusion kernel is more efficient than that of the proposed subgraph kernels, since it just compares a pair of graphs without establishing (centroid) expansion subgraphs or comparing pairs of the subgraphs. On the challenge D&D dataset containing large graphs, the

Table 4: Classification Accuracies (In $\%$ $\pm$ Standard Error) for JSDK

Datasets	MUTAG	D&D	ENZYMES	BAR31	BSPHERE31	GEOD31	CATH2
JSDK	$83.11 \pm .80$	$75.13 \pm .31$	$20.81 \pm .29$	$22.10 \pm .37$	$19.00 \pm .33$	$16.53 \pm .34$	$72.26 \pm .76$
Datasets	NCI1	NCI109	COIL5	Shock	PPIs	GatorBait	PTC
JSDK	$62.50 \pm .33$	$63.00 \pm .35$	$69.13 \pm .79$	$21.73 \pm .76$	$34.57 \pm .59$	$7.8 \pm .70$	$57.29 \pm .41$

Table 5: CPU Runtime for JSDK

Datasets	MUTAG	D&D	ENZYMES	BAR31	BSPHERE31	GEOD31	CATH2
JSDK	1”	1”	1”	1”	1”	1”	1”
Datasets	NCI1	NCI109	COIL5	Shock	PPIs	GatorBait	PTC
JSDK	1”	1”	1”	1”	1”	1”	1”

runtime of the diffusion kernel is only 1”. Unfortunately, the classification accuracies on all the datasets tend to be lower than those of the JSSK and ISK kernels. The reason for this is that the Jensen-Shannon diffusion kernel only measures the kernel between a pair of graphs. Compared to the proposed JSSK kernel, the diffusion kernel is just a similarity measure of their L-layer centroid expansion subgraphs, and is a restricted version of depth-based characterizations of a graph. As a result, the diffusion kernel only captures limited topological arrangement information for graphs.

Statistical Analysis: Table 2 indicates that our ISK and JSSK kernels outperform or are competitive to the alternative kernels. Moreover, we observe that there is no kernel that performs best on any dataset. To establish which kernel is the best one over all different datasets, for each kernel we also compute the average classification accuracy associated with its standard errors from the accuracies for all the datasets. The results are shown in Table 6. Note that, some kernels cannot complete the kernel matrix computation on some of the datasets. For these kernels, we perform the statistical analysis on those datasets on which the computation can be completed. In terms of the average classification accuracies, it is clear that our ISK kernel outperforms each of the alternative kernel. Only the JTQK and WL kernels are competitive to our ISK kernel. Moreover, the standard error for our ISK kernel is also lower than that of the JTQK and WL kernels. This indicates that the performance of our ISK kernel is more stable than that of the JTQK and WL kernels over all the datasets. On the other hand, the standard error of our ISK kernel is lower than most alternative kernels excluding the GCGK and JSDK kernels, i.e., the performance stability of our ISK kernel

Table 6: Statistical Analysis of All Kernels over All Datasets

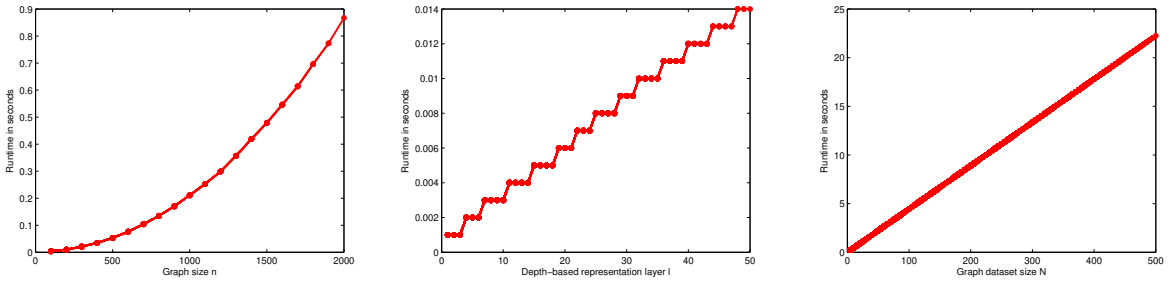
Datasets	JSSK	ISK	JTQK	UQJS	BRWK	WL	SPGK	GCGK	JSDK
Average Accuracy	52.51	**57.59**	57.14	50.07	41.03	56.15	55.02	45.74	44.64
Standard Error	±.82	±.73	±.85	±.90	±1.54	±.82	±.1.22	±.71	±**.57**

is lower than that of the GCGK and JSDK kernels. However, the classification accuracy of our ISK kernel is obviously higher than that of the GCGK and JSDK kernels. Moreover, through our ISK kernel is not the fast kernel, the runtime of our ISK kernel is still reasonable and applicable. As a summary of the statistical analysis, our ISK kernel is the best kernel in terms of either the classification accuracy and the stability of performance.

5.3. Computational Evaluation

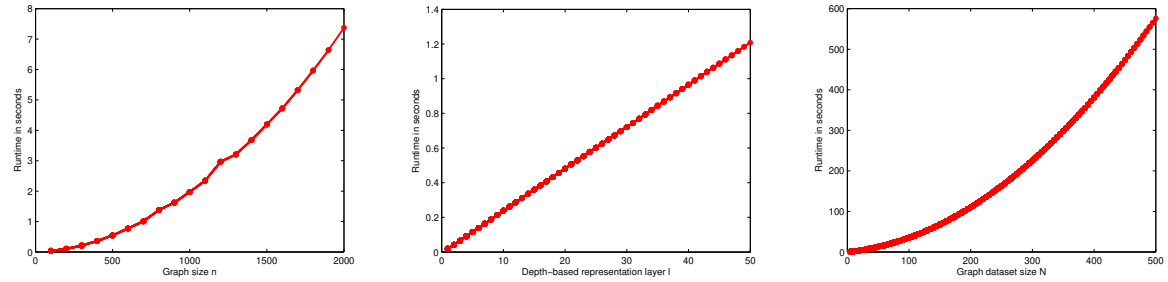
Finally, we evaluate the computational efficiency (i.e., the CPU runtime) of our new depth-based subgraph kernels, and explore the relationship between the computational overheads and the structural complexity or number of the associated graphs.

Experimental setup: For both of our JSSK and ISK kernels, we evaluate the computational efficiency on randomly generated graphs with respect three parameters: a) the graph size n, b) the largest layer l of the centroid-based complexity traces (for the JSSK kernel), and c) the h-layer ($h = l$) depth-based representations (for the ISK kernel) of graphs, and the graph dataset size N. We vary n over the set of values $\{100, 200, \ldots, 2000\}$, l over the set of values $\{1, 2, \ldots, 50\}$ and N over the set of values $\{5, 10, \ldots, 500\}$, separately. a) For the experiments with graph size n, we generate 20 pairs of graphs with increasing number of vertices. We report the runtime for computing the kernel values between pairwise graphs (for the ISK kernel, $h = 10$). b) For the experiments with the largest layer l, we generate a pair of graphs each of which has 200 vertices. We report the runtime for computing the kernel values of the pair of graphs as a function of l. c) For the graph dataset sized N, we generate 500 graph datasets with an increasing number of test graphs. In each dataset, one graph has 200 vertices. We report the runtime for computing the kernel matrices for each graph dataset (for the ISK kernel, $h = 10$). The CPU runtime is reported in Fig.2 and Fig.3 respectively, as operated in Matlab R2011b on a 2.5GHz Intel 2-Core processor (i.e., i5-3210m).



(a) For Graph Size n (b) For Layer l (c) For Data Size N

Figure 2: Runtime Evaluations for JSSK Kernel.



(a) For Graph Size n (b) For Layer l (c) For Data Size N

Figure 3: Runtime Evaluations for ISK Kernel.

Experimental results: Figs.2 (a), (b) and (c) show the results for the JSSK kernel when varying the parameters n, l and N, respectively. Figs.3 (a), (b) and (c) show the results for the JSSK kernel when varying the parameters n, l and N, respectively.

From Figs.2, we observe that the runtime of the JSSK kernel scales quadratically with n, and linearly with l and N. From Figs.3, we observe that the runtime of the ISK kernel scales quadratically with n, linearly with l, and quadratically with N. These results verify that our JSSK and ISK subgraph kernels can be computed in polynomial time. The JSSK kernel is much more efficient than the ISK kernel.

6. Conclusion and Further Work

In this paper, we have shown how to construct fast subgraph kernels using depth-based representations of graphs. We have proposed two new depth-based subgraph kernels, namely a) the Jensen-Shannon subgraph kernel and b) the entropic isomorphism kernel. The Jensen-Shannon subgraph kernel is based on a fast Jensen-Shannon diffusion kernel measure defined in terms of the Jensen-Shannon divergence on (sub)graphs and a graph decomposition through a centroid-based representation. On the other hand, the entropic isomorphism kernel is based on an entropy-based isomorphism test between the subgraphs of pairwise h-layer depth-based representations. Both of the new depth-based subgraph kernels can be computed in polynomial time. In particular, the Jensen-Shannon subgraph kernel overcomes the subgraph size restrictions arising in state-of-the-art graph kernels, and also renders an efficient computation. The experimental results have demonstrated both the effectiveness and efficiency of the new depth-based subgraph kernels.

Our future work is to extend the depth-based subgraph kernels to attributed graphs. Moreover, we would also like to extend the Jensen-Shannon subgraph kernel from classical random walks to quantum walks, and develop a quantum subgraph kernel. While in this paper, we have applied the classical Jensen-Shannon divergence to classical walks to compute the Jensen-Shannon diffusion kernel between (sub)graphs, in recent work Emms et al. [31] and Ren et al. [32] have explored both continuous-time and discrete-time quantum walks on graphs. It would be interesting to extend this work, using the quantum Jensen-Shannon divergence [16] to compare quantum walks between (sub)graphs.

Acknowledgments

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

We thank Prof. Karsten Borgwardt and Dr. Nino Shervashidze for providing the Matlab implementation for the various graph kernel methods, and Dr. Geng Li for providing the graph datasets.

References

[1] F. Escolano, E.R. Hancock, M.A. Lozano, Heat diffusion: Thermodynamic depth complexity of networks, Physical Review E 85 (2012) 036206.

[2] L. Bai, Edwin R. Hancock, Depth-based complexity traces of graphs, Pattern Recognition 47 (2013) 1772-1186.

[3] T. Gärtner, P.A. Flach, S. Wrobel, On graph kernels: Hardness results and efficient alternatives, in: Proceedings of Conference on Learning Theory, 2003, pp. 129-143.

[4] L. Bai, E.R. Hancock, Graph kernels from the Jensen-Shannon divergence, Journal of Mathematical Imaging and Vision 47 (2013) 60-69.

[5] N. Shervashidze, P. Schweitzer, E.J. Leeuwen, K, Mehlhorn, K.M. Borgwardt, Weisfeiler-Lehman graph kernels, Journal of Machine Learning Research 1 (2010) 1-48.

[6] G. Li, M. Semerci, B. Yener, M.J. Zaki, Effective graph classification based on topological and label attributes, Statistical Analysis and Data Mining 5 (2012) 265-283.

[7] H. Kashima, K. Tsuda, A. Inokuchi, Marginalized kernels between labeled graphs, in: Proceedings of International Conference on Machine Learing, 2003, pp. 321-328.

[8] K.M. Borgwardt, H. Kriegel, Shortest-Path kernels on graphs, in: Proceedings of the IEEE International Conference on Data Mining, 2005, pp. 74-81.

[9] F. Aziz, R.C. Wilson, E.R. Hancock, Backtrackless walks on a graph, IEEE Trans. Neural Netw. Learning Syst. 24 (2013) 977-989.

[10] P. Ren, R.C. Wilson, E.R. Hancock, Graph Characterization via Ihara Coefficients, IEEE Transactions on Neural Networks 22 (2011) 233-245.

[11] F. Costa, K.D. Grave, Fast neighborhood subgraph pairwise distance kernel, in Proceedings of International Conference on Machine Learing, 2010, 255-262.

[12] Z. Harchaoui, F. Bach, Image classification with segmentation graph kernels, in Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[13] F.R. Bach, Graph kernels between point clouds, in: Proceedings of International Conference on Machine Learing, 2008, 25-32.

[14] N. Kriege, P. Mutzel, Subgraph matching kernels for attributed graphs, in: Proceedings of International Conference on Machine Learing, 2012.

[15] L. Bai, P. Ren, Edwin R. Hancock, A hypergraph kernel from the isomorphism tests, to appear in: Proceedings of International Conference on Pattern Recognition, 2014.

[16] P.W. Lamberti, A.P. Majtey, A. Borras, M. Casas, A. Plastino, Metric character of the quantum Jensen-Shannon divergence, Physical Review A 77 (2008) 052311.

[17] J.P. Crutchfield, C.R. Shalizi, Thermodynamic depth of causal states: Objective complexity via minimal representations, Physical Review E 59 (1999) 275283.

[18] L. Bai, E.R. Hancock. L. Han, P. Ren, Graph clustering using graph entropy complexity traces, in: Proceedings of International Conference on Pattern Recognition, 2012, pp. 2881-2884.

[19] L. Bai, E.R. Hancock, Graph complexity from the Jensen-Shannon divergence, in: Proceedings of SSPR/SPR, 2012, pp. 79-98.

[20] L. Bai, E.R. Hancock, Graph clustering using the Jensen-Shannon kernel, in: Proceedings of International Conference on Computer Analysis of Images and Patterns, 2011, pp. 394-401.

[21] M. Dehmer, Information-theoretic concepts for the analysis of complex networks, Applied Artificial Intelligence 22 (2008) 684-706.

[22] K. Anand, G. Bianconi, S. Severini, Shannon and von Neumann entropy of random networks with heterogeneous expected degree, Physical Review E 83 (2011), 036169.

[23] F. Passerini, S. Severini, Quantifying complexity in networks: the von Neumann entropy, International Journal of Agent Technologies and Systems 1 (2009), 58-67.

[24] T.H. Cormen, C.E. Leiserson, R.L. Rivest (Eds), Introduction to Algorithms, MIT Press and McGraw-Hill, 2001.

[25] M. Gadouleau, S. Riis, Graph-theoretical constructions for graph entropy and network coding based communications, IEEE Transactions on Information Theory 57 (2011), 6703-6717.

[26] J. Köner, Coding of an information source having ambiguous alphabet and the entropy of graphs, in: Proceedings of the 6th Prague Conference on Information Theory, Statistical Decision Function, Random Processes, 1971, pp. 411-425.

[27] K. Riesen, H. Bunke, Graph Classification and Clustering based on Vector Space Embedding, World Scientific Publishing Co., Inc., 2010.

[28] M. Neuhaus, H. Bunke, Bridging the Gap between Graph Edit Distance and Kernel Machines, World Scientific, 2007.

[29] S. Biasotti, S. Marini, M. Mortara, G. Patanè, M. Spagnuolo, B. Falcidieno, 3D shape matching through topological structures, in: Proceedings of DGCI, 2003, 194-203.

[30] N. Shervashidze, S. V. N. Vishwanathan, T. Petri, K. Mehlhorn, K.M. Borgwardt, Efficient graphlet kernels for large graph comparison, Journal of Machine Learning Research 5 (2009) 488-495.

[31] D. Emms, R.C. Wilson, E.R. Hancock, Graph matching using the interference of continuous-time quantum walks, Pattern Recognition 42 (2009) 985-1002.

[32] P. Ren, T. Aleksic, D. Emms, R.C. Wilson, E.R. Hancock, Quantum walks, Ihara zeta functions and cospectrality in regular graphs, Quantum Information Processing 10 (2011), 405-417.

[33] A.K. Debnath, R.L. Lopez de Compadre, G. Debnath, A.J. Shusterman, C. Hansch, Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds, Correlation with molecular orbital energies and hydrophobicity, J. Med. Chem. 34 (1991) 786-797.

[34] K.M. Borgwardt, C.S. Ong, S. Schoenauer, S.V.N. Vishwanathan, A.J. Smola, H.P. Kriegel, Protein function prediction via graph kernels, Bioinformatics 21 (2005) (Suppl 1)47-56.

[35] P.D. Dobson, A.J. Doig, Distinguishing enzyme structures from non-enzymes without alignments, J. Mol. Biol. 330 (2003) 771-783.

[36] N. Wale, G. Karypis, Comparison of descriptor spaces for chemical compound retrieval and classification, in: Proceedings of IEEE International Conference on Data Mining, 2006, pp. 678-689.

[37] I. Schomburg, A. Chang, C. Ebeling, M. Gremse, C. Heldt, G. Huhn, D. Schomburg, The enzyme database: Updates and major new developments, Nucleic Acids Research 32 (2004) 431-433.

[38] L. Bai, L. Rossi, A. Torsello, E.R. Hancock, A quantum Jensen-Shannon graph kernel for unattributed graphs, Pattern Recognition 48 (2015) 344-355.

[39] L. Bai, L. Rossi, H. Bunke, E.R. Hancock, Attributed graph kernels using the Jensen-Tsallis q-differences, in: Proceedings of Eurepean Conference on Machine Learning and Knowledge Discovery in Databases(ECML-PKDD), 2014, Part I, pp. 15-19.

***Author Biography**

Lu Bai received the Ph.D. degree from the University of York, York, UK, and both the B.Sc. and M.Sc degrees from Faculty of Information Technology, Macau University of Science and Technology, Macau SAR, P.R. China. He is now a Asistant Professor in School of Information, Central University of Finance and Economics, Beijing, China. His current research interests include structural pattern recognition, machine learning, quantum walks on networks, and graph matching, especially in kernel methods and complexity analysis on (hyper)graphs and networks.

Edwin R. Hancock received the B.Sc., Ph.D., and D.Sc. degrees from the University of Durham, Durham, U.K. He is now a Professor of computer vision in the Department of Computer Science, University of York, York, U.K. He has published nearly 150 journal articles and 550 conference papers. Prof. Hancock was the recipient of a Royal Society Wolfson Research Merit Award in 2009. He has been a member of the editorial board of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, PATTERN RECOGNITION, COMPUTER VISION AND IMAGE UNDERSTANDING, and IMAGE AND VISION COMPUTING. His awards include the Pattern Recognition Society Medal in 1991, outstanding paper awards from the Pattern Recognition Journal in 1997, and the best conference best paper awards from the Computer Analysis of Images and Patterns Conference in 2001, the Asian Conference on Computer Vision in 2002, the International Conference on Pattern Recognition (ICPR) in 2006, British Machine Vision Conference (BMVC) in 2007, and the International Conference on Image Analysis and Processing in 2009. He is a Fellow of the International Association for Pattern Recognition, the Institute of Physics, the Institute of Engineering and Technology, and the British Computer Society. He was appointed as the founding Editor-in-Chief of the Institute of Engineering & Technology Computer Vision Journal in 2006. He was a General Chair for BMVC in 1994 and the Statistical, Syntactical and Structural Pattern Recognition in 2010, Track Chair for ICPR in 2004, and Area Chair at the European Conference on Computer Vision in 2006 and the Computer Vision and Pattern Recognition in 2008. He established the energy minimization methods in the Computer Vision and Pattern Recognition Workshop Series in 1997.