© 2016, Elsevier. Licensed under the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International
http://creativecommons.org/licenses/by-nc-nd/4.0/

Pattern Recognition Letters
journal homepage: www.elsevier.com

Quantum Kernels for Unattributed Graphs using Discrete-time Quantum Walks

Lu Bai[a,], Luca Rossi[b,], Lixin Cui[a,], Zhihong Zhang[c,], Peng Ren[d,], Xiao Bai[e,], Edwin Hancock[f,]

[a] *School of Information, Central University of Finance and Economics, Beijing, China*
[b] *School of Engineering and Applied Science, Aston University, Birmingham, UK*
[c] *Software school, Xiamen University, Fujian, China*
[d] *College of Information and Control Engineering, China University of Petroleum (Huadong), Shandong Province, China*
[e] *School of Computer Science and Engineering, Beihang University, Beijing, China*
[f] *Department of Computer Science, University of York, York, UK*

ABSTRACT

In this paper, we develop a new family of graph kernels where the graph structure is probed by means of a discrete-time quantum walk. Given a pair of graphs, we let a quantum walk evolve on each graph and compute a density matrix with each walk. With the density matrices for the pair of graphs to hand, the kernel between the graphs is defined as the negative exponential of the quantum Jensen-Shannon divergence between their density matrices. In order to cope with large graph structures, we propose to construct a sparser version of the original graphs using the simplification method introduced in Qiu and Hancock (2007). To this end, we compute the minimum spanning tree over the commute time matrix of a graph. This spanning tree representation minimizes the number of edges of the original graph while preserving most of its structural information. The kernel between two graphs is then computed on their respective minimum spanning trees. We evaluate the performance of the proposed kernels on several standard graph datasets and we demonstrate their effectiveness and efficiency.

Keywords: Graph Kernels, Discrete-time Quantum Walks, Quantum Jensen-Shannon Divergence

1. Introduction

Graph structures are important tools for representing structural data, since they can naturally reflect the structural and relational arrangements of objects Vento (2015); Foggia et al. (2014). One challenge arising in classifying graph-based data is that of converting discrete graph structures into numeric features. One way is to use graph kernels.

1.1. Graph Kernels

Graph kernels have been proven powerful for structural analysis in pattern recognition and machine learning Gärtner (2003). Typical applications include a) computer vision data classification Harchaoui and Bach (2007); Barra and Biasotti (2014), b) handwriting recognition Gärtner et al. (2003), c) bioinformatics data classification Bai et al. (2015b); Borgwardt et al. (2005); Emmert-Streib and Dehmer (2011), d) protein recognition Borgwardt et al. (2005), and e) 3D shape classification Bai et al. (2015c). The main advantages of using graph kernels are twofold. First, graph kernels can characterize graph features in a high dimensional space and thus have the capability of preserving the original structural information Brun et al. (2010). Second, graph kernels provide a way of making the rapidly developing kernel machinery for vectorial data applicable to graphs.

Generally speaking, a graph kernel is a similarity measure between a pair of graphs Schölkopf and Smola (2002); Vishwanathan et al. (2010). To extend the large spectrum of kernel methods from the general machine learning domain to the graph domain, Haussler Haussler (1999) has proposed a prin-

E-mail: bailu69@hotmail.com; bailucs@cufe.edu.cn
E-mail: l.rossi@aston.ac.uk
E-mail: cuilixin@cufe.edu.cn
E-mail: zhihong@xmu.edu.cn (Corresponding author)
E-mail: pengren@upc.edu.cn
E-mail: baixiao@huaa.edu.cn
E-mail: erh@cs.york.ac.uk

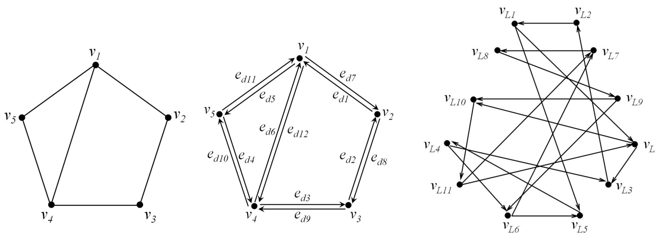
cipled way, namely the R-convolution, to define a graph kernel. For a pair of graphs, an R-convolution kernel is computed by decomposing each graph into smaller subgraphs and counting the number of isomorphic subgraph pairs between the two original graphs. Thus, a new type of decomposition of a graph usually results in a new graph kernel. Kashima et al. Kashima et al. (2003) have introduced the random walk kernel, which is based on the enumeration of common random walks between two graphs. Borgwardt et al. Borgwardt and Kriegel (2005), on the other hand, have proposed a shortest path kernel by counting the numbers of matching shortest paths over the graphs. Aziz et al. Aziz et al. (2013) have developed a backtrackless kernel using the cycles identified by the Ihara zeta function Ren et al. (2011b) in a pair of graphs. Shervashidze et al. Shervashidze et al. (2010) have developed a fast subtree kernel by comparing pairs of subtrees identified by the Weisfeiler-Lehman (WL) algorithm. Gaidon et al. Gaidon et al. (2011) have developed a subtree kernel for comparing videos. For each video, the method considers complex actions as decomposed spatio-temporal parts and builds corresponding binary trees. The resulting kernel is computed by counting the number of isomorphic subtree patterns. Wang and Sahbi Wang and Sahbi (2013) have defined a graph kernel for action recognition. They first describe actions in the videos using directed acyclic graphs (DAGs). The resulting kernel is defined as an extension of the random walk kernel by counting the number of isomorphic walks of DAGs. Some other alternative R-convolution kernels include a) the segmentation graph kernel developed by Harchaoui and Bach Harchaoui and Bach (2007), b) the point cloud kernel developed by Bach Bach (2008), c) the (hyper)graph kernel based on directed subtree isomorphism tests Bai et al. (2014a), and d) the depth-subgraph kernels Bai and Hancock (2016).

Recently, a number of alternative graph kernel measures have been introduced in the literature. These are based on the computation of the mutual information between two graphs in terms of the classical Jensen-Shannon divergence. In information theory, the classical Jensen-Shannon divergence is a dissimilarity measure between probability distributions. In Bai and Hancock (2013), Bai et al. have used the classical Jensen-Shannon divergence to define a Jensen-Shannon graph kernel. Unlike the R-convolution kernels that count the number of isomorphic substructure pairs, the Jensen-Shannon graph kernel is defined in terms of the entropy difference between a pair of graphs and their composite graph, e.g., the disjoint union graph or the product graph formed by the pair of graphs. Here, the entropy of a graph can be either the von Neumann entropy (associated with the graph spectrum information) or the Shannon entropy (associated with the steady state random walk or the information functional). Both the von Neumann entropy and the Shannon entropy of a graph can be directly computed without the need to decompose the graph into substructures. As a result, the Jensen-Shannon graph kernel avoids the computational burden of comparing all pairs of substructures for a pair of graphs. To develop this work further, in Bai et al. (2015a); Rossi et al. (2013b) Bai et al. have introduced a new quantum Jensen-Shannon graph kernel using the quantum Jensen-Shannon divergence Lamberti et al. (2008); Majtey et al. (2005) and continuous-time quantum walks Farhi and Gutmann (1998). Here the basic idea is to associate with each graph a mixed quantum state representing the time evolution of a quantum walk. The kernel between a pair of graphs is then defined as the quantum Jensen-Shannon divergence between their corresponding density matrices Bai et al. (2015a). Rossi et al. Rossi et al. (2013b,a, 2015) have also proposed to use continuous-time quantum walks to measure the similarity between a pair of graphs. However, their approach is fundamentally different from that of Bai et al. as it requires merging the input graphs in a superstructure over which the quantum walks take place. The resulting graph kernel is also based on the quantum Jensen-Shannon divergence, but so far it has not been proved to be positive definite.

1.2. Contribution

The aim of this paper is to develop the kernels of Bai et al. Bai et al. (2015a); Rossi et al. (2013b) one step further. In particular, in this paper we propose to probe the graph structure using discrete-time quantum walks. The discrete-time quantum walk is the quantum analogue of the discrete-time classical random walk Farhi and Gutmann (1998). Remarkably, the discrete-time quantum walk possesses a number of interesting properties that are not exhibited by its classical counterpart. In fact, the behaviour of the discrete-time quantum walk is governed by a unitary matrix rather than a stochastic matrix, as in the case of the classical random walk. As a consequence, its evolution is reversible and non-ergodic. However, unlike the continuous-time quantum walk, where the state space is the graph vertex set, the state space of the discrete-time quantum walk is the set of arcs residing on the graph edges. More specifically, given an undirected graph $G(V, E)$, each edge $\{u, v\} \in E$ is replaced by a pair of directed arcs (u, v) and (v, u), and the set of arcs is denoted by E_d. Then, the state space for the discrete-time quantum walk is the set of arcs E_d. Since the number of directed arcs is much larger than, or at least equal to, that of the vertices, the discrete-time quantum walk can capture the structural characteristics of the graph better than the continuous-time quantum walk. However, this clearly comes at the cost of higher computational complexity. In this paper, we are interested in developing a new family of kernels where the graph structure is probed by means of discrete-time quantum walks.

To this end, we simulate the evolution of a discrete-time quantum walk on each graph and we compute an associated density matrix. Given a pair of graphs and their density matrices, the kernel between them is defined as the negative exponential of the quantum Jensen-Shannon divergence between their respective density matrices. Unfortunately, we show that this kernel does not scale to large graphs. Therefore, we propose to compute a sparser version of the original graphs in order to reduce the computational complexity of the kernel. More specifically, we use the graph simplification method introduced in Qiu and Hancock (2007). Given an undirected graph, we compute the minimum spanning tree over its commute time matrix representation. This is shown to minimize the number of edges of the original graph while preserving most of its structural information Qiu and Hancock (2007). In particular, the commute



(a) Original graph. (b) Symmetric digraph. (c) Directed line graph.

Fig. 1. Directed line graph construction.

time is robust with respect to the structural noise, and therefore it represents an ideal candidate to sparsify the structure of the original graph Qiu and Hancock (2007). Such a method allows the kernel between two graphs to be computed on their respective minimum spanning trees. We show that this strategy significantly reduces the computational complexity of the original kernel making it feasible to larger graphs.

We evaluate the performance of both kernels on several standard graph datasets from both bioinformatics and computer vision. The experimental results demonstrate the effectiveness of the proposed quantum Jensen-Shannon graph kernels. Both kernels are shown to be competitive with respect to state-of-the-art graph kernels.

The remainder of this paper is organized as follows: Section 2 introduces the necessary quantum mechanical background, while Section 3 reviews the graph simplification techniques used in this paper. In Section 4 we define the proposed graph kernels. Section 5 provides experimental evaluations and Section 6 concludes the paper.

2. Quantum Mechanical Background

In this section, we introduce the quantum mechanical formalism that will be used in this work. We begin by reviewing the concept of discrete-time quantum walk on a graph. We show how to associate with each graph a density matrix describing the quantum walk evolution, and how to compute the quantum Jensen-Shannon divergence between a pair of density matrices. Finally, we discuss the relationship between the Perron-Frobenius operator Ren et al. (2011a) and the transition matrix of the discrete-time quantum walk, and thus we explain the advantage of discrete-time quantum walks over their continuous-time version.

2.1. Discrete-time Quantum Walks

The discrete-time quantum walk represents the quantum counterpart of the discrete-time classical random walk Emms et al. (2009). Quantum processes are reversible, so in quantum walks the states needs to specify both the current and the previous location of the walk. Let us replace each edge $e(u, v) \in E$ with a pair of directed arcs $e_d(u, v)$ and $e_d(v, u)$. We denote the new set of arcs as E_d. Then, the state space for the discrete-time quantum walk is E_d and we denote the state corresponding to the walker being on the arc $e_d(u, v)$ as $|uv\rangle$. That is, $|uv\rangle$ denotes the state in which the walk is at vertex v having previously been at vertex u. A general state of the walk is

$$|\psi\rangle = \sum_{e_d(u,v)\in E_d} \alpha_{uv}|uv\rangle, \tag{1}$$

where the quantum amplitudes α_{uv} are complex, i.e., $\alpha_{uv} \in \mathbb{C}$. The probability that the walk is in the state $|uv\rangle$ is given by $\Pr(|uv\rangle) = \alpha_{uv}\alpha^*_{uv}$, where α^*_{uv} is the complex conjugate of α_{uv}.

At each time step, the evolution of the walk is governed by the transition matrix $\boldsymbol{U}$. The entries of $\boldsymbol{U}$ determine the transition probabilities between states, i.e., $|\psi_{t+1}\rangle = \boldsymbol{U}|\psi_t\rangle$. Since the evolution of the walk is linear and conserves probability, the matrix $\boldsymbol{U}$ must be unitary, i.e., $\boldsymbol{U}^{-1} = \boldsymbol{U}^\dagger$, where $\boldsymbol{U}^\dagger$ denotes the Hermitian transpose of $\boldsymbol{U}$.

It is usual to adopt the Grover diffusion matrix L (1996) as the transition matrix. Using the Grover diffusion matrix, the transition matrix $\boldsymbol{U}$ has entries

$$U_{(u,v),(w,x)} = \begin{cases} \frac{2}{d_x} - \delta_{ux}, & v = w; \\ 0, & \text{otherwise}, \end{cases} \tag{2}$$

where d_x is the vertex degree for vertex x, $U_{(u,v),(w,x)}$ gives the quantum amplitude for the transition $e_d(u, v) \rightarrow e_d(w, x)$ and δ_{ux} is the Kronecker delta, i.e., $\delta_{ux} = 1$ if $u = x$ and 0 otherwise. Given a state $|u_1 v\rangle$, the Grover matrix assigns the same amplitude to all transitions $|u_1 v\rangle \rightarrow |vu_i\rangle$, and a different amplitude to the transition $|u_1 v\rangle \rightarrow |vu_1\rangle$, where u_i is a neighbour of v. Finally, note that although the entries of $\boldsymbol{U}$ are real, they can be negative as well as positive. It is important to stress that, as a consequence of this, negative quantum amplitudes can arise during the evolution of the walk. In other words, the definition in Eq.(2) allows *destructive interference* to take place.

2.2. Relation to the Perron-Frobenius Operator

In Ren et al. (2011a), Ren et al. have demonstrated that the Perron-Frobenius operator can be represented in terms of the transition matrix of discrete-time quantum walks. To show this connection, we first introduce the definitions of directed line graph and positive support of a matrix.

Definition 1 For a graph $G(V, E)$, the directed line graph $OLG(V_L, E_{dL})$ is a dual representation of $G(V, E)$. To obtain $OLG(V_L, E_{dL})$, we first construct the associated symmetric digraph $SDG(V, E_d)$, where we replace every edge $e(u, w) \in E(G)$ by a pair of arcs, i.e., directed edges $e_d(u, w) \in E_d(G)$ and $e_d(w, u) \in E_d(G)$ for $u, w \in V$. The directed line graph $OLG(V_L, E_{dL})$ is the directed graph with vertex set V_L and edge set E_{dL} defined as follows

$$\begin{aligned} V_L &= E_d(SDG), \\ E_{dL} &= \{(e_d(u, v), e_d(v, w)) \in E_d(SDG) \times E_d(SDG) \\ &\quad | \; u, v, w \in V, u \neq w\}. \end{aligned}$$

The Perron-Frobenius operator $\boldsymbol{T} = [T_{i,j}]_{|V_L|\times|V_L|}$ of $G(V, E)$ is the adjacency matrix of the associated directed line graph $OLG(V_L, E_{dL})$. □

An example of transforming an original graph into a directed line graph is shown in Figure 1. Figure 1(a) shows the original graph and Figure 1(b) shows the associated symmetric digraph.

Definition 2 The positive support $\mathrm{S}^{+}(\boldsymbol{M}) = [s_{i,j}]_{m\times n}$ of the matrix $\boldsymbol{M} = [M_{i,j}]_{m\times n}$ is defined to be a matrix with entries

$$s_{i,j} = \begin{cases} 1, & M_{i,j} > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where $1 \leq i \leq m, 1 \leq j \leq n$. □

Based on the definition in Ren et al. (2011a), we can redefine the Perron-Frobenius operator in terms of the unitary matrix of the discrete-time quantum walk. Let $G(V,E)$ be a sample graph and $\boldsymbol{U}$ be the unitary matrix associated with the discrete-time quantum walk on $G(V,E)$. The Perron-Frobenius operator $\boldsymbol{U}$ of $G(V,E)$ is

$$\boldsymbol{T} = \boldsymbol{S}^{+}(\boldsymbol{U}^{\top}). \tag{4}$$

Def. 1, Def. 2 and Eq.(4) show that the the discrete-time quantum walk and the Perron-Frobenius operator (i.e., the directed line graph) are correlated. For a graph $G(V,E)$ and its directed line graph $OLG(V_L, E_{dL})$, V_L is just the state space of the discrete-time quantum walk on $G(V,E)$, i.e., each vertex in $OLG(V_L, E_{dL})$ corresponds to a unique directed arc residing on the corresponding edge in $G(V,E)$. Moreover, if there is a directed edge from a vertex $v_L \in V_L$ to a vertex $u_L \in V_L$, the transition of the quantum walk on $G(V,E)$ is allowed from the arc corresponding to v_L to the arc corresponding to u_L, and vice versa. As a result, the discrete-time quantum walk on a graph can also be seen as a walk performed on its directed line graph. The state space of the walk is the vertex set of the line graph, and the transitions are constrained by the connections between pairs of vertices in the line graph.

Furthermore, in Bai et al. (2014a); Ren et al. (2011a), Bai et al. observed that the directed line graph possesses some interesting properties that are not available in the original graph. For instance, compared to the original graph the line graph spans a higher dimensional feature space and thus exposes richer graph characteristics. This is because the cardinality of the vertex set for the line graph is greater than, or at least equal to, that of the original graph. This property suggests that the discrete-time quantum walk may reflect richer graph characteristics than the continuous-time quantum walk on the original graph.

Finally, note that since the discrete-time quantum walk can be seen as a walk on the line graph and the state space of the walk is the vertex set of the line graph, we propose to use the rooting of the degree distribution of the line graph as the initial state of the discrete-time quantum walk.

2.3. From Quantum Walks to Density Matrices

In quantum mechanics, a pure state can be described as a single ket vector. A quantum system, however, can also be in a mixed state, i.e., a statistical ensemble of pure quantum states $|\psi_i\rangle$, each with probability p_i. The density matrix of such a system is defined as

$$\rho = \sum_i p_i \, |\psi_i\rangle \langle\psi_i| \tag{5}$$

Let $|\psi_t\rangle$ denote the state corresponding to a discrete-time quantum walk that has evolved from time $t = 0$ to time $t = T$. We define the time-averaged density matrix ρ_G^T for $G(V,E)$ as

$$\rho_G^T = \frac{1}{T+1} \sum_{t=0}^{T} |\psi_t\rangle \langle\psi_t| \, . \tag{6}$$

Since $|\psi_t\rangle = \boldsymbol{U}^t |\psi_0\rangle$, where $\boldsymbol{U}$ is the transition matrix of the discrete-time quantum walk, Eq.(6) can be re-written in terms of the initial state $|\psi_0\rangle$ as

$$\rho_G^T = \frac{1}{T+1} \sum_{t=0}^{T} (\boldsymbol{U})^t \, |\psi_0\rangle \langle\psi_0| \, (\boldsymbol{U}^{\top})^t. \tag{7}$$

The density matrix ρ_G^T describes a quantum system that has an equal probability of being in each of the pure states defined by the evolution of the discrete-time quantum walk from step $t = 0$ to step $t = T$.

2.4. The Quantum Jensen-Shannon Divergence

In quantum mechanics, the *von Neumann entropy* Nielsen and Chuang (2010) H_N of a density matrix ρ is defined as $H_N = -\mathrm{tr}(\rho \log \rho) = -\sum_i \xi_i \ln \xi_i$, where $\xi_1, \ldots, \xi_n$ denote the eigenvalues of ρ. Note that if the quantum system is in a pure state $|\psi_i\rangle$ with probability $p_i = 1$, then the Von Neumann entropy $H_N(\rho) = -\mathrm{tr}(\rho \log \rho)$ is zero. On the other hand, a mixed state generally has a non-zero Von Neumann entropy associated with its density matrix. Here we propose to compute the von Neumann entropy for each graph using the density matrix defined in Eq.(7). Consider a graph $G(V,E)$, the von Neumann entropy of $G(V,E)$ is defined as

$$H_N(\rho_G^T) = -\mathrm{tr}(\rho_G^T \log \rho_G^T) = -\sum_{j}^{|V|} \lambda_j^G \log \lambda_j^G, \tag{8}$$

where $\lambda_1^G, \ldots, \lambda_j^G, \ldots, \lambda_{|V|}^G$ are the eigenvalues of ρ_G^T.

With the Von Neumann entropy to hand, we can compute the *quantum Jensen-Shannon divergence* between two density operators ρ and σ Lamberti et al. (2008). The quantum Jensen-Shannon divergence has recently been developed as a generalization of the classical Jensen-Shannon divergence to quantum states Lamberti et al. (2008). Given two density operators ρ and σ, the quantum Jensen-Shannon divergence between them is defined as

$$D_{QJS}(\rho, \sigma) = H_N\Big(\frac{\rho + \sigma}{2}\Big) - \frac{1}{2} H_N(\rho) - \frac{1}{2} H_N(\sigma). \tag{9}$$

D_{QJS} is always well defined, symmetric, negative definite and bounded, i.e., $0 \leq D_{QJS} \leq 1$ Lamberti et al. (2008).

3. Graph Simplification Based on Commute Time

Let A denote the adjacency matrix of a graph $G(V,E)$. This is the matrix with $A(u,v) = 1$ if $(u,v) \in E$, and zero otherwise. The degree matrix D is the diagonal matrix with diagonal entries $D(u,u) = \sum_v A(u,v)$. Then, the graph Laplacian is given by $L = D - A$. The spectral decomposition of the Laplacian is $L = \Phi \Lambda \Phi^T$, where Φ is the $n \times n$ matrix $\Phi = (\phi_1 | \phi_2 | ... | \phi_n)$ with the ordered eigenvectors as columns and $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, ..., \lambda_n)$

is the $n \times n$ diagonal matrix with the ordered eigenvalues as elements, such that $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$.

The heat equation defines the dynamics of a diffusion process over a graph. This is a partial differential equation associated with the graph Laplacian, i.e.,

$$\frac{\partial \mathcal{H}_t}{\partial t} = -L\mathcal{H}_t, \tag{10}$$

where $\mathcal{H}_t$ is the heat kernel at time t. The solution of the heat equation is

$$\mathcal{H}_t = \exp(-Lt) = \Phi \exp(-\Lambda t)\Phi^T, \tag{11}$$

The evolution of a classical continuous-time random walk on the graph is also governed by the heat equation. Let p_t be the probability state vector of the walk, i.e., the vector whose components are the probabilities of the walk visiting the nodes of the graph at time t. Given an initial distribution p_0, the state vector of the walk at time t is

$$p_t = \mathcal{H}_t p_0. \tag{12}$$

The hitting time $O(u, v)$ of a random walk is defined as the expected number of steps to go from node u to node v. The commute time is similarly defined as the expected number steps to go from u to v, and then return to u. For an undirected graph, this is simply twice the hitting time. It can be shown that the hitting time can be written in terms of the eigendecomposition of the normalized Laplacian Qiu and Hancock (2007)

$$O(u, v) = \frac{1}{4}\sum_{j=2}^{n} \frac{1}{\lambda_j}(\phi_j(u) - \phi_j(v))^2, \tag{13}$$

where n is the number of nodes in the graph.

Qiu et al. show that it is possible to simplify the structure of a graph by computing the minimum spanning tree over the commute time matrix Qiu and Hancock (2007). This reduces the number of edges of the graph to $n - 1$, while retaining the characteristic structural information of the original graph. In fact, it can be shown that the simplified graph is robust with respect to small structural variations of the original graph Qiu and Hancock (2007).

Given a graph $G(V, E)$ we begin by computing the associated commute time matrix CT with entries $CT(u, v) = 2O(u, v)$. With the commute time matrix to hand, we construct a complete weighted graph $\mathbf{G}(\mathbf{V}, \mathbf{E})$ over the same vertex set of G, i.e., $\mathbf{V} = V$. The weight of the edge between a pair of vertices in $\mathbf{G}$ is the commute time between the vertices of the original graph. Using Prim's method Prim (1957), we can compute the minimum spanning tree $\mathbb{G}(\mathbb{V}, \mathbb{E})$ over $\mathbf{G}$, where the root node is the node v with minimum weight

$$W(u) = \sum_{u \in \mathbb{V}} CT(u, v). \tag{14}$$

In other words, the root vertex of the spanning tree is usually the most frequently visited node, since it has the lowest average commute time to the other nodes. From a structural point of view, this can be interpreted as the vertex being located near the centre of the original graph Qiu and Hancock (2007).

4. Graph Kernels from Discrete-time Quantum Walks

We now propose a graph kenel based on the quantum Jensen-Shannon divergence between discrete-time quantum walks. In order to reduced the computational complexity of the kernel, we also propose an alternative version of the kernel where the original graphs are first simplified as explained in Section 3.

Definition 3 (Kernel on Original Graphs): Given a set of graphs $\{G_1, \ldots, G_a, \ldots, G_b, \ldots, G_N\}$, we simulate a discrete-time quantum walk on each $G_a(V_a, E_a)$ and $G_b(V_b, E_b)$ for $t = 0, 1, \ldots, T$. Then, the density matrices $\rho^S_{G;a}$ and $\sigma^T_{G;b}$ associated with $G_a(V_a, E_a)$ and $G_b(V_b, E_b)$ can be computed using Eq.(7). With the density matrices to hand, the quantum Jensen-Shannon divergence $D_{QJS}(\rho_{G;a}, \sigma_{G;b})$ is computed as in Eq.(9). Finally, the quantum Jensen-Shannon kernel $k_{QJS}(G_a, G_b)$ between $G_a(V_a, E_a)$ and $G_b(V_b, E_b)$ is defined as

$$\begin{aligned} k_{QJS}(G_a, G_b) &= \exp(-\alpha D_{QJS}(\rho^T_{G;a}, \sigma^T_{G;b})) \\ &= \exp\{-\alpha H_N\Big(\frac{\rho^T_{G;a} + \sigma^T_{G;b}}{2}\Big) + \alpha\frac{1}{2}H_N(\rho^T_{G;a}) + \alpha\frac{1}{2}H_N(\sigma^T_{G;b})\}. \end{aligned} \tag{15}$$

where α is a decay factor satisfying $0 \leq \alpha \leq 1$, and $H_N(\cdot)$ is the von Neumann entropy defined in Eq.(8). For simplification, in this work we set α as 1. Note that if the graphs have different sizes, we add a number of disconnected nodes to the smaller graphs until their size is the same. □

Lemma *The quantum Jensen-Shannon kernel k_{QJS} is positive definite* ***pd****.*

Proof This follows from the definitions in Lamberti et al. (2008); Majtey et al. (2005). The quantum Jensen-Shannon divergence between a pair of density operators is a symmetric dissimilarity measure Lamberti et al. (2008). The proposed kernel k_{QJS} is computed as the negative exponential of the quantum Jensen-Shannon divergence, and thus it is **pd** Majtey et al. (2005). ■

For a pair of graphs with n vertices and m edges, the computation of the quantum kernel k_{QJS} has time complexity $O(m^3)$. This is because the state space of the discrete-time quantum walk is the vertex set of its line graph. The number of the vertices of the line graph is twice in the number of the edges of the original graph. Thus, the size of the density matrix associated with the quantum walk is $2m \times 2m$. Since the von Neumann entropy relies on the eigen-decomposition of the density matrix. The whole time complexity of the kernel k_{QJS} is $O(m^3)$. For the worst case, i.e., a pair of graphs are both complete graphs, the edge number of each graph is $m = \frac{n(n-1)}{2}$. Thus, the sizes of the quantum walk state space for the graphs are both $n(n - 1)$. As a result, for the worst case the time complexity of the kernel is $O(n^6)$, which implies that we can efficiently compute the proposed kernel only for relatively small graphs. To overcome this inefficiency, we propose to simplify each original graph using the approach of Qiu and Hancock (2007), thus reducing the complexity to $O(n^3)$.

Definition 4 (Kernel on Spanning Trees): For a pair of graphs $G_a(V_a, E_a)$ and $G_b(V_b, E_b)$, we construct their minimum spanning trees $\mathbb{G}_a(\mathbb{V}_a, \mathbb{E}_a)$ and $\mathbb{G}_b(\mathbb{V}_b, \mathbb{E}_b)$ from their commute

Table 1. Summary statistics for the graph datasets.

Datasets	MUTAG	PPIs	PTC	COIL5	Shock	GatorBait	CATH1	NCI1
Max # vertices	28	232	109	241	33	509	568	111
Min # vertices	10	3	2	72	4	260	44	3
Avg # vertices	17.93	109.60	25.60	144.90	13.16	367.36	205.72	29.87
Max # edges	33	1503	108	702	32	1266	2356	119
Min # edges	10	2	1	206	3	576	145	2
Avg # edges	19.79	432.18	25.96	419	12.16	842.79	819.86	32.30
# graphs	188	86	344	360	150	66	719	4110
# classes	2	2	2	5	10	10	2	2
Avg # edges/Avg # vertices	1.10	3.94	1.01	2.89	0.92	2.29	3.99	1.08

time matrices as described in Section 3. The quantum Jensen-Shannon kernel k_{QJST} between $G_a(V_a, E_a)$ and $G_b(V_b, E_b)$ is then defined as

$$k_{QJST}(G_a, G_b) = k_{QJS}(\mathbb{G}_a, \mathbb{G}_b), \qquad (16)$$

where k_{QJS} is compute as in Eq.(15).

The commute time captures the structural properties of the original graphs and it's robust under perturbation of their structural. Thus, the minimum spanning tree constructed on the commute time matrix can reflect the characteristic structural information of the original graph while yielding a sparser structure. Note that the computation of the commute time matrix is based on the eigendecomposition of the normalized Laplacian. Therefore, the graph simplification step has time complexity $O(n^3)$. As a result, the time complexity of the quantum kernel k_{QJST} is $O(n^3)$, a considerable improvement from the original $O(n^6)$ complexity of the kernel.

5. Experimental Evaluations

In this section, we evaluate empirically the performance of the proposed kernels on standard graph datasets. To this end, we compare our kernels with a number of commonly used graph kernels.

5.1. Graph Datasets

We explore the proposed kernels on five standard graph datasets from bioinformatics and computer vision. These datasets include: MUTAG, PPIs, PTC(MR), COIL5, Shock, CATH1 and NCI1. Table 1 shows a summary of statistics on these datasets.

MUTAG: The MUTAG dataset consists of graphs representing 188 chemical compounds labeled according to whether or not they affect the frequency of genetic mutations in the bacterium *Salmonella typhimuriums* and aims to predict whether each compound is associated with mutagenicity.

PPIs: The PPIs dataset consists of protein-protein interaction networks (PPIs). The graphs describe the interactions between *histidine kinase* in different species of bacteria. There are 219 PPIs in this dataset and they are collected from 5 different kinds of bacteria. Here we restrict out analysis to two kinds of bacteria.

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). These graphs are very small, i.e., 20−30 vertices, and sparsem, i.e., 25 – 40 edges. We select the graphs of male rats (MR) for evaluation. There are 344 test graphs in the MR class.

COIL5: We create a dataset referred to as COIL5 from the COIL image database. The COIL database consists of images of 100 3D objects. In our experiments, we use the images for the first five objects. For each of these objects 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. Each vertex is used as the seed of a Voronoi region, which expands radially with a constant speed. The linear collision fronts of the regions delineate the image plane into polygons, and the Delaunay graph is the region adjacency graph for the Voronoi polygons.

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.

GatorBait: the original GatorBait dataset contains 100 shapes representing fishes from 30 different classes Biasotti et al. (2003). In the original dataset, 10 of these classes contain a single element, and only 10 classes contain at least four shapes. In these experiments, we restrict our attention to the 66 graphs belonding to the 10 classes that contain at least 4 shape instances. For each of these 66 shapes, we extract the 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 intraclass variability in many cases. Therefore, the database, though having only 66 samples, plays a challenging role in testing graph classification. The number of maximum, minimum and average vertices for the dataset are 509, 260 and 367.36.

CATH1: The CATH1 dataset consists of proteins in the same class (i.e Mixed Alpha-Beta), but the proteins have different architectures (i.e. Alpha-Beta Barrel vs. 2-layer Sandwich). There are 712 test graphs in the CATH1 dataset.

NCI1: The NCI1 dataset consists 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. There are 4110 graph based structures in NCI1.

Table 2. Classification accuracy (in % ± standar error) on the graph datasets.

Datasets	MUTAG	PPIs	PTC(MR)	COIL5	Shock	GatorBait	CATH1	NCI1
QJSK	**83.83** ± 0.49	70.57 ± 1.20	**58.23** ± 0.80	69.78 ± 0.37	**44.86** ± 0.64	14.33 ± 0.90	−−	67.40 ± 0.20
QJSKT	81.55 ± 0.53	68.12 ± 0.84	57.44 ± 0.36	**70.15** ± 0.63	44.00 ± 1.20	**20.00** ± 1.19	**98.53** ± 0.18	67.00 ± 0.15
QJSU	82.72 ± 0.44	69.50 ± 1.20	56.70 ± 0.49	70.11 ± 0.61	40.60 ± 0.92	7.77 ± 0.41	98.12 ± 0.19	69.09 ± 0.20
WL	82.05 ± 0.57	**78.50** ± 1.40	56.05 ± 0.51	33.16 ± 1.01	36.40 ± 1.00	17.00 ± 1.01	94.36 ± 0.14	**80.68** ± 0.27
SPGK	83.38 ± 0.81	61.12 ± 1.09	56.55 ± 0.53	69.66 ± 0.52	37.88 ± 0.93	18.83 ± 0.96	98.32 ± 0.17	74.21 ± 0.30
JSGK	83.11 ± 0.80	57.87 ± 1.36	57.29 ± 0.41	69.13 ± 0.79	21.73 ± 0.76	13.16 ± 0.97	98.02 ± 0.11	62.50 ± 0.33
BRWK	77.50 ± 0.75	53.50 ± 1.47	53.97 ± 0.31	14.63 ± 0.21	0.33 ± 0.37	−−	−−	60.34 ± 0.17
RWGK	80.77 ± 0.72	55.00 ± 0.88	55.91 ± 0.37	20.80 ± 0.47	2.26 ± 1.01	−−	−−	−−

Table 3. CPU time usage for the kernel matrices computation in seconds.

Datasets	MUTAG	PPIs	PTC(MR)	COIL5	Shock	GatorBait	CATH1	NCI1
QJSK	$1.2 \cdot 10^1$	$1.4 \cdot 10^4$	$1.1 \cdot 10^2$	$6.0 \cdot 10^4$	$0.5 \cdot 10^1$	$4.8 \cdot 10^3$	> 1day	$1.6 \cdot 10^4$
QJSKT	$2.9 \cdot 10^1$	$1.5 \cdot 10^2$	$1.7 \cdot 10^2$	$1.9 \cdot 10^3$	$1.2 \cdot 10^1$	$4.8 \cdot 10^2$	$3.9 \cdot 10^4$	$1.4 \cdot 10^4$
QJSU	$2.0 \cdot 10^1$	$5.9 \cdot 10^1$	$1.1 \cdot 10^2$	$1.1 \cdot 10^3$	$1.4 \cdot 10^1$	$2.7 \cdot 10^3$	$3.7 \cdot 10^4$	$1.1 \cdot 10^4$
WL	$0.4 \cdot 10^1$	$1.3 \cdot 10^1$	$1.1 \cdot 10^1$	$6.5 \cdot 10^1$	$0.3 \cdot 10^1$	$3.5 \cdot 10^1$	$1.6 \cdot 10^2$	$1.5 \cdot 10^2$
SPGK	$0.1 \cdot 10^1$	$0.7 \cdot 10^1$	$0.1 \cdot 10^1$	$3.1 \cdot 10^1$	$0.1 \cdot 10^1$	$0.3 \cdot 10^1$	$1.3 \cdot 10^1$	$8.3 \cdot 10^1$
JSGK	$0.1 \cdot 10^1$	$0.1 \cdot 10^1$	$0.1 \cdot 10^1$	$0.1 \cdot 10^1$	$0.1 \cdot 10^1$	$0.1 \cdot 10^1$	$0.1 \cdot 10^1$	$0.1 \cdot 10^1$
BRWK	$0.1 \cdot 10^1$	$8.6 \cdot 10^2$	$0.3 \cdot 10^1$	$1.1 \cdot 10^3$	$0.8 \cdot 10^1$	> 1day	> 1day	$4.1 \cdot 10^2$
RWGK	$4.6 \cdot 10^1$	$6.7 \cdot 10^1$	$1.5 \cdot 10^2$	$1.2 \cdot 10^3$	$2.3 \cdot 10^1$	> 1day	> 1day	> 1day

5.2. Experiments on Standard Graph Datasets

Experimental Setup: We compare the performance of the proposed kernel (QJSK) as well as the one on minimum spanning trees (QJSKT) with that of several alternative state-of-the-art graph kernels. These kernels include 1) the unaligned quantum Jensen-Shannon kernel (UQJS) associated with the continuous-time quantum walk Bai et al. (2015a), 2) the Weisfeiler-Lehman subtree kernel (WL) Shervashidze et al. (2010), 3) the shortest path graph kernel (SPGK) Borgwardt and Kriegel (2005), 4) the Jensen-Shannon graph kernel associated with the steady state random walk (JSGK) Bai and Hancock (2013), 5) the backtrackless random walk kernel using the Ihara zeta function based cycles (BRWK) Aziz et al. (2013), and 6) the random-walk graph kernel Kashima et al. (2003). For the proposed kernels, we let $T = 30$. In fact, as we let $T \geqslant 30$ we observe that the von Neumann entropy of the density matrices reaches an asymptote. While the optimal procedure would be that of selecting the value of T through cross-validation, the computational complexity of the kernel makes it unfeasible to do so. Moreover, previous work has shown that letting $T \to \infty$ allows us to achieve a good trade-off in terms of accuracy and computational effort Rossi et al. (2013b, 2015). For the Weisfeiler-Lehman subtree kernel, we set the dimension of the number of iterations for the Weisfeiler-Lehman isomorphism test to 10. Based on the definition in Shervashidze et al. (2010), this means that we compute 10 different Weisfeiler-Lehman subtree kernel matrices, i.e., $k(1), k(2), \ldots, k(10)$, corresponding to different subtree heights $h(h = 1, 2, \ldots, 10)$. Note that the WL kernel is able to accommodate attributed graphs. In our experiments, we use the vertex degree as a vertex label for the WL kernel.

Given these datasets and kernels, we perform a 10-fold cross-validation using a C-Support Vector Machine (C-SVM) to evaluate the classification accuracies of the different kernels. More specifically, we use the C-SVM implementation of LIBSVM. For each class, we use 90% of the samples for training and the remaining 10% for testing. The parameters of the C-SVMs are optimized separately on the training set for each dataset. We report the average classification accuracies (± standard error) of each kernel in Table 2 and the time usage of computing the kernel matrices for each kernel in Table 3. The time usage is measured under Matlab R2011a running on a 2.5GHz Intel 2-Core processor, i.e., i5-3210m.

Results: Overall, in terms of classification accuracy the QJSK and QJSKT kernels outperform or are competitive with the state-of-the-art kernels. In particular, the classification accuracy of our quantum kernels is significantly better than that of the graph kernels based on classical random walks and backtrackless random walks, over all the datasets. This suggests that the proposed kernels can better capture the structural characteristics of the graphs. With respect to the quantum Jensen-Shannon kernel based on continuous-time quantum walks, we observe a significant improvement on the PTC, PPIs, GatorBait, CATH1 and Shock datasets. This is because a discrete-time quantum walk can be seen as a walk on the line graph, and therefore it can reflect richer graph characteristics than its continuous-time version. Furthermore, we observe that the performance of the QJSK kernel is marginally better than that of the QJSKT kernel. This is because the QJSKT kernel is computed on the minimum spanning trees, which inevitably lead to a partial loss of structural information. On the other hand, the performance of the QJSKT kernel is still competitive with respect to the QJSK kernel. This is because the spanning trees are constructed from the commute time matrix on the original graphs. As we observed in the previous sections, the commute time embedding is robust with respect to structural noise and it is well suited to capture the fundamental structural characteristics of the original graphs. As a result, the QJSKT kernel is still able to capture the dominant structural similarity between the original graphs while considerably reducing the computational complexity. Interestingly, we observe that on the GatorBait dataset the QJSKT kernel performs significantly better than the QJSK kernel. We argue that this is due to the fact that the graph simplification step

Table 4. Win/Loss/Tie matrix for the kernels used in the experiments.

Kernels	**QJSK**	**QJSKT**	QJSU	WL	SPGK	JSGK	BRWK	RWGK	Total
QJSK		3/0/2	2/0/3	4/1/0	3/0/2	3/0/2	5/0/0	5/0/0	**25/1/09**
QJSKT	0/2/3		2/2/1	3/1/1	3/0/2	3/1/1	5/0/0	5/0/0	*21/6/8*
QJSU	0/3/2	1/2/2		3/1/1	2/0/3	3/0/2	5/0/0	4/0/1	18/6/11
WL	1/4/0	1/3/1	1/3/1		1/3/1	2/3/0	5/0/0	4/0/1	15/16/4
SPGK	0/3/2	0/3/2	0/2/3	3/1/1		3/1/1	5/0/0	5/0/0	16/10/9
JSGK	0/3/2	1/3/1	0/3/2	3/2/0	1/3/1		5/0/0	5/0/0	15/14/6
BRWK	0/5/0	0/5/0	0/5/0	0/5/0	0/5/0	0/5/0		0/5/0	0/35/0
RWGK	0/5/0	0/5/0	0/4/1	0/4/1	0/4/1	0/5/0	5/0/0		0/32/3

in some cases may be able to separate the noise from the core structure of a graph, thus improving the classification accuracy.

In terms of time usage, the QJSK and QJSKT kernels are not the fastest kernels. This is because the time complexity of the proposed QJSK and QJSKT kernels is cubic in the number of edges of the graph. In contrast, the time complexity of other alternative kernels is cubic or quadratic in the number of vertices. Recall that for a connected graph with n vertices, the number of edges is between $n(n-1)/2$ and $n-1$. As a result, the computation of the QJSK and QJSKT kernels is more expensive than most alternative kernels. However, we observe that both kernels are computed in polynomial time. At the same time, the state space of the QJSK and QJSKT kernels is the set of arcs rather than the set of vertices, which allows to represent graphs into a higher dimensional space than alternative kernels. Thus, for most datasets, the classification accuracy of the proposed kernels is better or at least competitive to that of alternative kernels. Finally, note that as the size of the input graphs and the edge numbers grow, the time usage associated with the QJSKT kernel is significantly less than that associated with the QJSK kernel. This indicates that the QJSKT kernel represents a good trade-off between classification performance and computational efficiency, compared to the QJSK kernel. In particular, when the graphs are very dense, as in the PPI and CATH1 datasets, the QJSK kernel is orders of magnitude slower than the QJSKT as well as the QJSU kernel.

5.3. Statistical Analysis of Proposed and Alternative Kernels

Table 2 indicates that the proposed QJSK and QJSKT kernels significantly outperform the BRWK and RWGK kernels. However, for some datasets, the WL, SPGK and QJSU kernels are competitive with respect to the proposed kernels. In other words, there is no kernel that performs best on any dataset. To evaluate the best kernel over all datasets, we count the number of datasets on which each kernel performs better, worse, or equally to other kernels. Note that some kernels cannot complete the kernel matrix computation on some datasets. Therefore, we restrict our analysis on those datasets where all the kernels complete the computation.

Table 4 shows the Win/Loss/Tie matrix for the kernels used in the experiments. The (i, j)th element of the matrix shows the number of datasets where the kernel corresponding to the ith row has won/lost/tied against the kernel corresponding to the jth column. A tie is defined as a dataset on which the difference in classification accuracy between two kernels is not statistically significant. The last column of Table 4 shows the total number of wins/losses/ties for a given kernel, and the best and second best performing kernels are highlighted in bold and italic, respectively.

The experimental results show that the QJSK and QJSKT kernels clearly outperform the alternative kernels, with the QJSK kernel performing worse than the alternative kernels in only one case. However, recall from Table 3 that the QJSK kernel is considerably slower than the alternative kernels. In other words, the QJSK kernel is a better choice for small datasets where the graphs are relatively sparse. In particular, note that for the Shock dataset, where the graphs are trees, the QJSK and QJSKT kernels coincide, while the runtime of the latter is higher due to the commute time distance computation. On the other hand, the QJSKT kernel is less computationally demanding, thus making it more applicable to larger and denser graph datasets. At the same time, Table 4 shows that the QJSKT kernel performs favourably when compared with the alternative kernels, underlying once more the optimal trade-off between time usage and classification accuracy.

We also observe that the proposed kernels outperform the alternative kernels on all the computer vision datasets considered in this study (COIL5, Shock, and GatorBait), whereas the performance of the WL kernel seems to be consistently high on the bioinformatics datasets. Finally we should stress that the proposed kernels do not take potential node or edge attributes into account, while some of the alternative kernels do. Hence, the QJSK and QJST kernels are better suited for the analysis of unattributed graphs.

6. Conclusion and Future Work

In this paper, we have developed a new family of quantum Jensen-Shannon kernels for graphs using the quantum Jensen-Shannon divergence and discrete-time quantum walks. The kernels can reflect richer graph characteristics than kernels based on continuous-time quantum walks. The experimental results demonstrate the effectiveness and efficiency of the proposed kernels.

It is clear from their definition that both the proposed graph kernels are not permutation invariant. This is because the quantum Jensen-Shannon divergence requires mixing the quantum states corresponding to the walks on the input graphs. When constructing this mixed state, however, we do not take into account the correspondences between the vertices of two graphs. This in turn influences the estimation of the mixed state entropy and thus the precise kernel measure between the graphs. Future work will be aimed at developing our kernels one step further by introducing an alignment step that captures the correspondence information between the vertices of the graphs.

Furthermore, unlike the quantum Jensen-Shannon kernels associated with continuous-time quantum walks Bai et al. (2015a); Rossi et al. (2015), for which a closed form solution of the density matrix is available, the proposed kernels need to explicitly simulate the evolution of the discrete-time quantum walk step by step. As a result, the proposed kernels may be computational inefficient when applied on large graphs. To address this problem, future work will be aimed at developing a closed form solution of the density matrix associated with discrete-time quantum walks.

Finally, the proposed kernels cannot accommodate attributed graphs. In our previous work Bai et al. (2014b), we have developed an attributed graph kernel associated with continuous-time quantum walks. Similarly, future work will investigate the possibility of extending the discrete-time quantum walk kernels to deal with attributed graphs.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant no.61503422 and 61402389), the Open Projects Program of National Laboratory of Pattern Recognition, and the program for innovation research in Central University of Finance and Economics. Edwin R. Hancock is supported by a Royal Society Wolfson Research Merit Award.

References

Aziz, F., Wilson, R.C., Hancock, E.R., 2013. Backtrackless walks on a graph. IEEE Transactions on Neural Networks and Learning Systems 24, 977–989.

Bach, F.R., 2008. Graph kernels between point clouds, in: Proceedings of ICML, pp. 25–32.

Bai, L., Hancock, E.R., 2013. Graph kernels from the jensen-shannon divergence. Journal of Mathematical Imaging and Vision 47, 60–69.

Bai, L., Hancock, E.R., 2016. Fast depth-based subgraph kernels for unattributed graphs. Pattern Recognition 50, 233–245.

Bai, L., Ren, P., Hancock, E.R., 2014a. A hypergraph kernel from isomorphism tests, in: Proceedings of ICPR, pp. 3880–3885.

Bai, L., Rossi, L., Bunke, H., Hancock, E.R., 2014b. Attributed graph kernels using the jensen-tsallis q-differences, in: Proceedings of ECML-PKDD, pp. 99–114.

Bai, L., Rossi, L., Torsello, A., Hancock, E.R., 2015a. A quantum jensen-shannon graph kernel for unattributed graphs. Pattern Recognition 48, 344–355.

Bai, L., Rossi, L., Zhang, Z., Hancock, E.R., 2015b. An aligned subtree kernel for weighted graphs, in: Proceedings of ICML, pp. 30–39.

Bai, L., Zhang, Z., Wang, C., Bai, X., Hancock, E.R., 2015c. A graph kernel based on the jensen-shannon representation alignment, in: Proceedings of IJCAI, pp. 3322–3328.

Barra, V., Biasotti, S., 2014. 3d shape retrieval and classification using multiple kernel learning on extended reeb graphs. The Visual Computer 30, 1247–1259.

Biasotti, S., Marini, S., Mortara, M., Patanè, G., Spagnuolo, M., Falcidieno, B., 2003. 3d shape matching through topological structures, in: Proceedings of DGCI, pp. 194–203.

Borgwardt, K.M., Kriegel, H.P., 2005. Shortest-path kernels on graphs, in: Proceedings of the IEEE International Conference on Data Mining, pp. 74–81.

Borgwardt, K.M., Ong, C.S., Schönauer, S., Vishwanathan, S.V.N., Smola, A.J., Kriegel, H., 2005. Protein function prediction via graph kernels, in: Proceedings Thirteenth International Conference on Intelligent Systems for Molecular Biology 2005, Detroit, MI, USA, 25-29 June 2005, pp. 47–56.

Brun, L., Conte, D., Foggia, P., Vento, M., Villemin, D., 2010. Symbolic learning vs. graph kernels: An experimental comparison in a chemical application, in: Local Proceedings of the Fourteenth East-European Conference on Advances in Databases and Information Systems, Novi Sad, Serbia, September 20-24, 2010, pp. 31–40.

Emmert-Streib, F., Dehmer, M., 2011. Networks for systems biology: conceptual connection of data and function. IET Syst Biol. 5, 185–207.

Emms, D., Severini, S., Wilson, R.C., Hancock, E.R., 2009. Coined quantum walks lift the cospectrality of graphs and trees. Pattern Recognition 42, 1988–2002.

Farhi, E., Gutmann, S., 1998. Quantum computation and decision trees. Physical Review A 58, 915.

Foggia, P., Percannella, G., Vento, M., 2014. Graph matching and learning in pattern recognition in the last 10 years. International Journal of Pattern Recognition and Artificial Intelligence 28.

Gaidon, A., Harchaoui, Z., Schmid, C., 2011. A time series kernel for action recognition, in: Proceedings of BMVC, pp. 1–11.

Gärtner, T., 2003. A survey of kernels for structured data. SIGKDD Explorations 5, 49–58.

Gärtner, T., Flach, P., Wrobel, S., 2003. On graph kernels: Hardness results and efficient alternatives, in: Proceedings of COLT, pp. 129–143.

Harchaoui, Z., Bach, F., 2007. Image classification with segmentation graph kernels, in: Proceedings of CVPR.

Haussler, D., 1999. Convolution kernels on discrete structures, in: Technical Report UCS-CRL-99-10, Santa Cruz, CA, USA.

Kashima, H., Tsuda, K., Inokuchi, A., 2003. Marginalized kernels between labeled graphs, in: Proceedings of ICML, pp. 321–328.

L, G., 1996. A fast quantum mechanical algorithm for database search, in: Proceedings of ACM Symposium on the Theory of Computation, pp. 212–219.

Lamberti, P., Majtey, A., Borras, A., Casas, M., Plastino, A., 2008. Metric character of the quantum jensen-shannon divergence. Physical Review A 77, 052311.

Majtey, A., Lamberti, P., Prato, D., 2005. Jensen-shannon divergence as a measure of distinguishability between mixed quantum states. Physical Review A 72, 052310.

Nielsen, M., Chuang, I., 2010. Quantum computation and quantum information. Cambridge university press.

Prim, R.C., 1957. Shortest connection networks and some generalizations. Bell system technical journal 36, 1389–1401.

Qiu, H., Hancock, E.R., 2007. Graph simplification and matching using commute times. Pattern Recognition 40, 2874–2889.

Ren, P., Aleksic, T., Emms, D., Wilson, R.C., Hancock, E.R., 2011a. Quantum walks, ihara zeta functions and cospectrality in regular graphs. Quantum Information Process 10, 405–417.

Ren, P., Wilson, R.C., Hancock, E.R., 2011b. Graph characterization via ihara coefficients. IEEE Transactions on Neural Networks 22, 233–245.

Rossi, L., Torsello, A., Hancock, E.R., 2013a. Attributed graph similarity from the quantum jensen-shannon divergence, in: Similarity-Based Pattern Recognition. Springer, pp. 204–218.

Rossi, L., Torsello, A., Hancock, E.R., 2013b. A continuous-time quantum walk kernel for unattributed graphs, in: Proceedings of GbRPR, pp. 101–110.

Rossi, L., Torsello, A., Hancock, E.R., 2015. Measuring graph similarity through continuous-time quantum walks and the quantum jensen-shannon divergence. Physical Review E 91, 022815.

Schölkopf, B., Smola, A., 2002. Learning with Kernels. MIT Press.

Shervashidze, N., Schweitzer, P., van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M., 2010. Weisfeiler-lehman graph kernels. Journal of Machine Learning Research 1, 1–48.

Vento, M., 2015. A long trip in the charming world of graphs for pattern recognition. Pattern Recognition 48, 291–301.

Vishwanathan, S.V.N., Schraudolph, N.N., Kondor, R., Borgwardt, K.M., 2010. Graph kernels. Journal of Machine Learning Research 11, 1201–1242.

Wang, L., Sahbi, H., 2013. Directed acyclic graph kernels for action recognition, in: Proceedings of ICCV, pp. 3168–3175.