An Edge-Based Matching Kernel Through Discrete-Time Quantum Walks

Lu Bai[1(✉)], Zhihong Zhang[2], Peng Ren[3], Luca Rossi[4], and Edwin R. Hancock[5]

[1] School of Information, Central University of Finance and Economics, Beijing, China
bailu69@hotmail.com, lu@cs.york.ac.uk
[2] Software School, Xiamen University, Xiamen, Fujian, China
zhihong@xmu.edu.cn
[3] College of Information and Control Engineering, China University of Petroleum
(Huadong), Qingdao, Shandong Province, People's Republic of China
pengren@upc.edu.cn
[4] School of Computer Science, University of Birmingham, Birmingham, UK
l.rossi@cs.bham.ac.uk
[5] Department of Computer Science, University of York, York, UK
erh@cs.york.ac.uk

Abstract. In this paper, we propose a new edge-based matching kernel for graphs by using discrete-time quantum walks. To this end, we commence by transforming a graph into a directed line graph. The reasons of using the line graph structure are twofold. First, for a graph, its directed line graph is a dual representation and each vertex of the line graph represents a corresponding edge in the original graph. Second, we show that 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, i.e., the state space of the walk is the edges of the original graph. As a result, the directed line graph provides an elegant way of developing new edge-based matching kernel based on discrete-time quantum walks. For a pair of graphs, we compute the h-layer depth-based representation for each vertex of their directed line graphs by computing entropic signatures (computed from discrete-time quantum walks on the line graphs) on the family of K-layer expansion subgraphs rooted at the vertex, i.e., we compute the depth-based representations for edges of the original graphs through their directed line graphs. Based on the new representations, we define an edge-based matching method for the pair of graphs by aligning the h-layer depth-based representations computed through the directed line graphs. The new edge-based matching kernel is thus computed by counting the number of matched vertices identified by the matching method on the directed line graphs. Experiments on standard graph datasets demonstrate the effectiveness of our new kernel.

1 Introduction

Graph kernels are powerful tools for graph structure analysis in pattern recognition and machine learning [1]. The main advantage of using graph kernels is that

they characterize graph features in a high dimensional space and thus better preserve graph structures. Generally speaking, a graph kernel is a similarity measure between a pair of graphs [1]. To extend the large spectrum of kernel methods from the general machine learning domain to the graph domain, Haussler [2] has proposed a generic 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. Following this scenario, Kashima et al. [3] introduced the random walk kernel, which is based on the enumeration of common random walks between two graphs. Borgwardt et al. [4], on the other hand, proposed a shortest path kernel by counting the numbers of matching shortest paths over the graphs. Aziz et al. [5] introduced a backtrackless kernel using the cycles identified by the Ihara zeta function [6] in a pair of graphs. Shervashidze et al. [7] developed a fast subtree kernel by comparing pairs of subtrees identified by the Weisfeiler-Lehman algorithm. Some other alternative R-convolution kernels include a) the segmentation graph kernel developed by Harchaoui and Bach [8], b) the point cloud kernel developed by Bach [9], and c) the (hyper)graph kernel based on directed subtree isomorphism tests [10].

Unfortunately, R-convolution kernels tend to neglect the relative locations of substructures. This is because R-convolution kernels add an unit value to the kernel function by roughly identifying a pair of isomorphic substructures, i.e., any pair of isomorphic substructures will contribute an unit kernel value. As a result, the R-convolution kernels cannot establish reliable structural correspondences between the substructures. This drawback limits the precise kernel-based similarity measure for graphs.

To overcome the problem arising in existing R-convolution kernels, in our previous work [11,12], we have developed a new depth-based matching kernel for graphs. The depth-based matching kernel is based on aligning the h-layer depth-based representations around vertices of graphs (i.e., aligning the vertices of the graphs), and is computed by counting the number of matched vertex pairs. In [11], the depth-based matching kernel can be seen as an aligned subgraph kernel that encapsulates location correspondence information between pairwise inexact isomorphic h-layer expansion subgraphs. As a result, the depth-based matching kernel overcomes the shortcoming of neglecting location correspondences between substructures arising in R-convolution kernels.

In this work, we aim to develop our previous work in [11,12] one step further. We develop a new edge-based matching kernel for graphs based on discrete-time quantum walks. For a graph, we commence by transforming the graph into a directed line graph [6]. The reason of using the directed line graph is that the line graph is a dual representation of the original graph [6], i.e., the vertex of the directed line graph represents a corresponding edge of the original graph. Moreover, we show that 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, i.e., the state space of the walk is the edges of the original graphs (See details in Section 2). Finally,

the directed line graph may expose richer graph characteristics than the original graph [6]. As a result, the directed line graph provides an elegant way of developing new edge-based matching kernel based on discrete-time quantum walks, which not only reflects quantum-based information but also encapsulates richer characteristics of graphs. For a pair of graphs, we compute the h-layer depth-based representation for each vertex of their directed line graphs by computing entropic signatures (computed from discrete-time quantum walks on the line graphs) on the family of K-layer expansion subgraphs rooted at the vertex, i.e., we compute the depth-based representations for edges of the original graphs through their directed line graphs. Based on the new representations, we define an edge-based matching method for the pair of graphs by aligning the h-layer depth-based representations computed through the directed line graphs. The new edge-based matching kernel is thus computed by counting the number of matched vertices identified by the matching method on the directed line graphs. Experiments on standard graph datasets demonstrate the effectiveness of our new edge-based matching kernel.

2   Preliminary Concepts

2.1   Directed Line Graphs

In this subsection, we commence by introducing the concept of directed line graphs. The reasons of using the line graphs are threefold. First, for a graph, its directed line graph is a dual representation and each vertex of the line graph represents a corresponding edge in the original graph. Second, the required discrete-time quantum walk used in this work 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, i.e., the state space of the walk is the edges of the original graphs (see details in Section 2.3). Third, the directed line graph may expose richer graph characteristics than the original graph [6]. As a result, the directed line graph provides a way of developing new edge-based matching kernel based on discrete-time quantum walks, which not only reflects quantum-based information but also encapsulates richer characteristics of graphs (see details in Section 3).

Based on the definition of Ren et al. in [6], for a sample graph $G(V, E)$, the directed line graph $G_D(V_D, \overrightarrow{E}_D)$ is a dual representation of $G(V, E)$. To obtain $G_D(V_D, \overrightarrow{E}_D)$, we first construct the associated symmetric digraph $SDG(V, E_d)$ of $G(V, E)$, by replacing every edge $e(u, w) \in E(G)$ by a pair of reverse 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 $G_D(V_D, \overrightarrow{E}_D)$ is the directed graph with vertex set V_D and arc set $\overrightarrow{E}_D$ defined as

$$
\begin{aligned}
V_D &= E_d(SDG), \\
\overrightarrow{E}_D &= \{(e_d(u, v), e_d(v, w)) \in E_d(SDG) \times E_d(SDG) \mid u, v, w \in V, u \neq w\}.
\end{aligned} \tag{1}
$$

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 $G_D(V_D, \overrightarrow{E}_D)$.

2.2 Discrete-Time Quantum Walks

The discrete-time quantum walk is the quantum counterpart of the discrete-time classical random walk [13]. To simulate the evolution of a discrete-time quantum walk on a graph $G(V, E)$, we first replace each edge $e(u, v) \in E$ with a pair of directed arcs $e_d(u, v)$ and $e_d(v, u)$. This in turn ensures the reversibility of the quantum process. Let us 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 $|uv\rangle$. A general state of the walk is

$$|\psi\rangle = \sum_{e_d(u,v) \in E_d} \alpha_{uv} |uv\rangle, \tag{2}$$

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}.

The evolution of the state vector between the steps t and $t+1$ is determined 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 [14] as the transition matrix. Using the Grover diffusion matrices, the transition matrix $\boldsymbol{U}$ has entries

$$U_{(u,v),(w,x)} = \begin{cases} \frac{2}{d_x} - \delta_{ux}, & v = w; \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $U_{(u,v),(w,x)}$ 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 $|u_1v\rangle$, the Grover matrix assigns the same amplitudes to all transitions $|u_1v\rangle \rightarrow |vu_i\rangle$, and a different amplitude to the transition $|u_1v\rangle \rightarrow |vu_1\rangle$, where u_i denotes 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.(3) allows *destructive interference* to take place.

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 (or density operator) of such a system is defined as

$$\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i| \tag{4}$$

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

$$\rho_G^T = \frac{1}{T+1} \sum_{t=0}^{T} |\psi_t\rangle \langle\psi_t|. \tag{5}$$

Since $|\psi_t\rangle = \boldsymbol{U}^t |\psi_0\rangle$, where $\boldsymbol{U}$ is the transition matrix of the discrete-time quantum walk, Eq.(5) can be re-written in terms of the initial state $|\psi_0\rangle$ as

$$\rho_G^T = \frac{1}{T+1} \sum_{t=0}^{T} (\boldsymbol{U})^t |\psi_0\rangle \langle\psi_0| (\boldsymbol{U}^\top)^t. \tag{6}$$

As a result, 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 $t = T$. Note that, in fact, for a graph $G(V, E)$ the state space of the discrete-time quantum walk is the set of vertices of its directed line graph $G_D(V_D, \overrightarrow{E}_D)$ (see details in Section 2.3). As a result, based on the definitions in this Section, the time-averaged probability of the discrete-time quantum walk to visit a vertex $v_D \in V_D$ at time T is

$$P_{DQ}^T(v_D) = \rho_G^T(v_D, v_D). \tag{7}$$

where v_D corresponds to an arc $e_d(u, v)$ $(u, v \in V)$ residing on an edge $e \in E$.

2.3   The Relation to Perron-Frobenius Operators

The Perron-Frobenius operator (i.e., the adjacency matrix of the directed line graph) can be represented in terms of the transition matrix of discrete-time quantum walks. To show this connection, we first introduce the definition of the positive support of a matrix. 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

$$s_{i,j} = \begin{cases} 1, & M_{i,j} > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

where $1 \le i \le m, 1 \le j \le n$. Based on the definition in [15], we can re-define the Perron-Frobenius operator defined in Eq.(1) 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 $\boldsymbol{T}$ of $G(V, E)$ is

$$\boldsymbol{T} = \boldsymbol{S}^+(\boldsymbol{U}^\top). \tag{9}$$

Eq.(1), Eq.(8) and Eq.(9) show us how the discrete-time quantum walk and the Perron-Frobenius operator (i.e., the directed line graph) are co-related. For a graph $G(V, E)$ and its directed line graph $G_D(V_D, \overrightarrow{E}_D)$, V_D is just the state space of the discrete-time quantum walk on $G(V, E)$, i.e., each vertex in $G_D(V_D, \overrightarrow{E}_D)$ 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 $v_D \in V_D$ to a vertex $u_D \in V_D$, the transition of the quantum walk on $G(V, E)$ is allowed from the arc corresponding to v_D to the arc corresponding to u_D, 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 transition of the walk relies on the connections between pairs of vertices in the line graph.

Discussions. As we have stated, the directed line graph possesses some interesting properties that are not available on 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 suggests that the discrete-time quantum walk may reflect richer graph characteristics than the continuous-time quantum walk on the original graph [21]. Finally, 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 in-degree distribution of the line graph as the initial state of the walk.

2.4   Depth-Based Representations Based on Quantum Walks

For $G(V, E)$, let P_{DQ}^T be the probability distribution of discrete-time quantum walks visiting vertices of the directed line graph $G_D(V_D, \overrightarrow{E}_D)$. For G_D and a vertex $v_D \in V_D$, let a vertex set $N_{v_D}^K$ be defined as $N_{v_D}^K = \{u_D \in V_D \mid S_G(v_D, u_D) \leq K\}$, where $S_G(v_D, u_D)$ is the shortest path length between v_D and u_D. For G_D, the K-layer expansion subgraph $\mathcal{G}_{v_D}^K(\mathcal{V}_{v_D}^K; \overrightarrow{\mathcal{E}}_{v_D}^K)$ around v_D is

$$\begin{cases} \mathcal{V}_{v_D}^K & = \{u_D \in N_{v_D}^K\}; \\ \overrightarrow{\mathcal{E}}_{v_D}^K & = \{u_D, w_D \in N_{v_D}^K, \ (u_U, w_D) \in \overrightarrow{E}_D\}. \end{cases} \tag{10}$$

For G_D, the h-layer directed DB representation around $v_D \in V_D$ is defined as

$$\overrightarrow{DB}_{G_D}^h(v_D) = [H_E(\mathcal{G}_{v_D}^1), \cdots, H_E(\mathcal{G}_{v_D}^K), \cdots, H_E(\mathcal{G}_{v_D}^h)]^\top, \tag{11}$$

where $H_E(\mathcal{G}_{v_D}^K)$ is the Shannon entropic signature of $\mathcal{G}_{v_D}^K$ defined as

$$H_E(\mathcal{G}_{v_D}^K) = - \sum_{v_D \in \mathcal{V}_{v_D}^K} P_{DQ}^T(v_D) \log P_{DQ}^T(v_D), \tag{12}$$

where $P_{DQ}^T(v_D)$ defined in Eq.(7) is the probability of the discrete-time quantum walk visiting the vertex v_D in the directed line graph G_D. Note that, the Shannon entropic signature $H_E(\mathcal{G}_{v_D}^K)$ is not a strict Shannon entropy measure, since it is not computed by using all the probabilities of the discrete-time quantum walk visiting all the vertices in G_D. Finally, note that, since the vertices in G_D correspond to corresponding edges in G, the h-layer depth-based representations of vertices in G_D can be seen as the depth-based representations of edges in G.

3   An Edge-Based Matching Kernel for Graphs

3.1   Edge-Based Matching Through Discrete-Time Quantum Walks

As we have stated in Section 2.1, the directed line graph provides a way of defining an edge-based matching method for graphs based on discrete-time quantum

walks. Because, for an original graph and its directed line graph, each vertex of the line graph represents a corresponding edge in the original graph. Moreover, the discrete-time quantum walk can be seen as a walk on the line graph. For a pair of graphs G_p and G_q, we commence by computing the h-layer depth-based representations of the vertices in their directed line graphs $G_{D;p}(V_{D;p}, \overrightarrow{E}_{D;p})$ and $G_{D;q}(V_{D;q}, \overrightarrow{E}_{D;q})$, based on the discrete-time quantum walks on the line graphs. These representations can be seen as the vectorial signatures of corresponding edges of G_p and G_q. We compute the Euclidean distance between the h-layer depth-based representations $\overrightarrow{DB}^h(v_i)$ and $\overrightarrow{DB}^h(v_j)$ as the distance measure of the pairwise vertices v_i and u_j of the directed line graphs $G_{D;p}$ and $G_{D;q}$, respectively. The affinity matrix element $R(i,j)$ is defined as

$$R(i,j) = \| \overrightarrow{DB}^h(v_i) - \overrightarrow{DB}^h(u_j) \|_2 . \tag{13}$$

where R is a $|V_{D;p}| \times |V_{D;q}|$ matrix. The element $R(i,j)$ represents the dissimilarity between vertex v_i in $G_{D;p}$ and vertex u_j in $G_{D;q}$. The rows of $R(i,j)$ index the vertices of $G_{D;p}$, and the columns index the vertices of $G_{D;q}$. If $R(i,j)$ is the smallest element both in row i and in column j, there is a one-to-one correspondence between vertex v_i of $G_{D;p}$ and vertex u_j of $G_{D;q}$. We record the state of correspondence using the correspondence matrix $C \in \{0,1\}^{|V_{D;p}||V_{D;q}|}$ satisfying

$$C(i,j) = \begin{cases} 1 \text{ if } R(i,j) \text{ is the smallest element} \\ \quad \text{both in row } i \text{ and in column } j; \\ 0 \text{ otherwise.} \end{cases} \tag{14}$$

Eq.(14) implies that if $C(i,j) = 1$, the vertices v_i and v_j are matched. Note that, like the depth-based matching previously introduced in our previous work [12], for a pair of directed line graphs a vertex from a line graph may have more than one matched vertex in the other line graph. In our work, we assign each vertex from a line graph at most one vertex in the other line graph. To achieve this, we propose to randomly assign each vertex an unique matched vertex through the correspondence matrix C. We observe that, compared to the depth-based matching associating Hungarian algorithm [17] for the assignment, our strategy will not influence the effectiveness of the resulting kernel in Section 3.2, and the kernel will be more efficient.

3.2   An Edge-Based Matching Kernel

Definition 3.1 (The edge-based matching kernel) Consider G_p and G_q as a pair of sample graphs, $G_{D;p}(V_{D;p}, \overrightarrow{E}_{D;p})$ and $G_{D;q}(V_{D;q}, \overrightarrow{E}_{D;q})$ are their directed line graphs. Based on the definitions in Eq.(11), Eq.(13) and Eq.(14), we compute the correspondence matrix C. The edge-based matching kernel $k_{DQEB}^{(h,T)}$ using the h-layer depth-based representations of the line graphs that is computed based on discrete-time quantum walks evolved from time 0 to time T is

$$k_{DQEB}^{(h,T)}(G_p, G_q) = k_{DQEB}^{(h,T)}(G_{D;p}, G_{D;q}) = \sum_{i=1}^{|V_{D;p}|} \sum_{j=1}^{|V_{D;q}|} C(i,j). \tag{15}$$

which counts the number of matched vertex pairs between $G_{D;p}$ and $G_{D;q}$. Intuitively, the edge-based matching kernel $k_{DQEB}^{(h,T)}$ is positive definite. Because $k_{DQEB}^{(h,T)}$ counts pairs of matched vertices over the correspondence matrix C.   □

Discussions. Clearly, like our previous depth-based matching kernel [12], the edge-based matching kernel is also related to the depth-based representation defined in [18]. However, there are three significant differences. First, the depth-based representations in [18] are computed on original graphs. By contrast, the h-layer depth-based representations for graphs required in this work are computed based on discrete-time quantum walks on directed line graphs transformed from original graphs. Second, in [18], we only compute the depth-based representation rooted at a centroid vertex of an original graph which is identified by evaluating the variance of the shortest path lengths between vertices. By contrast, in this work, we compute the h-layer depth-based representation rooted at each vertex of the directed line graph. Third, the depth-based representation from the centroid vertex is a vectorial signature of an original graph, i.e., it is an embedding vector for the graph. Embedding a graph into a vector tends to approximate the structural correlations in a low dimensional space, and thus leads to information loss. By contrast, the edge-based matching kernel aligning the h-layer depth-based representation represents directed line graphs in a high dimensional space and thus better preserves graph structures.

Moreover, as we have stated, the directed line graph may expose richer graph characteristics. Since the cardinality of the vertex set for the directed line graph is much greater than, or at least equal to, that of the original graph. Moreover, the h-layer depth-based representations based on discrete-time quantum walks on directed line graphs encapsulate quantum information relying on the walks. As a result, the new edge-based matching kernel can not only encapsulate the correspondence information between edges of original graphs, but also reflect richer graph characteristics and quantum information through discrete-time quantum walks on directed line graphs. Finally, like our previous depth-based matching kernel [11], the new edge-based matching kernel can also be seen as an aligned subgraph kernel (details of the discussion can be found in [11]). Differently, the edge-based matching kernel identifies the locational correspondence between pairwise h-layer expansion subgraphs of the directed line graphs transformed from the original graphs. By contrast, our previous depth-based matching kernel identifies the locational correspondence between pairwise h-layer expansion subgraphs of the original graphs.

4   Experimental Results

We demonstrate the performance of our new kernel on standard graph datasets from computer vision databases. These datasets include BAR31, BSPHERE31, GEOD31, and SHOCK. Details of these datasets can be found as follows.

BAR31, BSPHERE31 and GEOD31. The SHREC 3D Shape database consists of 15 classes and 20 individuals per class, that is 300 shapes [19].

This is an usual 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 the all other points. 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.

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. The number of maximum, minimum and average vertices for the dataset are 33, 4 and 13.16 respectively.

Experimental Setup. We evaluate the performance of our new edge-based matching kernel through discrete-time quantum walks (DQMK), on graph classification problems. We also compare our kernel with several alternative state-of-the-art graph kernels. These graph kernels include 1) the depth-based matching kernel (DBMK) [11,12], 2) the Weisfeiler-Lehman subtree kernel (WLSK) [7], 3) the shortest path graph kernel (SPGK) [4], 4) the graphlet count graph kernel [20] with graphlet of size 4 (GCGK) [20], 5) the un-aligned quantum Jensen-Shannon kernel (UQJS) [21], and 6) the Jensen-Shannon graph kernel (JSGK) [22]. For the WLSK kernel, we set the highest dimension (i.e., the highest height of subtrees) of the Weisfeiler-Lehman isomorphism (for the WLSK kernel) as 10. For the DQMK kernel and the DBMK kernel, we set the highest layer of the required depth-based representation as 20 and 10 respectively. The reason for this is that the 10-layer or 20-layer expansion subgraph rooted at a vertex of an original graph (for the DBMK kernel) or a directed line graph (for the DQMK kernel) usually encapsulates most vertices of the original graph or the line graph. Moreover, for our DQMK kernel, we set the largest time T as 25. This is because when $T > 20$ the probabilities of discrete-time quantum walks visiting vertices on directed line graphs (i.e., edges in original graphs) tend to be stable. Finally, note that, some kernels (i.e., the WLSK kernel and the SPGK kernel) can accommodate vertex labels, we use the degree of a vertex as the label of the vertex. 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 to compute the classification accuracy, using LIBSVM [23]. We use nine samples for training and one for testing. All the C-SVMs were performed along with their parameters optimized on each dataset. We report the average classification accuracy ($\pm$ standard error) and the runtime for each kernel in Table 1 and Table 2, respectively. The runtime is measured under Matlab R2011a running on a 2.5GHz Intel 2-Core processor (i.e. i5-3210m).

Experimental Results. In terms of the classification accuracy from the C-SVM, we observe that our DQMK kernel can easily outperform all the alternative graph kernels excluding the DBMK kernel, on any dataset. The reasons

Table 1. Classification Accuracy (In % ± Standard Error) Using C-SVM.

Datasets	**DQMK**	DBMK	WLSK	SPGK	GCGK	UQJS	JSGK
BAR31	67.93 ± .57	**69.40** ± .56	58.53 ± .53	55.73 ± .44	23.40 ± .60	30.80 ± .61	24.10 ± .86
BSPHERE31	**58.16** ± .67	56.43 ± .69	42.10 ± .68	48.20 ± .76	18.80 ± .50	24.80 ± .61	21.76 ± .53
GEOD31	41.76 ± .47	**42.83** ± .50	38.20 ± .68	38.40 ± .65	22.36 ± .55	23.73 ± .66	18.93 ± .50
Shock	**47.00** ± .55	30.80 ± .93	36.40 ± .99	37.88 ± .93	27.06 ± .99	40.60 ± .92	21.73 ± .76

Table 2. Runtime of Computing the Kernel Matrix.

Datasets	**DQMK**	DBMK	WLSK	SPGK	GCGK	UQJS	JSGK
BAR31	198"	682"	30"	11"	1"	630"	1"
BSPHERE31	171"	720"	25"	14"	1"	828"	1"
GEOD31	97"	649"	15"	11"	1"	519"	1"
Shock	8"	7"	3"	1"	1"	14"	1"

for the effectiveness are threefold. First, the DQMK kernel can be seen as a vertex matching kernel through discrete-time quantum walks on directed line graphs transformed from original graphs. By contrast, other kernels are defined on original graphs. As we have stated in Section 2.1, the directed line graph transformed from an original graph can reflect richer characteristics than the original graph. As a result, the DQMK kernel defined on line graphs can encapsulate more information than other kernels defined on original graphs. Second, compared to the WLSK, SPGK and GCGK kernels that require decomposing graphs into substructures, our DQMK kernel can establish the substructure location correspondences in directed line graphs transformed from original graphs. By contrast, the WLSK, SPGK and GCGK kernels do not consider the location correspondence between pairwise substructures in the original graphs. Third, compared to the JSGK and UQJS kernels that rely on the similarity measure between original global graphs in terms of the classical or quantum JSD, our DQMK kernel can identify the correspondence information between both the vertices and the substructures of directed line graphs, and can thus reflect richer interior topological characteristics. By contrast, the JSGK and QJSK kernels only reflect the global graph similarity information of original graphs.

Finally, our DQMK kernel outperforms the DBMK kernel on the Shock and BSPHERE31 datasets. Especially, the classification accuracy of our new DQMK kernel on the Shock dataset is much better than that of the DBMK kernel. On the GEOD31 and BAR31 datasets, the classification accuracies of our DQMK kernel are a little lower than those of the DBMK kernel. Overall, our DQMK kernel outperforms or is competitive to the DBMK kernel. The effectiveness for this is that the DQMK kernel defined through discrete-time quantum walks on directed line graphs can reflect richer characteristics and quantum information than the DBMK kernel defined on original graphs.

5   Conclusion

In this paper, we develop a new edge-based matching kernel for graphs through discrete-time quantum walks on directed line graphs. Experiments demonstrate the effectiveness of the new kernel.

References

1. Bai, L., Rossi, L., Bunke, H., Hancock, E.R.: Attributed graph kernels using the jensen-tsallis q-differences. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014, Part I. LNCS, vol. 8724, pp. 99–114. Springer, Heidelberg (2014)
2. Haussler, D.: Convolution kernels on discrete structures. In: Technical Report UCS-CRL-99-10, Santa Cruz, CA, USA (1999)
3. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernels between labeled graphs. In: Proceedings of ICML, pp. 321–328 (2003)
4. Borgwardt, K.M., Kriegel, H.P.: Shortest-path kernels on graphs. In: Proceedings of the IEEE International Conference on Data Mining, pp. 74–81 (2005)
5. Aziz, F., Wilson, R.C., Hancock, E.R.: Backtrackless walks on a graph. IEEE Transactions on Neural Networks and Learning Systems **24**, 977–989 (2013)
6. Ren, P., Wilson, R.C., Hancock, E.R.: Graph characterization via ihara coefficients. IEEE Transactions on Neural Networks **22**, 233–245 (2011)
7. Shervashidze, N., Schweitzer, P., van Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-lehman graph kernels. Journal of Machine Learning Research **12**, 2539–2561 (2011)
8. Harchaoui, Z., Bach, F.: Image classification with segmentation graph kernels. In: Proceedings of CVPR (2007)
9. Bach, F.R.: Graph kernels between point clouds. In: Proceedings of ICML, pp. 25–32 (2008)
10. Bai, L., Ren, P., Hancock, E.R.: A hypergraph kernel from isomorphism tests. In: Proceedings of ICPR, pp. 3880–3885 (2014)
11. Bai, L.: Information Theoretic Graph Kernels. University of York, UK (2014)
12. Bai, L., Ren, P., Bai, X., Hancock, E.R.: A graph kernel from the depth-based representation. In: Fränti, P., Brown, G., Loog, M., Escolano, F., Pelillo, M. (eds.) S+SSPR 2014. LNCS, vol. 8621, pp. 1–11. Springer, Heidelberg (2014)
13. Emms, D., Severini, S., Wilson, R.C., Hancock, E.R.: Coined quantum walks lift the cospectrality of graphs and trees. Pattern Recognition **42**, 1988–2002 (2009)
14. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of ACM Symposium on the Theory of Computation, pp. 212–219 (1996)
15. Ren, P., Aleksic, T., Emms, D., Wilson, R.C., Hancock, E.R.: Quantum walks, ihara zeta functions and cospectrality in regular graphs. Quantum Information Process **10**, 405–417 (2011)
16. Escolano, F., Hancock, E., Lozano, M.: Heat diffusion: Thermodynamic depth complexity of networks. Physical Review E **85**, 206236 (2012)
17. Munkres, J.: Algorithms for the assignment and transportation problems. Journal of the Society for Industrial and Applied Mathematics **5** (1957)
18. Bai, L., Hancock, E.R.: Depth-based complexity traces of graphs. Pattern Recognition **47**, 1172–1186 (2014)
19. Biasotti, S., Marini, S., Mortara, M., Patané, G., Spagnuolo, M., Falcidieno, B.: 3D shape matching through topological structures. In: Nyström, I., Sanniti di Baja, G., Svensson, S. (eds.) DGCI 2003. LNCS, vol. 2886, pp. 194–203. Springer, Heidelberg (2003)

20. Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., Borgwardt, K.: Efficient graphlet kernels for large graph comparison. Journal of Machine Learning Research **5**, 488–495 (2009)
21. Bai, L., Rossi, L., Torsello, A., Hancock, E.R.: A quantum jensen-shannon graph kernel for unattributed graphs. Pattern Recognition **48**, 344–355 (2015)
22. Bai, L., Hancock, E.R.: Graph kernels from the jensen-shannon divergence. Journal of Mathematical Imaging and Vision **47**, 60–69 (2013)
23. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. Software (2011). http://www.csie.ntu.edu.tw/cjlin/libsvm