An Aligned Subgraph Kernel Based on Discrete-Time Quantum Walk

Kai Liu KAILIU@NUDT.EDU.CN
College of Computer, National University of Defense Technology, Changsha, 410073, China.

Lulu Wang* WANGLULUNUDT@163.COM
National Innovation Institute of Defense Technology, Beijing 100073, China.

Yi Zhang ZHANGYI@ZXWL-CFW.CN
Central Future Works Technology Co.,Ltd.,Beijing, China.

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

In this paper, a novel graph kernel is designed by aligning the amplitude representation of the vertices. Firstly, the amplitude representation of a vertex is calculated based on the discrete-time quantum walk. Then a matching-based graph kernel is constructed through identifying the correspondence between the vertices of two graphs. The newly proposed kernel can be regarded as a kind of aligned subgraph kernel that incorporates the explicit local information of substructures. Thus, it can address the disadvantage arising in the classical R-convolution kernel that the relative locations of substructures between the graphs are ignored. Experiments on several standard datasets demonstrate that the proposed kernel has better performance compared with other state-of-the-art graph kernels in terms of classification accuracy.

Keywords: Graph kernels, R-convolution kernels, Structured data, Similarity measures for graphs.

1. Introduction

Graph is a kind of important structure for information representation, since it can naturally reflect the structural and relational arrangements of entities. In many scientific fields such as social network Fan (2012), bioinformatics research Raymond and Willett (2002) and image matching Ta et al. (2009), researchers are often interested in computing the semantically meaningful similarities between these structured objects. For instance, two protein molecules with the same chemical properties usually have similar topological structures Mah and Vert (2009). Thus people can predict the property of an unknown molecule by means of topology comparison with the known ones.

One of the popular approaches to measure the similarity of graphs is to use the kernel method. Graph kernel has been proved to be powerful for structural analyses in machine learning Vishwanathan et al. (2010). There have been a lot of successful attempts Gärtner et al. (2003); Harchaoui and Bach (2007); Jebara et al. (2004) to cluster or classify graphs with the help of graph kernels. Roughly speaking, kernel methods provide an implicit embedding of graphs in a high dimensional space which can effectively preserve the structural information. They can measure the similarity between the graphs with a kernel function corresponding to an inner product in reproducing kernel Hilbert space(RKHS) Schölkopf and Smola (2002).

Most of the existing graph kernels belong to the family of R-convolution kernels proposed by Haussler Haussler (1999). This is a general framework to handle structural objects by the way of comparing all pairs of isomorphic substructures under different decomposition methods. The key idea of the R-convolution kernel is to recursively decompose structured objects into "atomic" substructures and define valid measurements between them. Given two graphs G and G', the kernel value of them is computed by

$$K\left(G, G'\right) = \left\langle \phi\left(G\right), \phi\left(G'\right)\right\rangle_H, \tag{1}$$

where $\phi\left(G\right)$ denotes vector representation of the graph G and $\langle\cdot,\cdot\rangle_H$ represents the dot product in reproducing kernel Hilbert space.

Generally speaking, the recently developed R-convolutional graph kernels can be divided into the following categories: the graph kernels based on comparing all pairs of a) subgraphs Horváth et al. (2004); Shervashidze et al. (2009), b) walks Kashima et al. (2003); Gärtner et al. (2003), c) paths Borgwardt and Kriegel (2005), and d) subtree patterns Ramon and Gärtner (2003); Mahé and Vert (2009). The first class, i.e., graph kernels based on subgraphs, mainly includes graphlet kernels which count all types of substructures with the node size $k \in \{3, 4, 5\}$. To improve the computation efficiency, optimization methods are proposed based on sampling or exploitation of the low maximum degree of subgraphs Shervashidze et al. (2009). The second class, i.e., graph kernels based on walks, indicates random walk kernel Gärtner et al. (2003) which counts the number of matching pairs of random walks in two graphs. There also exist efficient computation schemes which use dynamic programming-based approach to speed up this kind of kernel Harchaoui and Bach (2007). Paths based kernels are the third class, which mainly include the shortest path kernel proposed by Borgwardt and Kriegel Borgwardt and Kriegel (2005) where the pairs of the shortest paths having the same length are counted. Weisfeiler-Lehman kernel Shervashidze et al. (2011) is the representation of the subtree pattern based graph kernels. The key idea of this kernel is to compare the number of shared subtrees between the graphs using the procedures inspired by the Weisfeiler-Lehman test of graph isomorphism. The kernel recently proposed by Lu Bai Bai et al. (2015b,c) can also be classified into this category, where an aligned subtree kernel based on the calculation of entropy is implemented.

Although the R-convolution kernels have achieved great developments, the drawback of neglecting the relative locations of the substructures still exists. This is because the original R-convolution kernels lack an effective way to establish reliable correspondences between substructures. As a result, this shortcoming limits the precision of similarity measurements for graphs.

To overcome this drawback, we propose a novel matching kernel by aligning the amplitude representation of the vertices. In our method, the amplitude representation of the graph is calculated based on the simulation of the discrete-time quantum walk (DTQW). Because the amplitude representation can reflect rich characteristics of the K-layer expansion subgraphs centered on a vertex and the DTQW is a powerful tool to illustrate the topological information, their combination will provide us an elegant way of designing new effective graph kernels.

For each pair of graphs under comparison, the K-layer amplitude representation of each vertex is computed. More precisely, we characterize the degree distributions of the K-layer subgraph with the help of the DTQW and store this information in the amplitude representation for each vertex. Then we can find the matched vertex from the two graphs by aligning the amplitude representation. Therefore, a new vertex matching kernel is developed by counting the number of matched vertex pairs. Our kernel can be regarded as a kind of special aligned subgraph kernel, and thus we could explain the reasons for the effectiveness of our kernel after theoretical analyses. Through counting the aligned subgraph, the local information of the graph is embedded into the graph kernel. Therefore our kernel can overcome the aforementioned weakness of ignoring the correspondences between the substructures. The experiments on several standard datasets demonstrate that our kernel has a better performance than the state-of-the-art graph kernels as for graph classification accuracy.

This paper is organized as follows. In the next section, the theory of DTQW is introduced and the method of vertex matching is designed. Then in Section 3, the new graph kernel is proposed and the relationship between the proposed kernel and the classical graph kernels are discussed. In Section 4, we compare our kernel with the state-of-the-art graph kernels in terms of graph classification task. The results of classification accuracy on several graph benchmark datasets are reported. Finally, a summary in Section 5 concludes this paper.

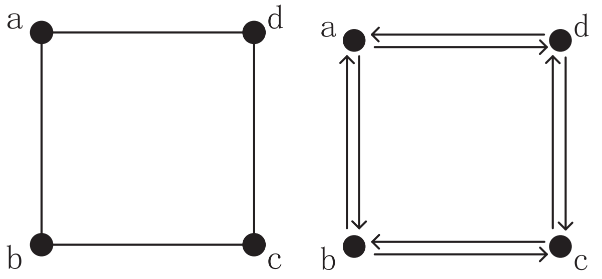
2. Vertex Matching Using Amplitude Representation

In this section, we first review the concept of the DTQW on a graph. Then the amplitude representation for a vertex is defined based on this model. Finally we present the vertex matching method by aligning the amplitude representation.

2.1. Discrete-Time Quantum Walks

The DTQW Kempe (2003) represents a quantum counterpart of the classical random walk. There are two reasons that the DTQW is a good way to explore information on a graph. On the one hand, unlike the classical random walk, the DTQW uses qubits rather than bits as the basic representational unit. By virtue of the natural parallelism of quantum computing, DTQW can capture abundant topological information in a graph structure. On the other hand, because quantum processes are reversible and non-ergodic, the DTQW could diminish the tottering problem arising in the classical random walk.

For an undirected graph $G(V, E)$ with vertex set V and edge set E, we refer to (u, v) as an edge of G, where $u, v \in V$. In order to store graph information in the quantum superposition, we need to first transfer the original graph into its associated symmetric digraph $G_d(V_d, E_d)$ through replacing each edge $(u, v) \in E$ with a pair of directed edges $\langle u, v\rangle \in E_d$ and $\langle v, u\rangle \in E_d$. An example of transforming the original graph to a symmetric digraph is shown in Fig. 1. Then we represent the states being on the edge $\langle u, v\rangle \in E_d$ and $\langle v, u\rangle \in E_d$ as $|uv\rangle$ and $|vu\rangle$ respectively. Consequently, the state of the DTQW on a graph G can be written as follows:



(a) Original graph. (b) Symmetric digraph.

Figure 1: Symmetric digraph construction.

$$|\varphi\rangle = \sum_{\langle u,v\rangle \in E_d} \alpha_{uv} \, |uv\rangle, \;\; \alpha_{uv} \in \mathbb{C} \tag{2}$$

where α_{uv} refers to the quantum amplitude. The probability that the state $|\varphi\rangle$ being in a particular basis state $|uv\rangle$ is given by $\Pr(|uv\rangle) = \alpha_{uv}\alpha_{uv}^{*}$ where α_{uv}^{*} is the complex conjugate of α_{uv}.

Using the common Grover diffusion matrix as the transition matrix Emms et al. (2009), every state can be transferred according to the following equation:

$$|uv\rangle \to \left(\frac{2}{d(v)} - 1\right)|vu\rangle + \frac{2}{d(v)} \sum_{\substack{\forall k \in V, (v,k) \in E \\ \& k \neq u}} |vk\rangle \tag{3}$$

where $d(v)$ represents the degree of vertex v. Given a basic state $|uv\rangle$, the Grover diffusion matrix assigns the same amplitude to all the state transitions $|uv\rangle \to |vk\rangle$ where k is the neighbors of v, and a different amplitude to the transition $|uv\rangle \to |vu\rangle$. Therefore the matrix that governs the evolution of the superposition state is

$$U_{(u,v),(w,x)} = \begin{cases} \frac{2}{d(x)} - \delta_{ux} & if \;\; v = w \\ 0 & otherwise \end{cases} \tag{4}$$

where δ_{ux} is the Kronecker delta function. If we use $\left|\varphi^{T}\right\rangle$ to represent the quantum superposition state at the time T, the evolution of the DTQW can be represented by $\left|\varphi^{T+1}\right\rangle = U\left|\varphi^{T}\right\rangle$. Given the initial superposition state, the degree of each node in the graph is one of the important factors that affects the implementation of the DTQW. Therefore the process of quantum walk on a graph, to some extent, reflects the characteristics of the degree distribution of the graph.

2.2. The Amplitude Feature for Graphs

In this subsection, we introduce an amplitude characteristic of a graph based on the DTQW. This is a new way to characterize the degree distribution of a graph.

For an undirected graph $G(V, E)$ and a vertex $v \in V$, we first add a new adjacent node to vertex v. This auxiliary vertex is called c. The updated graph can be denoted as $\mathrm{G}'(V',E')$ where $V' = V \cup \{c\}$ and $E' = E \cup \{(v,c)\}$. The aforementioned method is used to transform the undirected graph G' into the symmetric digraph G'_d. Then a T-step simulation of the DTQW is executed on the digraph G'_d and the amplitude of the directed edge $\langle v,c\rangle$ can be obtained. In the DTQW process, due to the interference effect of quantum computation, this amplitude is a result of the interactions between the basic states. Therefore the amplitude of $\langle v,c\rangle$ can be considered as a characteristic of the graph from the perspective of node v which is recorded as $Am_v^T(G)$.

As shown in (3), in a connected graph, the effect of the amplitude of the directed edge $\langle u_1,v_1\rangle$ can be transmitted to the amplitude of the directed edge $\langle u_2,v_2\rangle$ after T_1 steps where T_1 is the shortest path between u_1 and u_2. Generally speaking, we specify the number of steps in the DTQW according to the diameter of the graph.

2.3. The Amplitude Representation through DTQW

In this subsection, we design an amplitude representation of a vertex in the graph. This is a special method to compute the depth-based representation around a vertex. The DTQW is used to characterize the depth complexity information of each vertex. Accordingly, the local information is represented by a vectorial signature and the corresponding vertex is projected into a high dimensional space.

For an undirected graph $G(V,E)$ and the vertex $v \in V$, we defined a vertex set S_v^k as $S_v^k = \{u \in V | P_G(u,v) \le k\}$ where $P_G(u,v)$ indicates the shortest path between the vertex u and v in the graph G. Then the K-layer expansion subgraph $G_v^K\left(\upsilon_v^K, \varepsilon_v^K\right)$ around v is determined.

$$\begin{aligned} \upsilon_v^K &= S_v^k \\ \varepsilon_v^K &= \left\{(u',v')\,|u',v' \in S_v^k \& (u',v') \in E\right\} \end{aligned} \tag{5}$$

By calculating the amplitude features of different expansion subgraphs, the h-layer amplitude representation $DB_G^h(v)$ around v is obtained.

$$DB_G^h(v) = \left\{Am_v^1\left(G_v^1\right), Am_v^2\left(G_v^2\right), \cdots Am_v^h\left(G_v^h\right)\right\} \tag{6}$$

where $Am_v^k\left(G_v^k\right)$ represents the amplitude feature of the k-layer expansion subgraph G_v^k. In order to capture the comprehensive topological information of G_v^k, the number of steps for quantum walks simulation is set to be k which is not less than the maximum length of the shortest path from v to each vertices in G_v^k.

2.4. The Vertex Matching Based on The Amplitude Representation

In this subsection, we propose a novel vertex matching method for a pair of graphs by aligning their amplitude representations. We commence by computing the m-layer amplitude representation of each vertex as the point coordinate. For a pair of graphs $G_p(V_p,E_p)$ and $G_q(V_q,E_q)$, we use the m-layer vectorial representation $DB_{G_p}^m(u_i)$ and $DB_{G_q}^m(v_j)$ as the point coordinates of the vertices $u_i \in V_p$ and $v_j \in V_q$ respectively. Then we obtain the distance between u_i and v_j by computing the absolute difference value between $DB_{G_p}^m(u_i)$ and $DB_{G_q}^m(v_j)$.

$$R\left(i,j\right)=\left|DB_{G_p}^{m}\left(u_i\right)-DB_{G_q}^{m}\left(v_j\right)\right| \tag{7}$$

where $R(i,j)$ indicates the distance or dissimilarity between the vertices $u_i \in V_p$ and $v_j \in V_q$. The matrix R is a distance matrix with size $|V_p| \times |V_q|$ in which the row indexes are the vertices from the graph G_p and the column indexes are the vertices from G_q.

For unlabeled graph comparison, there is a correspondence between u_i and v_j if $R(i,j)$ equals to 0. And we define a correspondence matrix $C^m \in \{0,1\}^{|V_p|\times|V_q|}$ to record the results of these correspondences about all pairs of vertices.

$$C^m\left(i,j\right)=\begin{cases}1 & if\, R\left(i,j\right)=0\\ 0 & otherwise\end{cases} \tag{8}$$

For labeled graph, the correspondences between the vertex pairs are defined more strictly which should take the labels into consideration. We define a function L which can map all the identical label set of different graphs into the same number. For instance, $L(G)$ can be defined as the summation of labels from graph G when the labels are indicated by numerical values. Therefore the correspondence matrix $C^m \in \{0,1\}^{|V_p|\times|V_q|}$ for labeled graph pair G_p and G_q is written as follows,

$$C^m\left(i,j\right)=\begin{cases}1 & if\, R\left(i,j\right)=0\, and\, L\left(\left(G_p\right)_{u_i}^{m}\right)=L\left(\left(G_q\right)_{v_j}^{m}\right)\\ 0 & otherwise\end{cases} \tag{9}$$

where $\left(G_p\right)_{u_i}^{m}$ and $\left(G_q\right)_{v_j}^{m}$ indicate the m-layer expansion subgraph routed at u_i and v_j respectively. Therefore for both labeled and unlabeled graphs, we could calculate the correspondences between their vertices and obtain a correspondence matrix.

If $C^m\left(i,j\right)=1$, the corresponding subgraphs $\left(G_p\right)_{u_i}^{m}$ and $\left(G_q\right)_{v_j}^{m}$ are supposed to be approximately isomorphic and the vertices u_i and v_j are matched. For a pair of graphs, note that, a vertex from one graph may have multiple matched vertices from another graph. In our method, only one matching vertex is allowed for a vertex at most. In order to find as many vertex correspondences as possible, Hungarian algorithm Munkres (1957) is adopted to update the correspondence matrix $C^m\left(i,j\right)$. Here, the matrix $C^m\left(i,j\right)$ can be regarded as an incidence matrix of a bipartite graph and as the input of the Hungarian algorithm. By performing this algorithm, each vertex in the graph is assigned with only one matched vertex from the other graph which is chosen to be the most matched vertex. And the correspondence matrix is updated to $H^m \in \{0,1\}^{|V_p|\times|V_q|}$ which is defined as

$$H^m = Hungarian(C^m) \tag{10}$$

3. Graph Kernel Based on Amplitude Matching

In this section, a novel graph kernel is proposed based on vertex matching by aligning the amplitude representation of the vertices. And the relationship between the proposed kernel and the all subgraph (AS) kernel is discussed.

3.1. The Alignment Matching Kernel

For two graphs $G_p\left(V_p, E_p\right)$ and $G_q\left(V_q, E_q\right)$, the reasonable correspondence matrix H^m is calculated first based on the definition of vertex matching in the previous section.

The amplitude matching kernel $K_{Am}^{(M)}\left(G_p, G_q\right)$ is defined as

$$K_{Am}^{(M)}\left(G_p, G_q\right) = \sum_{r=1}^{M} \sum_{i=1}^{|V_p|} \sum_{j=1}^{|V_q|} H^r\left(i, j\right) \tag{11}$$

Eq. (11) indicates that $K_{Am}^{(M)}\left(G_p, G_q\right)$ counts the number of matched vertex pairs between the two graphs G_p and G_q over several correspondence matrices.

Lemma 1 *The proposed kernel $K_{Am}^{(M)}\left(G_p, G_q\right)$ is positive definite.*

Proof From (11), the amplitude matching kernel can be re-written as

$$K_{Am}^{(M)}\left(G_p, G_q\right) = \sum_{r=1}^{M} K^r\left(G_p, G_q\right) \tag{12}$$

where K^r is the base counting kernel that counts pairs of matched vertices between the two graphs G_p and G_q based on the correspondence matrix $H^r\left(i, j\right)$.

$$K^r\left(G_p, G_q\right) = \sum_{u_i \in V_p} \sum_{v_j \in V_q} \delta\left(u_i, v_j\right) \tag{13}$$

$$\delta\left(u_i, v_j\right) = \left\{ \begin{array}{ll} 1 & if \ H^r\left(i, j\right) = 1 \\ 0 & otherwise \end{array} \right. \tag{14}$$

Because δ is the Dirac kernel that is positive definite, the base counting kernel K^r which is the summation of the Dirac kernel is thus positive definite. Therefore, the amplitude matching kernel, which is the summation of the positive definite kernels K^r is also positive definite. ■

3.2. Relationship to Classical Graph Kernels

The amplitude matching kernel can be regarded as an R-convolution graph kernel. In this subsection, the proposed kernel is compared with the classical AS kernel and the effectiveness of the proposed kernel is demonstrated.

For a graph pair $G_p\left(V_p, E_p\right)$ and $G_q\left(V_q, E_q\right)$, the AS kernel is shown as follows:

$$K_{AS}\left(G_p, G_q\right) = \sum_{S_p \subseteq G_p} \sum_{S_q \subseteq G_q} k_{iso}\left(S_p, S_q\right) \tag{15}$$

$$k_{iso}\left(S_p, S_q\right) = \left\{ \begin{array}{ll} 1 & if \ S_p \simeq S_q \\ 0 & otherwise \end{array} \right. \tag{16}$$

where $S_p \simeq S_q$ indicates that the two subgraphs S_p and S_p extracted from G_p and G_q respectively are isomorphic. Therefore, the all subgraph kernel is obtained by computing the number of isomorphic subgraph pairs between the graph G_p and G_q.

The amplitude matching kernel proposed in this paper is similar to the all subgraph kernel. According to the definition in Section 2.3, the amplitude representation of a vertex can be regarded as a vectorial signature of the local information, which is determined by the k-layer expansion subgraph around this vertex. Thus, when two vertices are matched in the computation of the amplitude matching kernel, the k-layer expansion subgraphs around the corresponding vertices are supposed to be approximate isomorphism. From this point of view, the amplitude matching kernel is another expression of all subgraph kernel on sample graphs.

$$K_{Am}^{(M)}\left(G_p, G_q\right) = \sum_{S_p \subseteq G_p} \sum_{S_q \subseteq G_q} k_I\left(S_p, S_q\right) \tag{17}$$

where

$$k_I\left(S_p, S_q\right) = \begin{cases} 1 & \begin{array}{c} if\ S_p = DB_{G_p}^{h}(u)\ and\ S_p = DB_{G_q}^{h}(v), \\ u\ and\ v\ are\ matched \end{array} \\ 0 & \quad\quad\quad otherwise \end{cases} \tag{18}$$

For the kernel K_{Am} and K_{AS}, it is observed that both of them have the similar mechanism, which is to count the number of isomorphic subgraph pairs.

However, there are obvious differences between the two kernels. On one hand, the amplitude matching kernel avoids the NP-hard problem of identifying all pairs of isomorphic subgraphs appeared in the K_{AS} kernel. Because our method only evaluate the expansion subgraph around the vertex, the computation efficiency of the K_{Am} kernel is faster than the K_{AS} kernel. On the other hand, the expansion subgraph defined in Section 2.2 includes the local information around the corresponding vertex. Only the subgraphs around a pair of matched vertices are evaluated with regard to be isomorphic. Thus, there is a local correspondence between the isomorphic subgraphs in the K_{Am} kernel. On the contrary, the pair of isomorphic subgraphs having no local correspondence may also be considered in the computation of the K_{AS} kernel. Therefore our kernel can overcome the shortcoming of ignoring the local correspondence between substructures appearing in the R-convolution kernel.

4. Experimental Results

In this section, experiments on several graph datasets are done to demonstrate the performance of the proposed amplitude matching kernel. Some bioinformatic datasets used in the experiments are introduced first as well as the basic information of the experimental setup. Then the proposed kernel is compared with the other state-of-the-art graph kernels under the task of graph classification.

Table 1: Key information of the four bioinformatic datasets used in this paper.

Dataset	Size	Classes	Avg.Nodes	Avg.Edges	Node.Label	Edge.Label
MUTAG	188	2	17.93	39.5	√	√
PTC	344	2	14.29	14.69	√	√
BZR	405	2	35.75	38.36	√	-
ENZYNES	600	6	32.63	62.14	√	-

4.1. Data Sets

In our experiments, four benchmark graph kernel datasets, namely, MUTAG, PTC, BZR and ENZYMES are applied. All of them can be downloaded from the benchmark website Kersting et al. (2016). Table. 1 shows the detailed information of these datasets.

MUTAG. The MUTAG dataset Debnath et al. (1991) is based on weighted graphs representation of 188 chemical compounds. It can be used to predict whether each compound has mutagenicity. The minimum, maximum and average number of nodes are 10, 28, and 17.93 respectively.

PTC. The PTC (Predictive Toxicology Challenge) Li et al. (2012) is a dataset of many chemical compounds that reports the carcinogenicity for Male Rats(MR), Male Mice(MM), Female Rats(FR) and Female Mice(FM). We select 344 MR weighted graphs for evaluation.

BZR. The BZR dataset Sutherland et al. (2003) includes 405 graphs representing ligands of the benzodiazepine receptor. These graphs can be divided into two classes according to the activity of the compounds. The average number of vertices and edges are 35.75 and 38.36, respectively.

ENZYMES. The ENZYMES is a dataset consisting of 600 protein tertiary structures obtained from enzyme database Schomburg et al. (2004). It assigns each graph to one of the 6 classes according to the function of the corresponding compounds.

4.2. Experimental Setup

The graph classification performance of using the proposed amplitude matching kernel is evaluated based on the aforementioned datasets. And the proposed graph kernel is also compared with the other state-of-the-art graph kernels. Because there are a large number of graph kernels proposed in the literature, we can not compare to all of them. Some famous and representative graph kernels are chosen, which include 1) the Graphlet kernel Shervashidze et al. (2009) with weighted graphlets of 3, 2) the shortest path kernel (SPGK) with labeled path Borgwardt and Kriegel (2005), 3) the Weisfeiler-Lehman subtree graph kernel(WLGK) using the node labels Shervashidze et al. (2011), 4) the labeled random walk kernel Gärtner et al. (2003), 5) the quantum Jensen-Shannon graph kernel (QJSK) Bai et al. (2015a), 6) the Lovasz theta graph kernel(LTGK) Johansson et al. (2014), and 7) the GraphHopper kernel Feragen et al. (2013).

For the proposed amplitude matching kernel, M is set to be 5. For the WLGK and QJSK, the highest height of subtrees in the Weisfeiler-Lehman isomorphism for WLGK and the tree-index method for QJSK are both set to be 5.

Table 2: Comparison of classification accuracy (In% ± Standard Error) on bioinformatics datasets.

Datasets	MUTAG	PTC	BZR	ENZYMES
Graphlet	80.50±1.21	57.86±1.64	84.23±0.34	38.93±1.00
SPGK	86.83±1.62	59.85±2.64	85.05±0.51	41.38±2.01
WLGK	83.72±0.98	62.79±2.15	86.88±0.75	53.07±1.12
RWGK	84.78±1.42	58.24±1.93	79.18±0.29	20.32±1.58
QJSU	81.61±1.90	55.74±0.21	78.73±0.18	10.52±0.90
LTGK	82.06±1.57	57.15±1.80	79.08±0.72	23.12±0.68
GraphHopper	82.50±1.86	56.35±1.26	85.15±0.99	36.18±2.00
AMGK	**87.39±1.41**	**62.88±2.28**	**87.85±0.57**	**55.23±0.92**

All these kernel methods are coded in Matlab 2013a and performed on a laptop with Intel Core-i5 CPU at 2.30 GHz. In order to evaluate the performance of each kernel matrix, we perform 10-fold cross-validation with a binary C-SVM to compute the classification accuracy using LIBSVM library Chang and Lin (2011). In each experiment, nine-tenth of the samples are used to train the parameters and one-tenth are used for testing. To exclude the random effects, we repeat the whole experiment 10 times and report the average prediction accuracy (± standard error) for comparison.

4.3. Results

The results are shown in Table. 2. It can be concluded that the proposed amplitude matching kernel (AMGK) outperforms the other convolution-based counterparts on all the datasets in terms of classification accuracy. And the improvements on different datasets are various.

There are several reasons for the effectiveness of the AMGK method. Firstly, unlike those methods WLGK, SPGK and Graphlet which simply decompose the graphs into substructures, the proposed AMGK can incorporate the relative location information between the substructures in the kernel computation which is neglected in the above three kernels. Secondly, compared with LTGK and GraphHopper kernels that measure the global similarity information between the graphs using different method, AMGK can capture finer-grained local structural features of the graphs using the DTQW and recognize the correspondence information between both the substructures and the vertices. Thirdly, compared with WL-GK and Graphlet which utilize the isomorphism detection method to design graph kernel, our kernel can find the aligned isomorphic substructures via matching the amplitude features of the subgraphs.

5. Conclusion

In this paper, an amplitude matching graph kernel is proposed by aligning the amplitude features of the substructures based on the DTQW. The proposed kernel can capture explicit local substructure correspondence which overcomes the drawback of ignoring the relative location information between the substructures that exists in the classical R-convolution graph kernels. The experiments show that our kernel has a better performance compared with other state-of-the-art graph kernels in terms of graph classification task.

Acknowledgments

The authors appreciate the kind comments and professional criticisms of the anonymous reviewers.

References

Lu Bai, Luca Rossi, Andrea Torsello, and Edwin R Hancock. A quantum jensen–shannon graph kernel for unattributed graphs. *Pattern Recognition*, 48(2):344–355, 2015a.

Lu Bai, Luca Rossi, Zhihong Zhang, and Edwin Hancock. An aligned subtree kernel for weighted graphs. In *International Conference on Machine Learning*, pages 30–39, 2015b.

Lu Bai, Zhihong Zhang, Chaoyan Wang, Xiao Bai, and Edwin Hancock. A graph kernel based on the jensen-shannon representation alignment. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015c.

Karsten M Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Fifth IEEE international conference on data mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 34(2):786–797, 1991.

David Emms, Richard C Wilson, and Edwin R Hancock. Graph matching using the interference of discrete-time quantum walks. *Image and Vision Computing*, 27(7):934–949, 2009.

Wenfei Fan. Graph pattern matching revised for social network analysis. In *International Conference on Database Theory*, 2012.

Aasa Feragen, Niklas Kasenburg, Jens Petersen, Marleen de Bruijne, and Karsten Borgwardt. Scalable kernels for graphs with continuous attributes. In *Advances in Neural Information Processing Systems*, pages 216–224, 2013.

Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Learning theory and kernel machines*, pages 129–143. Springer, 2003.

Zaïd Harchaoui and Francis Bach. Image classification with segmentation graph kernels. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

David Haussler. Convolution kernels on discrete structures. Technical report, Technical report, Department of Computer Science, University of California , 1999.

Tamás Horváth, Thomas Gärtner, and Stefan Wrobel. Cyclic pattern kernels for predictive graph mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 158–167. ACM, 2004.

Tony Jebara, Risi Kondor, and Andrew Howard. Probability product kernels. *Journal of Machine Learning Research*, 5(Jul):819–844, 2004.

Fredrik Johansson, Vinay Jethava, Devdatt Dubhashi, and Chiranjib Bhattacharyya. Global graph kernels using geometric embeddings. In *Proceedings of the 31st International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, 2014.

Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 321–328, 2003.

Julia Kempe. Quantum random walks: an introductory overview. *Contemporary Physics*, 44(4):307–327, 2003.

Kristian Kersting, Nils M. Kriege, Christopher Morris, Petra Mutzel, and Marion Neumann. Benchmark data sets for graph kernels, 2016. URL http://graphkernels.cs.tu-dortmund.de.

Geng Li, Murat Semerci, Bülent Yener, and Mohammed J Zaki. Effective graph classification based on topological and label attributes. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 5(4):265–283, 2012.

Pierre Mah and Jean Philippe Vert. Graph kernels based on tree patterns for molecules. *Machine Learning*, 75(1):3–35, 2009.

Pierre Mahé and Jean-Philippe Vert. Graph kernels based on tree patterns for molecules. *Machine learning*, 75(1):3–35, 2009.

James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

Jan Ramon and Thomas Gärtner. Expressivity versus efficiency of graph kernels. In *First international workshop on mining graphs, trees and sequences*, pages 65–74. Citeseer, 2003.

John W. Raymond and Peter Willett. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *J Comput Aided Mol Des*, 16(7):521–533, 2002.

Bernhard Schölkopf and Alexander J Smola. Learning with kernels. 2002, 2002.

Ida Schomburg, Antje Chang, Christian Ebeling, Marion Gremse, Christian Heldt, Gregor Huhn, and Dietmar Schomburg. Brenda, the enzyme database: updates and major new developments. *Nucleic acids research*, 32(suppl_1):D431–D433, 2004.

Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In *Artificial Intelligence and Statistics*, pages 488–495, 2009.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(Sep):2539–2561, 2011.

Jeffrey J Sutherland, Lee A O'Brien, and Donald F Weaver. Spline-fitting with a genetic algorithm: A method for developing classification structure- activity relationships. *Journal of chemical information and computer sciences*, 43(6):1906–1915, 2003.

Vinh Thong Ta, Olivier Lzoray, Abderrahim Elmoataz, and Sophie Schpp. Graph-based tools for microscopic cellular image segmentation. *Pattern Recognition*, 42(6):1113–1125, 2009.

S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11(Apr):1201–1242, 2010.