Proceedings Paper:
Bai, L., Rossi, L., Cui, L. et al. (1 more author) (2016) A transitive aligned Weisfeiler-Lehman subtree kernel. In: Davis, L., Bimbo, A. Del and Lovell, B., (eds.) 2016 23rd International Conference on Pattern Recognition (ICPR). IEEE Computer Society , Los Alamitos, CA, USA , pp. 396-401.

https://doi.org/10.1109/ICPR.2016.7899666

A Transitive Aligned Weisfeiler-Lehman Subtree Kernel

Lu Bai*, Luca Rossi†, Lixin Cui*, and Edwin R. Hancock‡
*School of Information, Central University of Finance and Economics, 39 South College Road, Beijing, China
†School of Engineering and Applied Sciences, Aston University, Birmingham, B4 7ET, UK
‡Department of Computer Science, University of York, York, YO10 5DD, UK

***Abstract*—In this paper, we develop a new transitive aligned Weisfeiler-Lehman subtree kernel. This kernel not only overcomes the shortcoming of ignoring correspondence information between isomorphic substructures that arises in existing R-convolution kernels, but also guarantees the transitivity between the correspondence information that is not available for existing matching kernels. Our kernel outperforms state-of-the-art graph kernels in terms of classification accuracy on standard graph datasets.**

I. Introduction

Graph kernels are powerful tools for structural analysis on graphs [1]. The main advantage of using graph kernels is that they characterize graphs in a high dimensional space and thus better preserve graph structures. Most of the recently introduced graph kernels are in fact instances of the generic R-convolution kernel proposed by Haussler [2]. R-convolution is a generic way for defining graph kernels based on comparing all pairs of decomposed substructures. Specifically, any graph decomposition can be used to define a kernel, e.g., the graph kernels based on comparing all pairs of decomposed a) walks [3], b) paths [4] and c) restricted subgraphs [5] or subtree structures [6], [7], [8], [9]. One main drawback arising in these R-convolution kernels is that they ignore the relative locations of substructures. This occurs when an R-convolution kernel adds an unit value to the kernel function by roughly identifying a pair of isomorphic substructures. In other words, the R-convolution kernels cannot establish reliable structural correspondences between isomorphic substructures, i.e., the kernel measure does not depend on the relationships between substructures.

To overcome the drawback of ignoring structural correspondence information arising in the R-convolution kernel, Fröhlich et al. [10] introduce optimal assignment kernels. Here each pair of structures is aligned before comparison. Neuhaus and Bunke [11] propose an aligned random walk kernel where the alignments obtained are based on graph edit-distance. Bai et al. [12] develop an entropic matching kernel by aligning vectorial vertex signatures computed through depth-based representations [13]. This kernel can be seen as an aligned subtree kernel that incorporates explicit structural correspondences between subtrees. All these kernels address the drawback of neglecting relative locations between substructures arising in the R-convolution kernels. Unfortunately, these kernels cannot guarantee the transitivity between the alignments. In other words, given three vertices v, u and w, if v and u are aligned, and u and w are aligned, these kernels cannot guarantee that v and w are also aligned. On the other hand, Fröhlich et al. [10] have demonstrated that the transitive alignment step is necessary to guarantee the positive definiteness of the alignment kernels. As a result, these alignment kernels are not guaranteed to be positive definite. Furthermore, when computing the kernel value between a pair input graphs, the R-convolution kernels and the alignment kernels both ignore the information from other graphs in the dataset. These drawbacks limit the precision of the kernel-based similarity measure.

To address the shortcomings of existing graph kernels, we introduce a new transitive aligned subtree kernel, based on an improved transitive aligned Weisfeiler-Lehman method that is developed through depth-based representation alignments. Note that we choose to use the depth-based representation due to its ability to characterize the graph structure [13], but in theory any vectorial vertex signature could be employed. We theoretically show that this kernel not only overcomes the shortcoming of ignoring correspondence information between isomorphic substructures that arises in existing R-convolution kernels, but also guarantees the transitivity between the correspondence information, i.e., we can guarantee the positive definiteness for the resulting kernel. Furthermore, we show that the computation of the kernel value encapsulates structural information from the input pair of graphs as well as other graphs in the dataset. Thus, the new transitive aligned subtree kernel can reflect richer graph characteristics than existing graph kernels. Experiments demonstrate that our kernel can easily outperform state-of-the-art graph kernels in terms of classification accuracy.

II. Preliminary Concepts

In this section, we will introduce a number of preliminary notions that will be used in the remainder of the paper.

A. Depth-based Representations of Vertices

In this subsection, we show how to compute the depth-based representation around the vertices of a graph, as defined by Bai et al. in [13]. For a sample graph $G(V,E)$ and a vertex $v \in V$, let a vertex set N_v^K be

$$N_v^K = \{u \in V \mid S_G(v,u) \leq K\}, \tag{1}$$

where S_G is the shortest path matrix of G and $S_G(v,u)$ is the shortest path length between v and u. For the graph $G(V,E)$, the K-layer expansion subgraph $\mathcal{G}_v^K(\mathcal{V}_v^K;\mathcal{E}_v^K)$ around v is

$$\begin{cases} \mathcal{V}_v^K = \{u \in N_v^K\}; \\ \mathcal{E}_v^K = \{u,w \in N_v^K,\ (u,w) \in E\}. \end{cases} \tag{2}$$

Note that, if K is equal to the longest shortest path length L_{max} from v to the remaining vertices, the K-layer expansion

Preprint submitted to 23rd International Conference on Pattern Recognition.
Received April 17, 2016.

subgraph rooted at v is the global graph $G(V,E)$ itself. If $K > L_{max}$, we use the global graph G as the K-layer expansion subgraph, however we say that the K-expansion subgraph does not exist. For $G(V,E)$, the h-layer depth-based representation around $v \in V$ is

$$DB^h_G(v) = [H_S(\mathcal{G}^1_v), \cdots, H_S(\mathcal{G}^K_v), \cdots, H_S(\mathcal{G}^h_v)]^\top, \quad (3)$$

where $(K \leq h)$. $H_S(\mathcal{G}^K_v)$ is the Shannon entropy of $\mathcal{G}^K_v$ associated with the steady state random walk defined as

$$H_S(\mathcal{G}^K_v) = -\sum_{u \in \mathcal{V}^K_v} P(u) \log P(u), \quad (4)$$

where $P(u) = D_{\mathcal{G}^K_v}(u,u) / \sum_{w \in \mathcal{V}^K_{v_U}} D_{\mathcal{G}^K_{v_U}}(w,w)$ is the probability of the steady state random walk visiting $u \in \mathcal{V}^K_v$, and $D_{\mathcal{G}^K_v}$ is the diagonal degree matrix of $\mathcal{G}^K_v$.

The h-layer depth-based representation $DB^h_G(v)$ reflects the entropy-based information content flow through the family of K-layer expansion subgraphs rooted at v. It can be seen as a vectorial coordinate of v in the depth principle space.

B. Vertex Labels from Depth-based Alignments

For a sample graph $G(V,E)$, let $\{v_1, \ldots, v_{|V|}\}$ be the vertex set V of G. Based on Eq.(3), we compute the h-layer depth-based representations around the vertices of G. The h-layer centroid representation of G is defined as the mean vector of the h-layer depth-based representations of its vertices, i.e.,

$$\begin{aligned} C^h(G) &= [DB^h_G(v_1) + \cdots + DB^h_G(v_{|V|})]/|V| \\ &= [\frac{\Sigma^{|V|}_{i=1} H_S(\mathcal{G}^1_{v_i})}{|V|}, \cdots, \frac{\Sigma^{|V|}_{i=1} H_S(\mathcal{G}^h_{v_i})}{|V|}]^\top. \end{aligned} \quad (5)$$

Since the h-layer centroid representation of G is computed based on the h-layer depth-based representations around all its vertices, the centroid representation encapsulates the depth information of all vertices in terms of entropy measures. Thus, it can be seen as a vectorial signature representing the global structure of G.

Given a graph, we develop a centroid alignment method by aligning the depth-based representations of its vertices to a family of centroid representations of graphs as follows. Let $\mathbf{G} = \{G_1, \ldots, G_N\}$ be a set of N graphs under comparison. Given a sample graph $G_p(V_p, E_p) \in \mathbf{G}$, we commence by computing the h-layer depth-based representation $DB^h_{G_p}(v_{p;i})$ for each vertex $v_{p;i} \in V_p$, based on Eq.(3). Moreover, we compute the family of h-layer centroid representations $\mathbf{C} = \{C^h(G_1), \ldots, C^h(G_N)\}$ for the graphs in $\mathbf{G}$ through the h-layer depth-based representations, based on Eq.(5). For the graph $G_p \in \mathbf{G}$, we align its h-layer depth-based representations of vertices to the family of centroid representations $\mathbf{C}$. To this end, we compute the Euclidean distance between $DB^h_{G_p}(v_{p;i})$ and $C^h(G_n)$ as the element $R^h_{G_p;\mathbf{C}}(i,n)$ of the affinity matrix $R^h_{G_p;\mathbf{C}}$, i.e.,

$$R^h_{G_p;\mathbf{C}}(i,n) = \|DB^h_{G_p}(v_{p;i}) - C^h(G_n)\|_2, \quad (6)$$

where $C^h(G_n)$ is the centroid representation of the n-th graph G_n in $\mathbf{G}$, and $R^h_{G_p;\mathbf{C}}$ is a $|V_p| \times N$ matrix. $R^h_{G_p;\mathbf{C}}(i,n)$ represents the distance between the vertex $v_{p;i}$ of G_p and the graph G_n in depth principle space. Furthermore, for the affinity matrix $R^h_{G_p;\mathbf{C}}$, the rows index the vertices of G_p, and the columns index the graphs of the graph set $\mathbf{G}$.

Given the affinity matrix $R^h_{G_p;\mathbf{C}}$, if $R^h_{G_p;\mathbf{C}}(i,\hat{n})$ is the smallest element in the ith row, we say that the h-layer depth-based representation of $v_{p;i}$ is aligned to the centroid representation of $G_{\hat{n}}$. In other words, for the graphs in $\mathbf{G}$, $G_{\hat{n}} \in \mathbf{C}$ is the closest graph to the vertex $v_{p;i} \in V_p$ in the depth principle space. Thus, we assign $v_{p;i}$ a vertex label as $\mathcal{L}_{DB}(v_{p;i}) = \hat{n}$, i.e.,

$$\mathcal{L}^h_{DB}(v_{p;i}) = \arg \min_{n \in \{1,2,\ldots,N\}} R^h_{G_p;\mathbf{C}}(i,n). \quad (7)$$

Note that, in some instances, the smallest in the ith row may not be unique, i.e, $v_{p;i} \in V_p$ is aligned to more than one centroid in $\mathbf{C}$. In this case, we randomly select one element. This guarantees that each vertex is aligned to at most one centroid representation. In other words, we assign each vertex an unique label based on the centroid alignment method.

C. Depth-based Vertex Alignments

The centroid depth-based alignment method provides a way of identifying correspondence information between a pair of vertices. Let $\mathbf{G} = \{G_1, \ldots, G_{\hat{n}}, \ldots, G_N\}$ be a set of N graphs under comparisons. For a pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ from $\mathbf{G}$, if $R^h_{G_p;\mathbf{C}}(i,\hat{n})$ and $R^h_{G_q;\mathbf{C}}(j,\hat{n})$ are both the smallest elements in rows i and j of their affinity matrices $R^h_{G_p;\mathbf{C}}$ and $R^h_{G_q;\mathbf{C}}$ respectively, we say that the vertex $v_{p;i}$ and the vertex $v_{q;j}$ are aligned (i.e., we establish a correspondence between $v_{p;i}$ and $v_{q;j}$ if $\mathcal{L}^h_{DB}(v_{p;i}) = \mathcal{L}^h_{DB}(v_{q;j})$). Since the h-layer depth-based representations of $v_{p;i}$ and $v_{q;i}$ are both aligned to the centroid representation $C^h(G_{\hat{n}})$, the h-layer depth-based representations of $v_{p;i}$ and $v_{q;i}$ are close to each other in the depth principle space. In other words, if two vertices are assigned the same vertex label $\mathcal{L}^h_{DB}$ based on the centroid depth-based alignment and Eq.(7), we say that the vertices are aligned.

III. The New Weisfeiler-Lehman Kernel

In this section, we introduce an improved Weisfeiler-Lehman method, based on the vertex labels computed through the centroid depth-based alignment. This in turn allows us to overcome the lack of correspondence information of the standard Weisfeiler-Lehman method. Moreover, we define a novel transitive aligned subtree kernel based on the new method. Finally, we indicate the relationship between the new kernel and state-of-the-art kernels, explaining the effectiveness of the new transitive aligned kernel.

A. Transitive Aligned Weisfeiler-Lehman Method

The original (1-dimensional) Weisfeiler-Lehman algorithm is a standard technique that is used to test whether two graphs are isomorphic [14]. Let $G(V,E)$ be a sample graph and $\mathcal{N}(v) = \{u|(v,u) \in E\}$ of $v \in V$ denote the neighbourhood vertices of v. At each iteration m where $m \geqslant 1$, this algorithm strengthens the existing label $\mathcal{L}_{m-1}(v)$ of a vertex $v \in V$ as a new label $\mathcal{L}_m(v)$ by taking the union of both the existing vertex label and its existing neighbouring vertex labels, i.e.,

$$\mathcal{L}_m(v) = \bigcup_{u \in \mathcal{N}(v)} \{\mathcal{L}_{m-1}(v), \mathcal{L}_{m-1}(u)\}, \quad (8)$$

Preprint submitted to 23rd International Conference on Pattern Recognition.
Received April 17, 2016.

where the existing label $\mathcal{L}_{m-1}(v)$ of v is computed at last iteration $m-1$. When $m=1$, the existing label of v is its original vertex label written as $\mathcal{L}_0(v)$. Furthermore, note that at each iteration m the strengthened label of v corresponds to a subtree of height m rooted at v. It can be shown that, for a pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$, if the strengthened labels of $v_p \in V_p$ and $v_q \in V_q$ at m iteration are identical, then the corresponding subtrees rooted at v_p and v_q are isomorphic. Thus, the original Weisfeiler-Lehman method provides a way of defining a subtree kernel for graphs by counting the number of isomorphic subtree pairs, i.e., the number of identical strengthened vertex labels. Examples include the fast Weisfeiler-Lehman subtree kernel developed by Shervashidze et al. [7]. Unfortunately, like the existing R-convolution kernels, this kernel cannot establish reliable correspondence information between the isomorphic subtrees. Because the original Weisfeiler-Lehman method cannot identify the correspondences between pairwise vertices of the subtrees. To overcome this drawback, we propose a new improved transitive aligned Weisfeiler-Lehman method through the centroid depth-based alignment. Algorithm 1 shows the pseudo code of the new method.

Algorithm 1: Transitive aligned Weisfeiler-Lehman method

1) Initialization: Input a set of N graphs $\mathbf{G} = \{G_1, \ldots, G_N\}$. Set ***m=0***. Compute the h-layer depth-based and centroid representations for vertices and graphs, respectively. For a graph $G(V, E)$, assign each vertex $v \in V$ an initial label

$$\mathcal{L}_{WL}^{(m,h)}(v) = \mathcal{L}_{WL}^{(0,h)}(v) = \{\mathcal{L}_0(v), \mathcal{L}_{DB}^h(v)\}, \tag{9}$$

where $\mathcal{L}_0$ is the original vertex label of v, $\mathcal{L}_{DB}(v)$ is the vertex label defined by Eq.(7) through centroid depth-based alignments, and $\mathcal{L}_{DB}^h(v)$ is an integer between 1 and N.

2) Updating: Set ***m=m+1***. For v, assign it a new label list as

$$\mathcal{L}_{WL}^{(m,h)}(v) = \bigcup_{u \in \mathcal{N}(v)} \{\mathcal{L}_{WL}^{(m-1,h)}(v), \mathcal{L}_{WL}^{(m-1,h)}(u)\}, \tag{10}$$

where $\mathcal{L}_{WL}^{(m-1,h)}(u)$ is arranged in ascending order.

3) Repeat steps 2 to 3 until m achieves a pre-specified value.

The transitive aligned Weisfeiler-Lehman method essentially follows the definition of its original version. The main difference lies in the fact that we associate each original vertex label $\mathcal{L}_0(v)$ with an extra vertex label $\mathcal{L}_{DB}(v)$ defined by Eq.(7). In other words, the improved method encapsulates the correspondence information identified by the centroid depth-based alignment into the process of strengthening vertex labels. More specifically, we make the following observations. **1)** First, like the original Weisfeiler-Lehman method, each strengthened vertex label $\mathcal{L}_{WL}^{(m,h)}(v)$ corresponds to a subtree of height m rooted at v. For a pair of vertices v and u, if their strengthened vertex labels are equivalent, the subtrees rooted at v and u are isomorphic. **2)** Second, unlike the original Weisfeiler-Lehman method, which is unable to establish reliable structural correspondences between the identified isomorphic subtrees, the new method encapsulates the locational correspondence between the rooted vertices of the isomorphic subtrees. In fact, if two subtrees identified by the new method are isomorphic, their vertex labels assigned by Eq.(7) are the same, i.e., the root vertices are aligned. In other words, the isomorphic subtrees identified by the new method can be seen as isomorphic subtrees rooted at aligned vertices. Thus, the new Weisfeiler-Lehman method overcomes the shortcoming of ignoring locational correspondence information. **3)** Third, the correspondences between vertices identified by the new method are transitive, i.e., for three vertices $v_p \in V_p$, $v_q \in V_q$ and $v_r \in V_r$ of three graphs $G_p(V_p, E_p)$, $G_q(V_q, E_q)$ and $G_r(V_r, E_r)$ in graph set $\mathbf{G}$, if v_p and v_q are both aligned to the centroid representation $C^h(G_{\hat{n}})$ of the $\hat{n}$-th graph $G_{\hat{n}} \in \mathbf{G}$ (i.e., v_p and v_q are aligned), and v_r and v_q are both aligned to $C^h(G_{\hat{n}})$ (i.e., v_r and v_q are aligned), then v_p and v_r are also aligned since they are also both aligned to $C^h(G_{\hat{n}})$. This property follows from Eq.(7) and Algorithm 1.

The new improved Weisfeiler-Lehman method provides an elegant way of defining a transitive aligned subtree kernel, that not only identifies pairs of isomorphic subtrees but also establishes locational correspondences between them.

B. The Transitive Aligned Subtree Kernel

Let $\mathbf{G} = \{G_1, \ldots, G_p, \ldots, G_q, \ldots, G_N\}$ be a set of N graphs under comparisons. For each graph in $\mathbf{G}$, We compute the h-layer depth-based representations of its vertices. Similarly, we compute the set of h-layer centroid representations for the graphs in $\mathbf{G}$. Let $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ be a pair of graphs from $\mathbf{G}$. Based on Eq.(7) and Algorithm 1, we compute the strengthened labels $\mathcal{L}_{WL}^{(m,h)}(v_p)$ and $\mathcal{L}_{WL}^{(m,h)}(v_q)$ for vertices $v_p \in V_p$ and $v_q \in V_q$. The transitive aligned Weisfeiler-Lehman subtree kernel $k_{TAWL}^{(M,H)}$ is defined as

$$k_{TAWL}^{(M,H)}(G_p, G_q) = \sum_{m=0}^{M} \sum_{h=1}^{H} \sum_{v_p \in V_p} \sum_{v_q \in V_q} \delta\{\mathcal{L}_{WL}^{(m,h)}(v_p), \mathcal{L}_{WL}^{(m,h)}(v_q)\}, \tag{11}$$

where δ is the Dirac kernel, i.e., it is 1 if the arguments are equal and 0 otherwise. H is the maximum value of the parameter h for depth-based and centroid representations. M is the maximum value of the parameter m for the transitive aligned Weisfeiler-Lehman method. Furthermore, $\mathcal{L}_{WL}^{(m,h)}(v_p) = \mathcal{L}_{WL}^{(m,h)}(v_q)$ indicates that the subtrees corresponding to $\mathcal{L}_{WL}^{(m,h)}(v_p)$ and $\mathcal{L}_{WL}^{(m,h)}(v_q)$ are isomorphic, and the subtree rooted vertices v_p and v_q are aligned. The transitive aligned Weisfeiler-Lehman subtree kernel $k_{TAWL}^{(M,H)}$ is positive definite. This is because $k_{TAWL}^{(M,H)}$ counts the number of aligned isomorphic subtree pairs identified by the transitive aligned Weisfeiler-Lehman method, i.e., $k_{TAWL}^{(M,H)}$ can be seen as a kind of R-convolution kernel that counts pairs of isomorphic substructures. □

C. Advantages of the New Subtree Kernel

The new transitive aligned Weisfeiler-Lehman subtree kernel has some important properties that are not available to some state of the art graph kernels, e.g., the depth-based matching kernel [15] and the original Weisfeiler-Lehman subtree kernel [16]. To demonstrate the properties of the new kernel, we re-define our kernel in a manner that make its advantages and effectiveness clear.

Let $\mathbf{G} = \{G_1, \ldots, G_p, \ldots, G_q, \ldots, G_N\}$ be the set of N graphs under comparison. Based on Eq.(6), we compute the

affinity matrices for $G_p(V_p, E_q)$ and $G_q(V_q, E_q)$ as $R^h_{G_p;\mathbf{C}}$ and $R^h_{G_q;\mathbf{C}}$. Both $R^h_{G_p;\mathbf{C}}$ and $R^h_{G_q;\mathbf{C}}$ record the distances between the h-layer depth-based representations of the corresponding graph vertices and the h-layer centroid representations of the graphs in $\mathbf{G}$. Assume $G_n(V_n, E_n) \in \mathbf{G}$, and $n \in \{1, 2, \ldots, N\}$. If $R^h_{G_p;\mathbf{C}}(i, n)$ and $R^h_{G_q;\mathbf{C}}(j, n)$ are both the smallest elements in columns n of $R^h_{G_p;\mathbf{C}}$ and $R^h_{G_p;\mathbf{C}}$ respectively, we say that the vertex $v_{p;i}$ of G_p and the vertex $v_{q;j}$ of G_q are aligned, since they are both aligned to the same h-layer centroid representation of graph G_n. As a result, we identify an one-to-one correspondence between $v_{p;i}$ and $v_{q;j}$, i.e., $v_{p;i}$ and $v_{q;j}$ are matched. More formally, let the correspondence matrix $M^{(h)}_{G_p;G_q} \in \{0,1\}^{|V_p|\times|V_q|}$ record the correspondences between the vertices of G_p and G_q and satisfy

$$M^{(h)}_{G_p;G_q}(i,j) = \begin{cases} 1 & \text{if } R^{(h)}_{G_p;\mathbf{C}}(i,n) \text{ and } R^{(h)}_{G_p;\mathbf{C}}(j,n) \\ & \text{are both the smallest elements} \\ & \text{in their columns } n; \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

where $n \in \{1, 2, \ldots, N\}$, the rows i index the vertices of G_p, and the columns j index the the vertices of G_q.

Note that, for any pair of vertices $v_{p;i} \in V_p$ and $v_{q;j} \in V_q$, if their strengthened labels $\mathcal{L}^{(m,h)}_{WL}(v_{p;i})$ and $\mathcal{L}^{(m,h)}_{WL}(v_{q;j})$ computed from Algorithm 1 are identical, they must be aligned, i.e., $M^{(h)}_{G_p;G_q}(i,j) = 1$. This can be observed from Eq.(7) and Eq.(9), since the vertex alignment information computed from Eq.(7) is associated by Eq.(9). Moreover, if $\mathcal{L}^{(m,h)}_{WL}(v_{p;i})$ and $\mathcal{L}^{(m,h)}_{WL}(v_{q;j})$ are identical, the strengthened labels $\mathcal{L}_m(v_{p;i})$ and $\mathcal{L}_m(v_{q;j})$ computed from Eq.(8) (i.e., the original Weisfeiler-Lehman method) are also identical. In essence, the new transitive aligned Weisfeiler-Lehman method can be seen as the original method associated with the aligned labels from the centroid depth-based alignments. This can be observed from Eq.(8), Eq.(9) and Eq.(10).

Based on the above observations, we can define a new correspondence matrix $C^{(m,h)}_{G_p;G_q} \in \{0,1\}^{|V_p|\times|V_q|}$ that not only records the correspondence information between vertices but also reflects the equivalence between the vertex strengthened labels computed from the new transitive aligned method. The element $C^{(m,h)}_{G_p;G_q}(i,j)$ is defined as

$$C^{(m,h)}_{G_p;G_q}(i,j) = \begin{cases} 1 & \text{if } M^{(h)}_{G_p;G_q}(i,j) = 1 \\ & \text{and } \mathcal{L}_m(v_{p;i}) = \mathcal{L}_m(u_{q;j}); \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

The transitive aligned Weisfeiler-Lehman subtree kernel $k^{(M,H)}_{TAWL}$ can be re-defined as

$$k^{(M,H)}_{TAWL}(G_p, G_q) = \sum_{m=0}^{M}\sum_{h=1}^{H}\sum_{v_p\in V_p}\sum_{v_q\in V_q} C^{(m,h)}_{G_p;G_q}. \tag{14}$$

As a result, $k^{(M,H)}_{TAWL}(G_p, G_q)$ can be seen as a matching kernel that counts the number of aligned vertex pairs between G_p and G_q over the M correspondence matrices $C^{(m,h)}_{G_p;G_q}$ based on the h-layer depth-based and centroid representations. Compared to the existing depth-based matching kernel [15] and the original Weisfeiler-Lehman subtree kernel [16], Eq.(13) and Eq.(14) indicate the following advantages of the new subtree kernel $k^{(M,H)}_{TAWL}$.

1) First, unlike the depth-based matching kernel that identifies the vertex correspondences by directly aligning the h-layer depth-based representations between vertices, the new kernel $k^{(M,H)}_{TAWL}$ identifies the vertex correspondences by evaluating whether the depth-based representations of vertices are aligned to the same h-layer centroid representation. Thus, the new kernel $k^{(M,H)}_{TAWL}$ can guarantee the transitivity between pairs of aligned vertices, and is thus positive definite. By contrast, the depth-based matching kernel cannot guarantee the alignment transitivity and the **positive definiteness**.

2) Second, unlike the Weisfeiler-Lehman subtree kernel that roughly counts pairs of isomorphic subtrees identified by the original Weisfeiler-Lehman method, for the new kernel $k^{(M,H)}_{TAWL}$ only a pair of isomorphic subtrees rooted at aligned vertices will be considered to contribute the kernel value. In other words, $k^{(M,H)}_{TAWL}$ establishes the locational correspondence information between isomorphic subtrees. Thus, the new kernel $k^{(M,H)}_{TAWL}$ overcomes shortcoming of ignoring correspondence information between substructures that arises in the original Weisfeiler-Lehman subtree kernel and other R-convolution kernels [2], [17], [6], [18].

3) Third, unlike most existing state of the art graph kernels [19], [4], [17], [6], [5], [18] that only capture graph characteristics for each pair of graphs under comparison, the computation of the new kernel $k^{(M,H)}_{TAWL}$ for a pair of graphs incorporates the information of all graphs under comparison. This is because the transitive aligned Weisfeiler-Lehman method associates the alignment between the h-layer depth-based representations of the vertices and the family of centroid representations of all graphs $G_n \in \mathbf{G}$. This can be observed from Eq.(13). As a result, the new kernel $k^{(M,H)}_{TAWL}$ can reflect richer graph characteristics than most existing graph kernels.

The above observations indicate the advantages of the new transitive aligned Weisfeiler-Lehman subtree kernel, and thus explain the effectiveness of the new kernel.

D. Discussion and Related Work

The new transitive aligned subtree kernel $k^{(M,H)}_{TAWL}$ is related to the depth-based representation defined in [13]. However, there is a significant difference. The depth-based representation in [13] is rooted at the centroid vertex that is selected by evaluating the variance of shortest path lengths between vertices, and the resulting representation is used as a kind of embedding vector for a graph. Embedding graphs onto a vectorial space inevitably approximates the structural correlations in a low dimensional space, and thus leads to information loss. By contrast, the transitive aligned subtree kernel $k^{(M,H)}_{TAWL}$ is computed through both h-layer depth-based and centroid representations. Moreover, the kernel computation of $k^{(M,H)}_{TAWL}$ can represents graphs in a high dimensional kernel space and thus better preserves structural information.

Note that the depth-based matching kernel [15], the entropic matching kernel [12] and the Jensen-Shannon matching kernel [20] are also related to the depth-based representation developed in [13]. Furthermore, like the new transitive aligned

Preprint submitted to 23rd International Conference on Pattern Recognition.
Received April 17, 2016.

subtree kernel $k_{TAWL}^{(M,H)}$, these matching kernels attempt to preserve graph structures by kernelizing the depth-based representation and locating the correspondence information between substructures. However, unlike the proposed kernel $k_{TAWL}^{(M,H)}$, all these kernels identify the correspondences between vertices by directly aligning their vectorial representations, e.g., the h-layer entropic representations for the entropic matching kernel and the h-layer Jensen-Shannon representations for the Jensen-Shannon matching kernel. Thus, like the depth-based matching kernel, these matching kernels also cannot guarantee the transitivity between pairs of aligned vertices. By contrast, $k_{TAWL}^{(M,H)}$ ensures the transitivity of the correspondence information, since it relies on the centroid depth-based alignment. As a result, the proposed subtree kernel $k_{TAWL}^{(M,H)}$ can capture more precise information for graphs than the alternative matching kernels.

Furthermore, the fast depth-based subgraph kernels in [21] are also related to the depth-based representations. However, unlike the proposed kernel, these subgraph kernels are instances of the R-convolution kernels, and thus they also suffer from the drawback of ignoring the correspondences between substructures. Finally, note that, unlike the aforementioned alignment kernels in Section I, our new kernel can guarantee the transitive alignment between substructures, and thus guarantee the positive definiteness.

E. Computational Analysis

For a set of N graphs each of which has X vertices, computing the transitive aligned Weisfeiler-Lehman subtree kernel $k_{TAWL}^{(M,H)}$ has time complexity $O(HNX^3 + MX^2)$. In fact, the computation of the expansion subgraphs relies on the shortest path computation, which in turn has time complexity $O(X^3)$. Given an expansion subgraph, computing the required Shannon entropy has time complexity $O(X^2)$. Thus, computing the h-layer depth-based representations around the X vertices of each of the N graphs has time complexity $O(HNX^3)$ (H is the largest value of h). Computing the N centroid representations for the N graphs has time complexity $O(HXN)$. Furthermore, computing the transitive aligned Weisfeiler-Lehman method over M iterations has time complexity $O(MX^2)$ (M is the largest value of m), since it needs to visit the X^2 entries of the adjacency matrix of each graph. Identifying the identical strengthened labels between X^2 pairs of vertices also has time complexity $O(MX^2)$. As a result, the total time complexity for $k_{TAWL}^{(M,H)}$ is $O(HNX^3 + MX^2)$. This indicates that our kernel can be computed in polynomial time.

IV. Experimental Results

A. Evaluations on Graph Classification

We evaluate our kernels on standard graph datasets, including BAR31, COIL5, GatorBait, PPIs and CATH2. We choose these datasets as they represents standard yet challenging benchmarks for testing graph classification. A summary of the characteristics of these datasets is shown in Table.I. More details can be found in [21].

Experimental Setup: We evaluate the performance of the transitive aligned Weisfeiler-Lehman subtree kernel (TAWL) on graph classification problems. We compare our kernel with several alternative state-of-the-art graph kernels. These graph kernels include 1) the fast depth-based subgraph kernels (ISK) [21], 2) the aligned subtree kernel (ASK) [12], 3) the Weisfeiler-Lehman subtree kernel (WLSK) [7], 4) the shortest path graph kernel (SPGK) [4], 5) the graphlet count graph kernel with graphlet of size 4 (GCGK) [22]. For the TAWL kernel, we set $M = 3$ (this is because subtrees of depth greater than 3 are often too descriptive, resulting in the identification of a very small number of isomorphic subtrees) and $H = 30$ (this reflects the fact that the length of the shortest paths in the datasets considered is seldom above 30). The parameters for other kernels are individually optimized on each dataset. The classification performance of each kernel is evaluated using 10-fold cross-validation and a C-Support Vector Machine (C-SVM). In particular, we make use of the LIBSVM library [23]. For each fold, we choose the parameters of each kernel as well as the C parameter of the C-SVM by cross-validation on the training data. For each kernel and dataset, we repeat the whole experiment 10 times and we compute the average classification accuracy and standard error. The average classification accuracy ($\pm$ standard error) and the runtime for each kernel are shown in Table II and Table III, respectively. The runtime is measured under Matlab R2011a running on a 2.5GHz Intel 2-Core processor.

Results and Discussions: In terms of classification accuracy, the TAWL kernel outperforms all alternative kernels, with the exception of the ASK kernel on the BAR31 dataset. However, the accuracy of the ASK kernel is only marginally higher than that of the TAWL kernel. On the other hand, the TAWL kernel significantly outperforms the ASK kernel on the PPIs, COIL and CATH2 datasets. The reasons for the effectiveness are threefold. First, unlike the ISK, WLSK, SPGK and GCGK kernels, which ignore correspondence information between substructures, the TAWL kernel establishes reliable locational correspondences between subtrees. Second, compared to the ASK kernel, which also establishes correspondence information between substructures, only the TAWL kernel can guarantee the transitivity between pairs of aligned substructures. Third, unlike all alternative kernels, which only take into account the structural characteristics of the pair of input graphs, our kernel also encapsulates the information from other graphs. In this sense, our kernel can be seen as an instance of transductive learning [24], where all the graphs available (both from the traning and test sets) are used to compute the graph centroids. Note, however, that we do not observe the class labels of the test graphs during the training. Finally, note that, compared to the WLSK kernel associated with the original Weisfeiler-Lehman method, the performance of our kernel is significantly improved. This demonstrates the effectiveness of the new method. In terms of the runtime, our kernel is not the fastest kernel, but its runtime is sufficiently low.

V. Conclusions

In this paper we have developed a transitive aligned Weisfeiler-Lehman subtree kernel. We have theoretically shown its advantages when compared against most existing kernels. An extensive set of experiments demonstrated the effectiveness of the new kernel.

Preprint submitted to 23rd International Conference on Pattern Recognition.
Received April 17, 2016.

TABLE I. GRAPH DATASETS.

Datasets	BAR31	GatorBait	COIL5	PPIs	CATH2
Max# nodes	220	545	241	218	568
Min# nodes	41	239	72	3	143
Mean# nodes	95.42	348.70	144.90	109.63	308.03
#graphs	300	100	360	219	190
#classes	15	30	5	5	2

TABLE II. CLASSIFICATION ACCURACY (IN % $\pm$ STANDARD ERROR).

Datasets	BAR31	GatorBait	COIL5	PPIs	CATH2
TAWL	$70.53 \pm .70$	**18.30** $\pm .93$	**75.30** $\pm .36$	**88.85** $\pm .43$	**85.57** $\pm .77$
ASK	**73.10** $\pm .67$	$7.50 \pm .74$	$53.67 \pm .64$	$80.14 \pm .73$	$78.52 \pm .67$
WLSK	$58.53 \pm .53$	$10.10 \pm .61$	33.16 ± 1.01	$88.09 \pm .41$	$67.36 \pm .63$
ISK	$62.80 \pm .47$	$11.40 \pm .52$	$38.30 \pm .56$	$79.47 \pm .32$	$67.55 \pm .67$
SPGK	$55.73 \pm .44$	9.00 ± 75	$69.66 \pm .52$	$59.04 \pm .44$	$81.89 \pm .63$
GCGK	$23.40 \pm .60$	$8.40 \pm .83$	$66.41 \pm .53$	$46.61 \pm .47$	73.68 ± 1.09

TABLE III. RUNTIME FOR COMPUTING THE KERNEL MATRICES.

Datasets	BAR31	GatorBait	COIL5	PPIs	CATH2
TAWL	$11'12"$	$5'16"$	$8'12"$	$2'56"$	$5'57"$
ASK	$8'40"$	$14'50"$	$30'9"$	$4'45"$	$44'10"$
WLSK	$30"$	$33"$	$1'5"$	$20"$	$53"$
ISK	$3'50"$	$6'59"$	$9'55"$	$2'40"$	$6'51"$
SPGK	$11"$	$2'25"$	$31"$	$22"$	$4'13"$
GCGK	$1"$	$3"$	$4"$	$4"$	$8"$

Acknowledgments

This work is supported by National Natural Science Foundation of China(Grant no.61503422) and the Open Projects Program of National Laboratory of Pattern Recognition. Edwin R. Hancock is supported by a Royal Society Wolfson Research Merit Award.

REFERENCES

[1] M. Neuhaus and H. Bunke, "Bridging the gap between graph edit distance and kernel machines," *World Scientific*, 2007.

[2] D. Haussler, "Convolution kernels on discrete structures," in *Technical Report UCS-CRL-99-10*, Santa Cruz, CA, USA, 1999.

[3] T. Gärtner, P. Flach, and S. Wrobel, "On graph kernels: hardness results and efficient alternatives," in *Proceedings of COLT*, 2003, pp. 129–143.

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

[5] N. Kriege and P. Mutzel, "Subgraph matching kernels for attributed graphs," in *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012.

[6] F. Costa and K. D. Grave, "Fast neighborhood subgraph pairwise distance kernel," in *Proceedings of ICML*, 2010, pp. 255–262.

[7] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels," *Journal of Machine Learning Research*, vol. 12, pp. 2539–2561, 2011.

[8] A. Feragen, N. Kasenburg, J. Petersen, M. de Bruijne, and K. M. Borgwardt, "Scalable kernels for graphs with continuous attributes," in *Proceedings of NIPS*, 2013, pp. 216–224.

[9] M. Neumann, N. Patricia, R. Garnett, and K. Kersting, "Efficient graph kernels by randomization," in *Proceedings of ECML-PKDD*, 2012, pp. 378–393.

[10] H. Fröhlich, J. K. Wegner, F. Sieker, and A. Zell, "Optimal assignment kernels for attributed molecular graphs," in *Proceedings of ICML*, 2005, pp. 225–232.

[11] M. Neuhaus and H. Bunke, "Edit distance-based kernel functions for structural pattern classification," *Pattern Recognition*, vol. 39, no. 10, pp. 1852–1863, 2006.

[12] L. Bai, L. Rossi, Z. Zhang, and E. R. Hancock, "An aligned subtree kernel for weighted graphs," in *Proceedings of ICML*, 2015, pp. 30–39.

[13] L. Bai and E. R. Hancock, "Depth-based complexity traces of graphs," *Pattern Recognition*, vol. 47, no. 3, pp. 1172–1186, 2014.

[14] B. Weisfeiler and A. Lehman, "A reduction of a graph to a canonical form and an algebra arising during this reduction," *Nauchno-Technicheskaya Informatsia*, vol. Ser.2, no. 9, 1968.

[15] L. Bai, P. Ren, X. Bai, and E. R. Hancock, "A graph kernel from the depth-based representation," in *Proceedings of S+SSPR*, 2014, pp. 1–11.

[16] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt, "Weisfeiler-lehman graph kernels," *Journal of Machine Learning Research*, vol. 1, pp. 1–48, 2010.

[17] F. R. Bach, "Graph kernels between point clouds," in *Proceedings of ICML*, 2008, pp. 25–32.

[18] F. Aziz, R. C. Wilson, and E. R. Hancock, "Backtrackless walks on a graph," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 6, pp. 977–989, 2013.

[19] H. Kashima, K. Tsuda, and A. Inokuchi, "Marginalized kernels between labeled graphs," in *Proceedings of ICML*, 2003, pp. 321–328.

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

[21] L. Bai and E. R. Hancock, "Fast depth-based subgraph kernels for unattributed graphs," *Pattern Recognition*, vol. 50, pp. 233–245, 2016.

[22] N. Shervashidze, S. Vishwanathan, T. Petri, K. Mehlhorn, and K. Borgwardt, "Efficient graphlet kernels for large graph comparison," *Journal of Machine Learning Research*, vol. 5, pp. 488–495, 2009.

[23] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm*, 2011.

[24] A. Gammerman, V. Vovk, and V. Vapnik, "Learning by transduction," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1998, pp. 148–155.