A Graph Kernel from the Depth-Based Representation

Lu Bai[1], Peng Ren[2], Xiao Bai[3], and Edwin R. Hancock[1,⋆]

[1] Department of Computer Science, University of York, York, UK
[2] College of Information and Control Engineering, China University of Petroleum, China
[3] School of Computer Science and Engineering, Beihang University, Beijing, China

Abstract. In this paper we develop a novel graph kernel by matching the depth-based substructures in graphs. We commence by describing how to compute the Shannon entropy of a graph using random walks. We then develop an h-layer depth-based representations for a graph, which is effected by measuring the Shannon entropies of a family of K-layer expansion subgraphs derived from a vertex of the graph. The depth-based representations characterize graphs in terms of high dimensional depth-based complexity information. Based on the new representation, we establish a possible correspondence between vertices of two graphs that allows us to construct a matching-based graph kernel. Experiments on graphs from computer vision datasets demonstrate the effectiveness of our kernel.

Keywords: Depth-based representation, graph matching, graph kernels.

1 Introduction

Graph-based representations are widely used in computer vision and pattern recognition for characterizing shapes and structures [1, 2]. In this context, there has recently been an increasing interest in evolving graph kernels into kernel machines (e.g., a Support Vector Machine (SVM)) for graph classification [3–5]. A graph kernel is usually defined in terms of a (dis)similarity measure between graphs. Haussler [6] proposed a general graph kernel formulation referred to as R-convolution kernel, which is effected by decomposing graphs into substructures separately and then measuring the pairwise (dis)similarities between the resulting substructures. For a pair of sample graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$, suppose $\{\mathcal{S}_{p;1}, \ldots, \mathcal{S}_{p;x}, \ldots, \mathcal{S}_{p;N_p}\}$ and $\{\mathcal{S}_{q;1}, \ldots, \mathcal{S}_{p;y}, \ldots, \mathcal{S}_{q;N_q}\}$ are the sets of the substructures of $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ respectively. A R-convolution kernel k_R between $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ can be defined as

$$k_R(G_p, G_q) = \sum_{x=1}^{N_p} \sum_{y=1}^{N_q} s(\mathcal{S}_{p;x}, \mathcal{S}_{q;y}),$$

where $s(\mathcal{S}_{p;x}, \mathcal{S}_{q;y})$ is the (dis)similarity measure between the substructures $\mathcal{S}_{p;x}$ and $\mathcal{S}_{q;y}$, and k_R proves to be a positive definite kernel.

From the perspective of R-convolution, existing graph kernels can be generally categorized into three classes [3], i.e. graph kernels based on comparing all pairs of a)

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

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 1–11, 2014.

walks, b) paths and c) restricted subgraph and subtree structures. One major limitation of these existing graph kernels is that in practical computation they do not easily scale up to structures of large sizes. To overcome this problem, most existing graph kernels compromise to use substructures of limited sizes, and examples include a) the shortest path graph kernel [7], b) the graphlet count graph kernel [3], c) the fast neighborhood subgraph pairwise distance kernel [9], and d) the backtrackless kernel [10] based on non-backtracking cycles that are identified by the Ihara zeta function [11]. Although this strategy curbs the notorious inefficiency of comparing large substructures, graph kernels with limited sized substructures simply can only reflect restricted topological characteristics of a graph.

In this work, we aim to overcome the topological restrictions by characterizing graph substructures in terms of depth-based representations [12], which provide richer topological features but also easily scale up to as large size as the original graph. To this end, we investigate how to incorporate the depth-based representations into graph matching and thus develop a novel graph kernel which not only reflects the rich depth-based structure of graphs but also enables a fast computation. We commence by computing the depth-based complexity traces [13] of a graph around each vertex. To avoid the burdensome subgraph enumeration in computing the intrinsic complexity [12], we compute the depth-based representation around a vertex by measuring a fast Shannon entropy of its expansion subgraph. The depth-based representation gauges the Shannon entropy flow via the expansion subgraphs, and thus reflects a high dimensional complexity characteristics of the graph around the vertex. Based on the obtained depth-based representations for two graphs we develop a matching strategy similar to that Scott et al. [16] previously used for point set matching. The purpose of this step is to match the vertices of the graphs by using the vertex information extracted from the depth-based representations. For a pair of graphs, we use the Euclidean distance between the depth-based representations to compute an affinity matrix. The correspondences between pairwise vertices are obtained from the affinity matrix. The affinity matrix characterizes local structural similarity between a pair of graphs and can be used for graphs of different sizes. Finally, we develop the novel depth-based graph matching kernel by counting the matched vertex pairs. We empirically demonstrate the effectiveness and efficiency of our new graph kernel on graphs from computer vision datasets.

The remainder of this paper is organized as follows. Section 2 presents the definition of depth-based representations for graphs. Section 3 presents the definition of the new graph matching kernel. Section 4 provides our experimental evaluations. Finally, Section 5 concludes our work.

2 Depth-Based Representations

We commence by introducing a fast Shannon entropy measure for a graph. Moreover, we show how to compute a depth-based representation around a vertex of a graph.

2.1 The Shannon Entropy of a Graph

We compute the Shannon entropy of a graph based on steady state random walks on the graph. Consider a graph $G(V, E)$ where V denotes the set of vertices and $E \subseteq V \times V$

denotes the set of undirected edges. The adjacency matrix A for $G(V,E)$ is a symmetric $|V| \times |V|$ matrix with the (v,u)th entry

$$A(v,u) = \begin{cases} 1 \text{ if} (v,u) \in E; \\ 0 \text{ otherwise.} \end{cases} \tag{1}$$

The vertex degree matrix of $G(V,E)$ is a diagonal matrix D whose vth diagonal element is given by $D(v,v) = d(v) = \sum_{v,u \in V} A(v,u)$. As a result, the probability of a steady state random walk on $G(V,E)$ visiting vertex v is $P_G(v) = d(v)/\sum_{v,u \in V} d(u)$. The Shannon entropy of $G(V,E)$ associated with the steady state random walk is

$$H_S(G) = -\sum_{v \in V} P_G(v) \log P_G(v). \tag{2}$$

For a graph $G(V,E)$ ($|V| = n$), computing the Shannon entropy $H_S(G)$ requires time complexity $O(n^2)$. This is because $H_S(G)$ relies on the degree matrix D that is computed by visiting all the n^2 entries of the adjacency matrix A. This indicates that the Shannon entropy H_S defined in Eq.(2) can be efficiently computed. By contrast, the von Neumnann entropy defined in [14] and the Shannon entropy associated with an information functional defined in [15] both require time complexity $O(n^3)$.

2.2 The Depth-Based Representation for a Graph

For an undirected graph $G(V,E)$, the shortest path $S_G(v,u)$ between a pair of vertices v and u can be computed by using Dijkstra algorithm. The matrix S_G whose element $S_G(v,u)$ represents the shortest path length between v and u is referred to as the shortest path matrix for G. Let N_v^K be a subset of V satisfying $N_v^K = \{u \in V \mid S_G(v,u) \leq K\}$. For G, the K-layer expansion subgraph $\mathcal{G}_v^K(\mathcal{V}_v^K; \mathcal{E}_v^K)$ around vertex v is

$$\begin{cases} \mathcal{V}_v^K &= \{u \in N_v^K\}; \\ \mathcal{E}_v^K &= \{(u,v) \subset N_v^K \mid (u,v) \in E\}. \end{cases} \tag{3}$$

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

Definition 2.1 (h-layer depth-based representation). For a graph $G(V,E)$ and a vertex $v \in V$, the h-layer depth-based representation around v is a h dimensional vector

$$D_G^h(v) = [H_S(\mathcal{G}_v^1), \cdots, H_S(\mathcal{G}_v^K), \cdots, H_S(\mathcal{G}_v^h)]^T \tag{4}$$

where h ($h \leq L_v$) is the length of the shortest paths from v to other vertices in $G(V,E)$, $\mathcal{G}_v^K(\mathcal{V}_v^K; \mathcal{E}_v^K)$ ($K \leq h$) is the K-layer expansion subgraph of $G(V,E)$ around v, and $H_S(\mathcal{G}_v^K)$ is the Shannon entropy of $\mathcal{G}_v^K$ and is defined in Eq.(2). □

For a graph $G(V,E)$ ($|V| = n$) and a vertex $v \in V$, computing the h-layer depth-based representation $D_G^h(v)$ of $G(V,E)$ around v requires time complexity $O(hn^2)$. This follows the definitions in Eq.(3). For $G(V,E)$, the Dijkstra algorithm requires time complexity $O(n^2)$. Computing the Shannon entropies of the h K-layer expansion

subgraphs, which are derived from v, requires time complexity $O(hn^2)$. Hence, the whole time complexity is $O(hn^2)$. This indicates that the h-layer depth-based representation around a vertex of a graph can be efficiently computed. Key to this efficiency is that the Shannon entropy on an expansion subgraph only requires time complexity $O(n^2)$. By contrast, in [12] the intrinsic complexity measure of an expansion subgraph for measuring the depth-based complexity requires time complexity $O(n^5)$.

Moreover, the h-layer depth-based representation $D_G^h(v)$ characterizes the depth-based complexity of $G(V,E)$ with regard to the vertex v in a h dimensional feature space. It captures the rich depth-based complexity characteristics of substructures around the vertex v in terms of the entropies of the K-layer expansion subgraphs with K increasing from 1 to h. In contrast, the existing graph kernels in the literatures [4, 4, 5] tend to compute similarities on global subgraphs of limited sizes and can only capture restricted characteristics of graphs.

3 Depth-Based Graph Matching Kernel

We describe how the depth-based representations can be used for graph matching. Furthermore, we define a novel graph kernel based on the proposed matching method.

3.1 Depth-Based Graph Matching

We develop a matching method similar to that introduced in [16, 17] for point set matching, which computes an affinity matrix in terms of the distances between points. In our work, for a vertex v of $G(V,E)$, we treat the h-layer depth-based representations $D_G^h(v)$ as the point coordinate associated with v. We use the Euclidean distance between the depth-based representations $D_{G_p}^h(v_i)$ and $D_{G_q}^h(u_j)$ as the distance measure of the pairwise vertices v_i and u_j of graphs $G_p(V_p,E_p)$ and $G_q(V_q,E_q)$, respectively. The affinity matrix element $R(i,j)$ is defined as

$$R(i,j) = \sqrt{[D_{G_p}^h(v_i) - D_{G_q}^h(u_j)]^T[D_{G_p}^h(v_i) - D_{G_q}^h(u_j)]}. \tag{5}$$

where R is a $|V_p| \times |V_q|$ matrix. The element $R(i,j)$ represents the dissimilarity between the vertex v_i in $G_p(V_p,E_p)$ and the vertex u_j in $G_q(V_q,E_q)$. The rows of $R(i,j)$ index the vertices of $G_p(V_p,E_p)$, and the columns index the vertices of $G_q(V_q,E_q)$. If $R(i,j)$ is the smallest element both in row i and in column j, there should be a one-to-one correspondence between the vertex v_i of G_p and the vertex u_j of G_q. We record the state of correspondence using the correspondence matrix $C \in \{0,1\}^{|V_p||V_q|}$ satisfying

$$C(i,j) = \begin{cases} 1 \text{ if } R(i,j) \text{ is the smallest element} \\ \quad \text{both in row } i \text{ and in column } j; \\ 0 \text{ otherwise.} \end{cases} \tag{6}$$

Eq.(6) implies that if $C(i,j) = 1$, the vertices v_i and v_j are matched. Note that, in row i or column j there may be two or more than two elements satisfying Eq.(6). In other words, for a pair of graphs a vertex from a graph may have two or more than two

matched vertices from the other graph. To assign a vertex one matched vertex at most, we update the matrix C by employing the Hungarian method that is widely used for solving the assignment problem (e.g., the bipartite graph matching problem) in polynomial time [18]. Here the matrix $C \in \{0,1\}^{|V_p||V_q|}$ can be seen as the incidence matrix of a bipartite graph $G_{pq}(V_p, V_q, E_{pq})$, where V_p and V_q are the two sets of partition parts and E_{pq} is the edge set. By performing the Hungarian algorithm on the incidence matrix $C \in \{0,1\}^{|V_p||V_q|}$ (i.e., the correspondence matrix of G_p and G_q) of the bipartite graph G_{pq}, we assign each vertex from G_p or G_q at most one matched vertex from the other graph G_q or G_p. Note finally that, directly performing the Hungarian algorithm on the matrix R can also assign each vertex from G_p or G_q an unique matched vertex. However, it cannot guarantee that each identified element is the smallest both in the row and column in R. This is because some vertices will not have matched vertices.

For a pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ ($|V_p| = |V_q| = n$). Computing the correspondence matrix $C \in \{0,1\}^{|V_p||V_q|}$ (i.e. the final correspondence matrix updated by the Hungarian algorithm) requires time complexity $O(hn^3)$. This follows the definition in Section 3.1. For G_p, computing its n h-layer depth-based representations derived from each of its vertices requires time complexity $O(hn^3)$, and it is the same for G_q. Computing each element of the affinity matrix R requires time complexity $O(h)$, and hence computing the whole affinity matrix R requires time complexity $O(hn^2)$. The computation of the correspondence matrix C need to enumerate all the n^2 pairs of elements in R and thus requires time complexity $O(n^2)$. The Hungarian algorithm on the matrix C requires time complexity $O(n^3)$. As a result, the whole time complexity is $O(hn^3)$.

3.2 A Depth-Based Graph Kernel

Based on the graph matching strategy in Section 3.1, we define a new graph kernel.

Definition 3.1 (The depth-based graph kernel). Consider $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ as a pair of sample graphs. Based on the definitions in Eq.(4), Eq.(5) and Eq.(6), and the Hungarian algorithm, we compute the correspondence matrix C. The depth-based graph kernel $k_{DB}^{(h)}$ using the h-layer depth-based representations of the graphs is

$$k_{DB}^{(h)}(G_p, G_q) = \sum_{i=1}^{|V_p|} \sum_{j=1}^{|V_q|} C(i,j). \tag{7}$$

which counts the number of matched vertex pairs between $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$. Note that, the kernel $k_{DB}^{(h)}$ can also accommodate attributed graphs by computing the number of pairwise matched vertices that have the same vertex label. □

Lemma 3.1. *The depth-based graph kernel $k_{DB}^{(h)}$ is positive definite (**pd**).*

Proof. Intuitively, the proposed depth-based graph kernel is **pd** because it counts pairs of matched vertices (i.e., the smallest subgraphs). More formally, let the base kernel k be a function counting pairs of matched vertices in the pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$

$$k(G_p, G_q) = k_{DB}^{(h)}(G_p, G_q) = \sum_{v_i \in V_p} \sum_{u_j \in V_q} \delta(v_i, u_j). \tag{8}$$

where

$$\delta(v_i, u_j) = \begin{cases} 1 \text{ if } C(i,j) = 1; \\ 0 \text{ otherwise.} \end{cases} \tag{9}$$

where δ is the Dirac kernel, that is, it is 1 if the arguments are equal and 0 otherwise (i.e. it is 1 if a pair of vertices are matched and 0 otherwise). Hence the proposed kernel function $k_{DB}^{(h)}$ is the sum of several positive definite Dirac kernels, and is thus **pd**. ■

The depth-based graph kernel $k_{DB}^{(h)}$ on a pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$ ($|V_p| = |V_q| = n$) requires time complexity $O(hn^3)$. For the pair of graphs $G_p(V_p, E_p)$ and $G_q(V_q, E_q)$, computing their correspondence matrix C in terms of h-layer depth-based representations requires time complexity $O(hn^3)$, and counting the number of matched vertex pairs from the matrix C needs to enumerate all the n^2 pairs of elements in C. Hence, the whole time complexity of the depth-based graph kernel $k_{DB}^{(h)}$ is $O(hn^3)$. This indicates that our depth-based graph kernel $k_{DB}^{(h)}$ can be computed in polynomial time. Key to this efficiency is that the required h-layer depth-based representations and the corresponding matching can be efficiently computed.

Discussions. The depth-based graph kernel is related to the depth-based representation defined in [13]. However, there are two significant differences. First, the depth-based representation in [13] is computed by measuring the complexities of subgraphs from the centroid vertex, which is identified by evaluating the minimum shortest path length variance to the remaining vertices. By contrast, we compute the h-layer depth-based representation for each vertex as a point coordinate. Second,the depth-based representation from the centroid vertex can be seen as an embedding vector. 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 depth-based graph kernel computed by matching the h-layer depth-based representation characterizes graphs in a high dimensional space and thus better preserves graph structure.

4 Experimental Results

4.1 Graph Datasets from the SHREC 3D Shape and COIL Image Databases

We demonstrate the performance of our kernel on several standard graph datasets from computer vision databases (i.e., the SHREC 3D Shape and COIL image databases).

a) BAR31, b) BSPHERE31 and c) GEOD31: The SHREC 3D Shape database consists of 15 classes and 20 individuals per class, that is 300 shapes [20]. This is a 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.

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

Table 1. Classification Accuracy (In $\%$ $\pm$ Standard Error) Using C-SVM and Runtime

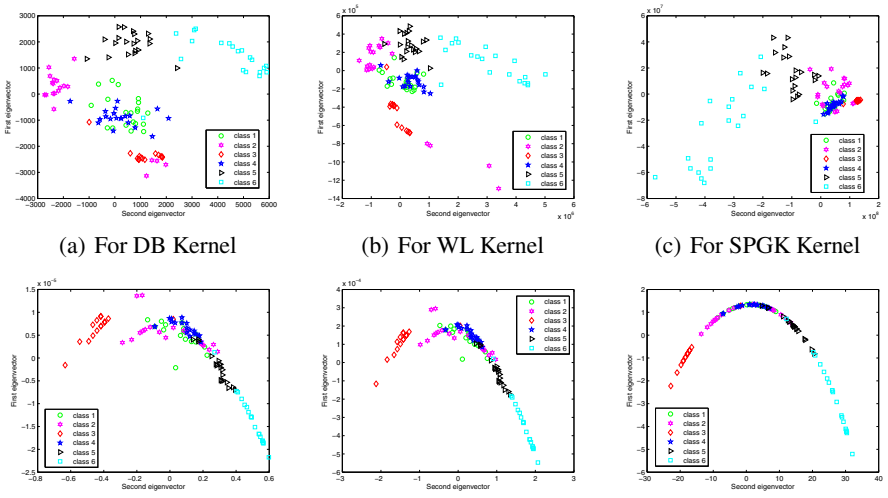
Datasets	DB	WL	SPGK	GCGK	GCGK4	JSK
BAR31	**69.40** $\pm$ .56	58.53 $\pm$ .53	55.73 $\pm$ .44	22.96 $\pm$ .65	23.40 $\pm$ .60	24.10 $\pm$ .86
BSPHERE31	**56.43** $\pm$ .69	42.10 $\pm$ .68	48.20 $\pm$ .76	17.10 $\pm$ .60	18.80 $\pm$ .50	21.76 $\pm$ .53
GEOD31	**42.83** $\pm$ .50	38.20 $\pm$ .68	38.40 $\pm$ .65	15.30 $\pm$ .68	22.36 $\pm$ .55	18.93 $\pm$ .50
COIL5	**74.22** $\pm$ .41	33.16 $\pm$ 1.01	69.97 $\pm$ .92	67.00 $\pm$ .55	68.77 $\pm$ .56	57.25 $\pm$ .46

Experimental Setup: a) First, we evaluate the performance of our depth-based graph kernel (DB) on graph classification problems. We also compare our kernel with several alternative state of the art graph kernels. These graph kernels include 1) the Weisfeiler-Lehman subtree kernel (WL) [3], 2) the shortest path graph kernel (SPGK) [7], 3) the graphlet count graph kernel with graphlets of size 3 (GCGK) and size 4 (GCGK4) [8], and 4) the Jensen-Shannon graph kernel (JSK) with the von Neumann entropy (i.e., the approximated von Neumann entropy [22] computed through the vertex degree) [21]. For our DB kernel, we set h as 10. For the WL kernel, we set the highest dimension (i.e. the highest height of subtrees) of the Weisfeiler-Lehman isomorphism as 10. For each kernel, we compute the kernel matrix on each graph dataset. We perform 10-fold cross-validation using the C-Support Vector Machine (C-SVM) Classification to compute the classification accuracies, 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 repeat the experiment 10 times. We report the average classification accuracies and standard errors for each kernel in Table.1. **b)** Second, we evaluate the performance of different kernels on graph clustering problems. We commence by performing the kernel Principle Component Analysis (kPCA) on the kernel matrix to embed graphs into a 2-dimensional principal space. We visualize the embedding results of each kernel using the first two principal components. The embedding results on the BAR31 and COIL5 datasets are shown in Fig.1 and Fig.2 respectively. Note that, for the BAR31 dataset we only visualize the embedding points of the first six classes of graphs. For each kernel, the embedding results on the BSPHERE31 and GEOD31 datasets are similar to that on the BAR31 dataset. The space in the paper is also not sufficient to include all of the results obtained. Thus, we only show the results on the BAR31 dataset. Finally, to place our analysis of graph clustering on a more quantitative footing, for

each kernel we apply the K-means method to all the kernel embeddings. We calculate the Rand Index for the resulting clusters. The Rand indicating each kernel is listed in Table 2.

4.2 Experiments on Graph Datasets

Experimental Results and Discussions: a) In terms of the classification accuracies, we observe that the accuracies of our DB kernel are the greatest for any dataset. The performance of our kernel exceeds that of all other kernels. The reason for its effectiveness is that the required depth-based representations of graphs used in our framework capture a high dimensional depth-based complexity information of graphs. In contrast, the alternative graph kernels with limited sized substructures (including the vertex degree required for the JSK) only capture local topological information and reflects restricted characteristics of graphs. Moreover, we also observe that the classification performance of our kernel is more stable than that of the alternative kernels. This verifies again that our kernel defined by depth-based matching reflects precise similarities of graphs. **b)** In terms of the embedding results, it is clear that our DB kernel produces the best clusters. The different classes are separated better than other kernels on any dataset. Note that, for the COIL5 dataset the 72 images for each object are taken from different viewing directions spaced at intervals of 5° around the object. Hence, the embedded graphs for each class are expected to form a circular trajectory rather than a cluster in the feature space. In the light of this observation, our method shows a greater representational power in terms of giving a more trajectory-like embedding than the alternative methods. Moreover, Table 2 indicates that our DB kernel outperforms all the alternative kernels for all the object classes studied on any dataset. These observations verify that our proposed kernel has good ability to distinguish different classes of graphs.



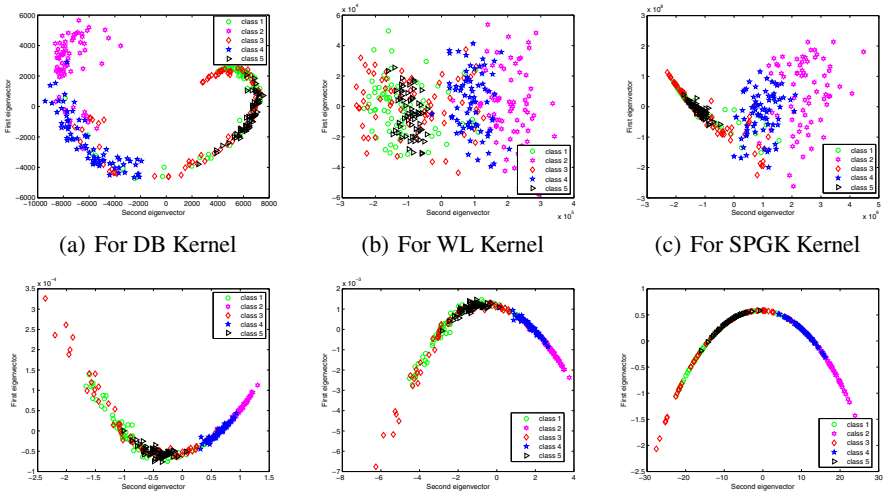
(a) For DB Kernel (b) For WL Kernel (c) For SPGK Kernel

(d) For GCGK Kernel (e) For GCGK4 Kernel (f) For JSK Kernel

Fig. 1. Clusters of Graphs from the BAR31 Dataset

Table 2. Rand Index for K-means Method

Datasets	DB	WL	SPGK	GCGK	GCGK4	JSK
BAR31	**0.2319**	0.2047	0.1734	0.1638	0.1641	0.1697
BSPHERE31	**0.1615**	0.1304	0.1582	0.1210	0.1238	0.1202
GEOD31	**0.1502**	0.1136	0.1142	0.1002	0.1207	0.1025
COIL5	**0.4436**	0.3503	0.4124	0.4119	0.4272	0.3295



(a) For DB Kernel (b) For WL Kernel (c) For SPGK Kernel

(d) For GCGK Kernel (e) For GCGK4 Kernel (f) For JSK Kernel

Fig. 2. Clusters of Graphs from the COIL5 Dataset

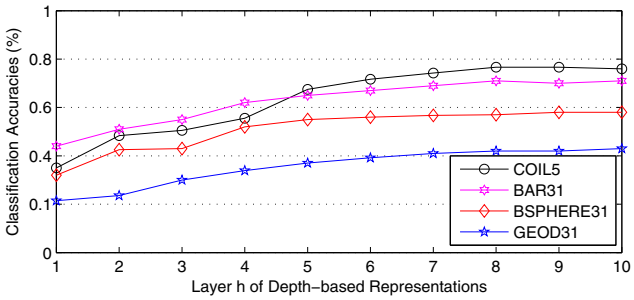


Fig. 3. The Accuracy with Different h Layer

Comparisons with Increasing h: To take our study one step further, we evaluate the performance of our DB graph kernel on graph datasets with increasing h. Here, we evaluate how the classification accuracies vary with increasing h (i.e. $h = 1, 2, \ldots, 10$). We report the results in Fig.3, in which the x-axis gives the varying of h, and the y-axis gives the classification accuracies of our DB kernel. The lines of different colours represent the results on different datasets. The classification accuracies tend to become greater with increasing h. This is because the greater the h, the higher dimensional depth-based complexity information of our kernel can be captured.

5 Conclusion

In this paper, we have described how to construct a depth-based graph kernel in terms of matching graphs based on the depth-based representations. The depth-based representations for graphs capture a high dimensional depth-based complexity information of graphs. Furthermore, our matching strategy incorporates structural correspondence into the kernel. We have empirically demonstrated the effectiveness and efficiency of our new kernel on synthetic graphs and real-world graphs abstracted from computer vision datasets.

References

1. Harchaoui, Z., Bach, F.: Image classification with segmentation graph kernels. In: Proc. CVPR (2007)
2. Barra, V., Biasotti, S.: 3D shape retrieval using Kernels on Extended Reeb Graphs. Pattern Recognition 46, 2985–2999 (2013)
3. Shervashidze, N., Schweitzer, P., Leeuwen, E.J., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-Lehman graph kernels. Journal of Machine Learning Research 1, 1–48 (2010)
4. Gärtner, T., Flach, P.A., Wrobel, S.: On graph kernels: Hardness results and efficient alternatives. In: Schölkopf, B., Warmuth, M.K. (eds.) COLT/Kernel 2003. LNCS (LNAI), vol. 2777, pp. 129–143. Springer, Heidelberg (2003)
5. Jebara, T., Kondor, R.I., Howard, A.: Probability product kernels. Journal of Machine Learning Research 5, 819–844 (2004)
6. Haussler, D.: Convolution kernels on discrete structures. In Technical Report UCS-CRL-99-10, UC Santa Cruz (1999)
7. Borgwardt, K.M., Kriegel, H.P.: Shortest-path kernels on graphs. In: Proc. ICDM, pp. 74–81 (2005)
8. Shervashidze, N., Vishwanathan, S.V.N., Petri, T., Mehlhorn, K., Borgwardt, K.M.: Efficient graphlet kernels for large graph comparison. Journal of Machine Learning Research 5, 488–495 (2009)
9. Costa, F., Grave, K.D.: Fast neighborhood subgraph pairwise distance kernel. In: Proc. ICML, pp. 255–262 (2010)
10. Aziz, F., Wilson, R.C., Hancock, E.R.: Backtrackless walks on a graph. IEEE Trans. Neural Netw. Learning Syst. 24(6), 977–989 (2013)
11. Ren, P., Wilson, R.C., Hancock, E.R.: Graph characterization via Ihara coefficients. IEEE Transactions on Neural Networks 22(2), 233–245 (2011)
12. Escolano, F., Hancock, E.R., Lozano, M.A.: Heat diffusion: Thermodynamic depth complexity of networks. Physical Review E 85, 036206 (2012)
13. Bai, L., Hancock, E.R.: Depth-based complexity traces of graphs. Pattern Recognition 47(3), 1172–1186 (2014)
14. Dehmer, M., Mowshowitz, A.: A history of graph entropy measures. Information Sciences 181, 57–78 (2011)
15. Anand, K., Bianconi, G., Severini, S.: Shannon and von Neumann entropy of random networks with heterogeneous expected degree. Physical Review E 83, 036109 (2011)
16. Scott, G.L., Longuett-Higgins, H.C.: An algorithm fro associating the features of two images. Proc. the Royal Society of London B 244, 313–320 (1991)
17. Xiao, B., Hancock, E.R., Wilson, R.C.: A generative model for graph matching and embedding. Computer Vision and Image Understanding 113(7), 777–789 (2009)

18. Munkres, J.: Algorithms for the assignment and transportation Problems. Journal of the Society for Industrial and Applied Mathematics 5(1), 32–38 (1957)
19. Borgwardts, K.M.: Graph Kernels. PhD thesis, Munchen (2007)
20. 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)
21. Bai, L., Hancock, E.R.: Graph kernels from the Jensen-Shannon divergence. Journal of Mathematical Imaging and Vision 47(1-2), 60–69 (2013)
22. Han, L., Escolano, F., Hancock, E.R., Wilson, R.C.: Graph characterizations from von Neumann entropy. Pattern Recognition Letters 33(15), 1958–1967 (2012)
23. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines (2011), Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`