A Mixed Weisfeiler-Lehman Graph Kernel

Lixiang Xu[1,2,4], Jin Xie[2,4], Xiaofeng Wang[3], and Bin Luo[1(✉)]

[1] School of Computer Science and Technology
Anhui University, Hefei 230601, Anhui, People's Republic of China
{xulixianghf,luobinahu}@163.com
[2] Department of Mathmatics & Physics
Hefei University, Hefei 230601, Anhui, People's Republic of China
hfuuxiejin@126.com
[3] Department of Computer Science and Technology
Hefei University, Hefei 230601, Anhui, People's Republic of China
xfwang@iim.ac.cn
[4] Institute of Scientific Computing
Hefei University, Hefei 230601, Anhui, People's Republic of China
hfuuxiejin@126.com

Abstract. Using concepts from the Weisfeiler-Lehman (WL) test of isomorphism, we propose a mixed WL graph kernel (MWLGK) framework based on a family of efficient WL graph kernels for constructing mixed graph kernel. This family of kernels can be defined based on the WL sequence of graphs. We apply the MWLGK framework on WL graph sequence taking into account the structural information which was overlooked. Our MWLGK is competitive with or outperforms the corresponding single WL graph kernel on several classification benchmark data sets.

Keywords: graph kernel · Graph classification · Weisfeiler-Lehman algorithm · Mixed graph kernel

1 Introduction

Machine learning in domains such as bioinformatics, drug discovery, and web data mining involves the study of relationships between objects. Graphs are natural data structures to model such relations, with nodes representing objects and edges the relationships between them. Simple ways of comparing graphs are based on pairwise comparison of nodes or edges, and are possible in quadratic time. There exist many other graph similarity measures based on graph isomorphism or related concepts such as subgraph isomorphism or the largest common subgraph. Kernel methods [14] offer a natural framework to study graph comparison. The graph kernels were originally proposed by Gärtner et al. [9] [11], where the structural information of graphs were taken into account. They work by counting the number of common random walks between two graphs. Later, many different graph kernels have been defined, which focus on different types of substructures in graphs, such as random walks [9] [11], shortest path kernel [4],

C.-L. Liu et al. (Eds.): GbRPR 2015, LNCS 9069, pp. 242–251, 2015.
DOI: 10.1007/978-3-319-18224-7_24

subtrees kernel [13], edge kernel [17], graphlet kernels [15], Weisfeiler-Lehman graph kernels [16], and quantum Jensen-Shannon graph kernel [3].

Several studies have recently shown that these graph kernels can achieve results competitive with the state of the art on benchmark data sets from bioinformatics and chemistry. For example, Borgwardt et al. [5] presented a graph model for proteins and defined a protein graph kernel that measures similarity between these graphs, and successfully tested the performance of this classifier on two function prediction tasks. The next year, they [6] extended common concepts from linear algebra to Reproducing Kernel Hilbert Spaces, and used these extensions to define a unifying framework for random walk kernels. Moreover, Vishwanathan et al. [17] presented a unified framework to study graph kernels, special cases of which included the random walk and marginalized graph kernels [10] [12]. Through reduction to a Sylvester equation they improved the time complexity of kernel computation between unlabeled graphs with n vertices. However, aforementioned state-of-the-art graph kernels did not scale to large graphs with hundreds of nodes and thousands of edges. So Shervashidze et al. [15] proposed efficient graph kernels based on counting or sampling limited size subgraphs in a graph. To consider higher order relations in the neighborhood to iteratively compute a kernel matrix, Camps-Valls et al. [12] presented a graph kernel for spatio-spectral remote sensing image classification with support vector machines. Another algebraic approach to graph kernels was appeared in [8], it showed how to represent scenes as graphs that encode models and their semantic relationships. To deal with high dimensional data and measure the mutual information between pairs of graphs, Bai et al. [2] showed how to construct Jensen Shannon kernels for graph data sets using the von-Neumann entropy and Shannon entropy.

However, aforementioned graph kernel methods rarely evolved to mixture of graph kernels. The mixed graph kernel can enhance classification accuracy as compared to single graph kernel that takes into account one aspect or part of the structural information only. Furthermore, the mixed graph kernel may flexibly capture structure information among the subtree, edge and shortest path in graph data sets in the classification.

The main contributions of this paper are as follows. Firstly, using concepts from the WL test of isomorphism, we propose a MWLGK framework based on a family of efficient WL graph kernels. This family of kernels is defined based on the WL sequence of graphs. Secondly, we apply the MWLGK framework on WL graph sequence taking into account the structural information which was overlooked. Thirdly, key to our approach is the WL test of isomorphism and mixed graph kernel, which allow us to compute a sequence of graphs which capture the topological and label information of the original graph. It may flexibly capture structure information among the subtree, edge and shortest path in graph data sets in the classification tasks.

The remainder of this article is structured as follows. In Section 2, we describe the related works, including the WL graph sequence and the general graph kernel framework based on them. In Section 3, we first describe three instances of WL

graph kernels. We then propose a mixed graph kernel framework based on a family of three instances of efficient WL graph kernels for graphs. In Section 4, we compare MWLGK to these single WL graph kernels. We report results on kernel computation efficiency and classification accuracy on graph benchmark data sets. Section 5 summarizes conclusion and future work.

2 Related Works

In order to know the principle of mixed WL graph kernel, we must make clear the concepts below. In this section, we define the WL graph sequence and the general graph kernel framework based on them. The content of this section refers mainly to the work in [16]. Our graph kernels use concepts from the WL test of isomorphism [18]. Assume we are given two graphs G and G' and we would like to test whether they are isomorphic. The 1-dimensional WL test is an iterative algorithm described in [16]. The key idea of the algorithm is to augment the node labels by the sorted set of node labels of neighbouring nodes, and compress these augmented labels into new, short labels. These steps are then repeated until the node label sets of G and G' differ, or the number of iterations reaches n. In each iteration i of the WL algorithm (see Algorithm 1 [16]), we get a new labeling for all nodes v. Recall that this labeling is concordant in G and G' meaning that if nodes in G and G' have identical multi-set labels, and only in this case, they will get identical new labels. Therefore, we can imagine that one iteration of WL relabeling as a function $r(V, E, l_i) = (V, E, l_{i+1})$ that transforms all graphs in the same manner. Note that r depends on the set of graphs that we consider. The 1-dimensional WL algorithm has been shown to be a valid isomorphism test for almost all graphs [1].

Definition 1. Define the WL graph at height i of the graph $G = (V, E, l) = G = (V, E, l_0)$ as the graph $G_i = (V, E, l_i)$. We call the sequence of WL graphs

$$\{G_0, G_1, ..., G_h\} = \{(V, E, l_0), (V, E, l_1), ..., (V, E, l_h)\}, \tag{1}$$

where $G_0 = G$ and $l_0 = l$, the WL sequence up to height h of G. G_0 is the original graph, $G_1 = r(G_0)$ is the graph resulting from the first relabeling, and so on. Note that neither V, nor E ever change in this sequence, but we define it as a sequence of graphs rather than a sequence of labeling functions for the sake of clarity of definitions that follow.

Definition 2. Let k be any kernel for graphs, that we will call the base kernel. Then the WL graph kernel with h iterations with the base kernel k is defined as

$$k_{WL}^{(h)}(G, G') = k(G_0, G'_0) + k(G_1, G'_1) + ... + k(G_h, G'_h), \tag{2}$$

where h is the number of WL iterations and $\{G_0, G_1, ..., G_h\}$ and $\{G'_0, G'_1, ..., G'_h\}$ are the WL sequences of G and G', respectively.

3 The Mixed Graph Kernels Framework

In this section, we first present three instances of WL graph kernels, the WL Subtree Kernel (WLSK), the WL Edge Kernel (WLEK) and the WL Shortest Path Kernel (WLSPK).We then propose a mixed graph kernels framework based on a family of three instances of efficient WL graph kernels for graphs.

3.1 The WL Subtree Kernel

In this section we present the WL subtree kernel, which is a natural instance of Definition 2.

Definition 3. Let G and G' be graphs. Define $\sum_i \subseteq \sum$ as the set of letters that occur as node labels at least once in G or G' at the end of the i-th iteration of the WL algorithm. Let $\sum_0$ be the set of original node labels of G and G'. Assume all $\sum_i$ are pairwise disjoint. Without loss of generality, assume that every $\sum_i = \{\sigma_{i1}, \sigma_{i2}, ..., \sigma_{i|\sum_i|}\}$ is ordered. Define a map $c_i : \{G, G'\} \times \sum_i \to N$ such that $c_i(G, \sigma_{ij})$ is the number of occurrences of the letter σ_{ij} in the graph G.

The WL subtree kernel on two graphs G and G' with h iterations is defined as:

$$k^{(h)}_{WLsubtree}(G, G') = \langle \phi^{(h)}_{WLsubtree}(G), \phi^{(h)}_{WLsubtree}(G') \rangle, \tag{3}$$

where

$$\phi^{(h)}_{WLsubtree}(G) = (c_0(G, \sigma_{01}), ..., c_0(G, \sigma_{0|\sum_0|}), ..., c_h(G, \sigma_{h1}), ..., c_h(G, \sigma_{h|\sum_h|})), \tag{4}$$

and

$$\phi^{(h)}_{WLsubtree}(G') = (c_0(G', \sigma_{01}), ..., c_0(G', \sigma_{0|\sum_0|}), ..., c_h(G', \sigma_{h1}), ..., c_h(G', \sigma_{h|\sum_h|})). \tag{5}$$

That is, the WL subtree kernel counts common original and compressed labels in two graphs.

The following Lemma 1 shows that (3) is indeed a special case of the general WL graph kernel (2).

Lemma 1. Let the base kernel k be a function counting pairs of matching node labels in two graphs:

$$k(G, G') = \sum_{v \in V} \sum_{v' \in V'} \delta(l(v), l(v')), \tag{6}$$

where δ is the Dirac kernel, that is, it is 1 when its arguments are equal and 0 otherwise. Then $k^{(h)}_{WL}(G, G') = k^{(h)}_{WLsubtree}(G, G')$ for all G, G'.

3.2 The WL Edge Kernel

The WL edge kernel is another instance of the WL graph kernel framework. In the case of graphs with unweighted edges, we consider the base kernel that counts with identically labeled endpoints in two graphs. In other words, the base kernel is defined as

$$k_E = \langle \phi_E(G), \phi_E(G') \rangle, \tag{7}$$

where $\phi_E(G)$ is a vector of numbers of occurrences of pairs $(a, b), a, b \in \sum$ which represent ordered labels of endpoints of an edge in G. Denoting (a, b) and (a', b') the ordered labels of endpoints of edges e and e' respectively, and σ the Dirac kernel, k_E can equivalently be expressed as $\sum_{e \in E} \sum_{e' \in E'} \delta(a, a')\delta(b, b')$. If the edges are weighted by a function w that assigns weights, the base kernel k_E can be defined as $k(G, G') = \sum_{e \in E} \sum_{e' \in E'} \delta(a, a')\delta(b, b')k_w(w(e), w(e'))$, where k_w is a kernel comparing edge weights. Following (2), we have

$$k^{(h)}_{WLedge}(G) = k_E(G_0, G'_0) + k_E(G_1, G'_1) + ... + k_E(G_h, G'_h). \tag{8}$$

3.3 The WL Shortest Path Kernel

Another example of the general WL graph kernels that we consider is the WL shortest path kernel. Here we use a node-labeled shortest path kernel as the base kernel [4]. In the particular case of graphs with unweighted edges, we consider the base kernel k_{sp} of the form

$$k_{SP}(G, G') = \langle \phi_{SP}(G), \phi_{SP}(G') \rangle, \tag{9}$$

where $\phi_{SP}(G)$ (resp. $\phi_{SP}(G')$) is a vector whose components are numbers of occurrences of triplets of the form (a, b, p) in G (resp. G'), where $a, b \in \sum$ are ordered endpoint labels of a shortest path and $p \in N_0$ is the shortest path length. According to (2), we have

$$k^{(h)}_{WLshortestpath}(G) = k_{SP}(G_0, G'_0) + k_{SP}(G_1, G'_1) + ... + k_{SP}(G_h, G'_h). \tag{10}$$

3.4 The Mixed WL Graph Kernel

In the following, we introduce the mixed WL graph kernel.

Mixed WL Graph Kernel. A graph G consists of a set of nodes (or vertices) V and edges E. In this article, n denotes the number of nodes in a graph and m the number of edges in a graph. Key to our exposition is the notion of a complement graph which we define below. Because each WL graph kernel can capture a different features. However, existing single WL graph kernels cannot take into account all features of graph data sets. Based on three instances of efficient WL graph kernels for graphs, we propose to modify them appropriately. In the following, using the concept of a complement graph, we present a unifying

framework which includes the above mentioned three instances of efficient WL graph kernels as special cases. We now define a novel graph kernel called the mixed graph kernel.

$$k_{comp}^{(h)}(G,G') = \alpha \cdot k_{WLsubtree}^{(h)}(G_h,G'_h) + \beta \cdot k_{WLedge}^{(h)}(G_h,G'_h) + \gamma \cdot k_{WLshortestpath}^{(h)}(G_h,G'_h), \quad (11)$$

where α, β, γ is weight, $\alpha + \beta + \gamma = 1$. Although this kernel seems simple minded at first, it is in fact rather useful. This simple kernel improves substantial gains in performance in our experiments comparing corresponding single WL graph kernel.

4 Experiments

Here we compare the performance of the WLSK, the WLEK and the WLSPK to the proposed Mixed WL Graph Kernel in terms of classification accuracy and MSE on graph benchmark data sets. These data sets include MUTAG, ENZYMES, PTC, NCI1 and NCI109. Some statistics concerning the data sets are given in Table 1. Our WL graph kernel matrix is computed from the description of Shervashidze [16].

MUTAG. The MUTAG benchmark is based on graphs representing 188 chemical compounds, and aims to predict whether each compound possesses mutagenicity. The maximum and average number of vertices are 28 and 17.93 respectively. As the vertices and edges of each compound are labeled with a real number, we transform these graphs into unweighted graphs.

ENZYMES. The ENZYMES data set is a data set based on graphs representing protein tertiary structures consisting of 600 enzymes from the BRENDA enzyme database. In this case the task is to correctly assign each enzyme to one of the 6 EC top-level classes. The maximum and average number of vertices are 126 and 32.63 respectively.

PTC. The Predictive Toxicology Challenge data set reports the carcinogenicity of several hundred chemical compounds for Male Mice (MM), Female Mice (FM), Male Rats (MR) and Female Rats (FR).

NCI1 and NCI109. The NCI1 and NCI109 data sets consist of graphs representing two balanced subsets of data sets of chemical compounds screened for activity against non-small cell lung cancer and ovarian cancer cell lines respectively. There are 4110 and 4127 graph based structures in NCI1 and NCI109 respectively. The maximum, minimum and average number of vertices in NCI1 and NCI109 are 111, 3 and 29.87, and 111, 4 and 29.68 respectively.

In the experiments, we perform 10-fold cross-validation of C-Support Vector Machine Classification using LIBSVM [7], using 9 folds for training and 1 for testing. All parameters of the SVM were optimised on the training data sets only. To exclude random effects of fold assignments, we repeat the whole experiment 20 times. We report average prediction accuracies and standard deviations in Tables 2.

Table 1. Information of the Graph based Data sets

Data Set	Size	Classes	Avg Nodes	Avg Edges
MATUG	188	2	17.9	39.5
ENZYME	600	6	32.6	124.2
PTC	344	2	25.5	51.9
NCI1	4110	2	29.8	64.6
NCI109	4127	2	29.6	62.2

Due to the large number of graph kernels in the literatures, we could not compare every graph kernel. But the representative instances of the major families of graph kernels were compared in [16]. On these data sets, we only compared our MWLGK to the corresponding WL graph kernels.

We chose h for our all the WL graph kernels and mixed WL graph kernel by cross-validation on the training data sets for $h = 0, 1, 2, 5, 10, 20$ in MUTAG and ENZYMES data sets, which means that we compute 6 different WL graph kernel matrices in each experiment. We report the total classification accuracy of these computations. It is worth mentioning that small values of h, such as 2 or 5, systematically can give the best results for all data sets used, and the accuracy obtains relatively stable value. When h equals 5, 10 or more larger, this kernel is more computationally expensive. However, While h equals 0 or 1, classification accuracy happen phenomenon of tottering and instability (see Fig. 1). So for the rest of the experimental data sets, we only need to choose $h = 2$.

In each experiment, Our MWLGK use two of the three WL graph kernels for reducing the complexity of the calculation in this paper (i.e., $k_{comp} = (1-\alpha)K_1 + \alpha K_2, (0 \leq \alpha \leq 1)$, where α is the mixed coefficient. K_1, K_2 denotes the WLSK or WLEK or WLSPK, respectively. If some instances of the MWLGK can't obtain better performance, they are not showed and discussed in this paper. In Fig. 2(a) we show the efficiency of mixing a WLSK and a WLEK on PTC, where K_1 denotes WLSK, K_2 denotes WLEK. The classification accuracy of the WLEK reaches higher accuracy than the WLSK. Fig. 2(a) shows that the mixed kernel

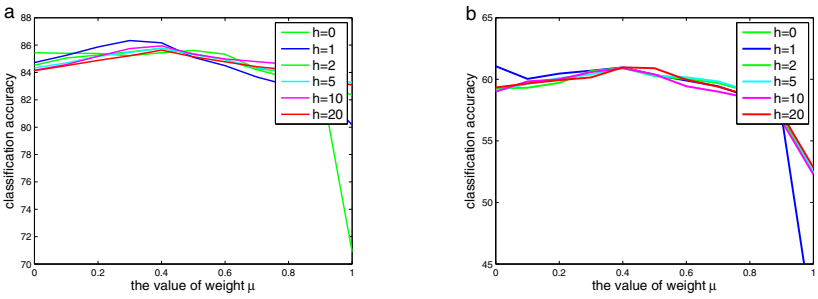

Fig. 1. a. The mixture of WLSPK and WLSK on graph data set MUTAG, b. The mixture of WLSPK and WLSK on graph data set ENZYMES.

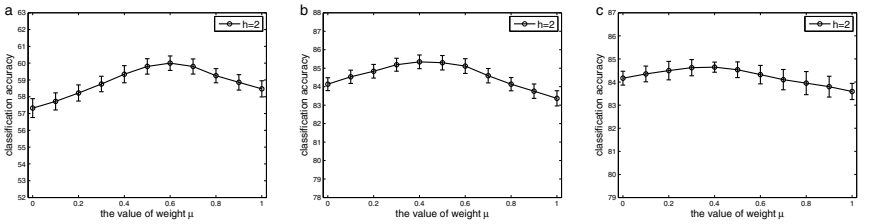

Fig. 2. a. The mixture performance of WLSK and WLEK on PTC, b. The mixture of WLEK and WLSPK on NCI1, c. The mixture of WLEK and WLSPK on NCI109. Error bars correspond to standard errors.

not only has a WLSK efficiency, but also a WLEK effect. By increasing the mixing coefficient, the influence of the WLEK is also increased. Furthermore, the results have the highest accuracy using the mixed weight α in the range [0.5, 0.7].

In Fig. 2(b) and Fig. 2(c) we show the efficiency of mixing a WLEK and a WLSPK on NCI1 and NCI109, where K_1 denotes WLEK, K_2 denotes WLSPK. The classification accuracy of the WLEK reaches higher accuracy than the WLSPK on two data sets. Fig. 2(b) and Fig. 2(c) show that the results have the highest accuracy using the mixed weight α in the range [0.3, 0.5].

Furthermore, from the Fig. 2, we obtain the influence of WLEK is stronger in WL graph kernels. At the same time, the highest accuracy of MWLGK is higher than the corresponding single WL graph kernel.

Our MWLGK is competitive with or outperform corresponding single WL graph kernel on several classification benchmark data sets. Even the MWLGK can reach the highest accuracy level on all data sets. In our experimental evaluation, our mixed WL graph kernel framework is an attempt to applications of the mixture of graph kernels in various disciplines such as computational biology and social network analysis. To summarize, the MWLGK turns out to be competitive in terms of classification accuracy on all data sets, and its accuracy levels are highest on all data sets in experiments.

Table 2. Prediction accuracy (± standard deviation) on graph classification benchmark data sets (- : the bad performance of the WL graph kernels)

Kernel	MUTAG	ENZYMES	PTC	NCI1	NCI109
WLSK	82.34±0.57	52.40±1.23	57.32±0.56	-	-
WLEK	-	-	58.47±0.48	84.13±0.35	84.16±0.29
WLSPK	84.53±0.49	59.26±1.17	-	83.36±0.41	83.59±0.37
MWLGK	85.76±0.47	60.93±1.09	60.00±0.43	85.34±0.37	84.64±0.34

5 Conclusions

We define a general mixed graph kernel framework for constructing graph kernel on graph sets with unlabeled or discretely labeled nodes. Instances of our framework base on three instances of WL graph kernels, the WLSK, the WLEK and the WLSPK. Our MWLGK is competitive in terms of accuracy with corresponding single WL graph kernel on several classification benchmark data sets. In the future, we will be to consider the MWLGK on graphs with continuous or high-dimensional node labels and their efficient computation.

Acknowledgement. This work was supported by National High Technology Research and Development Program (863 Program) of China under Grant (2014AA015104), National Nature Science Foundation of China (61472002), Key Project of Scientific Research, Education Department of Anhui Province (KJ2014ZD30), Key Construction Disciplines of Applied Mathematics of Hefei University (2014XK08), Anhui Provincial Natural Science Foundation (1308085MF84), Training Object Project for Academic Leader of Hefei University (2014dtr08), Natural Science Research Project of Colleges of Anhui Province (KJ2013B234), MOE Youth Project of Humanities and Social Sciences (14YJCZH169).

References

1. Babai, L., Kucera, L.: Canonical labelling of graphs in linear average time. In: 20th Annual Symposium on Foundations of Computer Science, pp. 39–46. IEEE (1979)
2. Bai, L., Hancock, E.R.: Graph clustering using the jensen-shannon kernel. In: Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W. (eds.) CAIP 2011, Part I. LNCS, vol. 6854, pp. 394–401. Springer, Heidelberg (2011)
3. Bai, L., Hancock, E.R., Torsello, A., Rossi, L.: A quantum jensen-shannon graph kernel using the continuous-time quantum walk. In: Kropatsch, W.G., Artner, N.M., Haxhimusa, Y., Jiang, X. (eds.) GbRPR 2013. LNCS, vol. 7877, pp. 121–131. Springer, Heidelberg (2013)
4. Borgwardt, K.M., Kriegel, H.-P.: Shortest-path kernels on graphs. In: Fifth IEEE International Conference on Data Mining, 8 p. IEEE (2005)
5. Borgwardt, K.M., Ong, C.S., Schönauer, S., Vishwanathan, S.V.N., Smola, A.J., Kriegel, H.-P.: Protein function prediction via graph kernels. Bioinformatics 21(suppl. 1), i47–i56 (2005)
6. Borgwardt, K.M., Schraudolph, N.N., Vishwanathan, S.V.N.: Fast computation of graph kernels. In: Advances in Neural Information Processing Systems, pp. 1449–1456 (2006)
7. Chang, C.-C., Lin, C.-J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3), 27 (2011)
8. Fisher, M., Savva, M., Hanrahan, P.: Characterizing structural relationships in scenes using graph kernels. ACM Transactions on Graphics (TOG) 30, 34 (2011)
9. Gärtner, T., Flach, P., 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)
10. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernels between labeled graphs. In: ICML, vol. 3, pp. 321–328 (2003)

11. Kashima, H., Tsuda, K., Inokuchi, A.: Kernels for graphs. Kernel Methods in Computational Biology 39(1), 101–113 (2004)
12. Mahé, P., Ueda, N., Akutsu, T., Perret, J.-L., Vert, J.-P.: Extensions of marginalized graph kernels. In: Proceedings of the Twenty-First International Conference on Machine Learning, p. 70. ACM (2004)
13. Ramon, J., Gärtner, T.: Expressivity versus efficiency of graph kernels. In: First International Workshop on Mining Graphs, Trees and Sequences, pp. 65–74. Citeseer (2003)
14. Schölkopf, B., Smola, A.J.: Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT Press (2002)
15. Shervashidze, N., Petri, T., Mehlhorn, K., Borgwardt, K.M., Vishwanathan, S.V.N.: Efficient graphlet kernels for large graph comparison. In: International Conference on Artificial Intelligence and Statistics, pp. 488–495 (2009)
16. Shervashidze, N., Schweitzer, P., Leeuwen, E.J.V., Mehlhorn, K., Borgwardt, K.M.: Weisfeiler-lehman graph kernels. The Journal of Machine Learning Research 12, 2539–2561 (2011)
17. Vichy, S., Vishwanathan, N., Schraudolph, N.N., Kondor, R., Borgwardt, K.M.: Graph kernels. The Journal of Machine Learning Research 11, 1201–1242 (2010)
18. Ju Weisfeiler, B., Leman, A.A.: Reduction of a graph to a canonical form and an algebra which appears in the process. NTI, Ser. 2(9), 12–16 (1968)