ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

Deep Weisfeiler-Lehman Assignment Kernels via Multiple Kernel Learning

Nils M. Kriege*

Department of Computer Science, TU Dortmund University
Otto-Hahn-Str. 14, 44227 Dortmund, Germany

Abstract. Kernels for structured data are commonly obtained by decomposing objects into their parts and adding up the similarities between all pairs of parts measured by a base kernel. Assignment kernels are based on an optimal bijection between the parts and have proven to be an effective alternative to the established convolution kernels. We explore how the base kernel can be learned as part of the classification problem. We build on the theory of valid assignment kernels derived from hierarchies defined on the parts. We show that the weights of this hierarchy can be optimized via multiple kernel learning. We apply this result to learn vertex similarities for the Weisfeiler-Lehman optimal assignment kernel for graph classification. We present first experimental results which demonstrate the feasibility and effectiveness of the approach.

1 Introduction

Graphs are a versatile concept used to represent structured data in many domains such as chem- and bioinformatics, or social network analysis. Graph kernels have become an established and widely-used technique for solving classification tasks on graphs. In the past 15 years a large number of graph kernels have been proposed. One of the most successful in practice is the *Weisfeiler-Lehman subtree kernel* [1], which is based on the color refinement heuristic for graph isomorphism testing. Recently, the *Weisfeiler-Lehman optimal assignment kernel* [2] was proposed, which also uses color refinement, but derives the kernel from an optimal bijection of the vertices instead of summing over all pairs of vertices. In classification experiments, this approach yields higher accuracy scores than the Weisfeiler-Lehman subtree kernel for many data sets. More recently, several deep learning approaches to graph classification based on neural networks have emerged. These methods construct a vector representation for each vertex by iteratively applying a neighborhood aggregation function with learned weights. Most of them fit into the *neural message passing framework* proposed in [3] and show promising results on several graph classification benchmarks [4]. Compared to these approaches, kernel methods are considered "shallow" since they do not learn a representation by means of weights organized in a hierarchical manner. This also is the case for the deep graph kernels proposed in [5], which support weights between graph features, but do not learn them end-to-end. However, there are kernel methods that can justifiably be described as deep [6]. This is the case for multiple kernel learning, which was, for example, used to combine base kernels organized in a hierarchy according to their level

*This work was supported by the German Science Foundation (DFG) within the SFB 876 "Providing Information by Resource-Constrained Data Analysis", project A6 "Resource-efficient Graph Mining".

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

of abstraction [7]. Moreover, it was used to alleviate the diagonal dominance problem of convolution kernels for graphs [8].

We propose deep assignment kernels for structured data which learn the base kernel on substructures as part of the training. To this end, we consider base kernels represented by a hierarchy for which the weights are obtained via multiple kernel learning. Building on this, we propose a deep method for graph classification based on the Weisfeiler-Lehman method. The feasibility of the approach is demonstrated experimentally.

2 Basic Techniques

We introduce the fundamentals and key techniques relevant for our contribution in the following. A *kernel* on a set $\mathcal{X}$ is a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that there is a real Hilbert space $\mathcal{H}$ and a mapping $\phi : \mathcal{X} \to \mathcal{H}$ with $k(x, y) = \langle \phi(x), \phi(y) \rangle$ for all x, y in $\mathcal{X}$, where $\langle \cdot, \cdot \rangle$ denotes the inner product of $\mathcal{H}$. We consider simple undirected graphs $G = (V, E)$, where $V(G) = V$ is the set of *vertices* and $E(G) = E$ the set of *edges*. An edge $\{u, v\}$ is for short denoted by uv or vu, where both refer to the same edge. A graph with a unique path between any two vertices is a *tree*. A *rooted tree* is a tree T with a distinguished vertex $r \in V(T)$ called *root*.

Optimal assignment kernels. A common approach to compare two graphs is to construct an assignment between their vertices that maximizes the structural overlap or agreement of vertex attributes. This principle was proposed to obtain graph kernels, where the similarity between two vertices is determined by a base kernel [9]. However, it was shown that the resulting similarity measure is not a valid kernel in general [10]. More recently, it was proven that for a specific class of base kernels, the similarity derived from optimal assignments is guaranteed to be a valid kernel [2].

Let $[\mathcal{X}]^n$ denote the set of all n-element subsets of a set $\mathcal{X}$ and $\mathfrak{B}(X, Y)$ the set of all bijections between X, Y in $[\mathcal{X}]^n$ for $n \in \mathbb{N}$. The *optimal assignment kernel* $K^k_{\mathfrak{B}}$ on $[\mathcal{X}]^n$ is defined as

$$K^k_{\mathfrak{B}}(X, Y) = \max_{B \in \mathfrak{B}(X,Y)} \sum_{(x,y) \in B} k(x, y), \tag{1}$$

where k is a *base kernel* on $\mathcal{X}$. For the application to sets of different cardinality, the smaller set can be augmented by dummy elements with no effect on the solution value. Given that the base kernel k satisfies the *strong kernel property*, i.e., $k(x, y) \geq \min\{k(x, z), k(z, y)\}$ for all $x, y, z \in \mathcal{X}$, the function $K^k_{\mathfrak{B}}$ is a valid kernel [2]. Strong kernels are equivalent to kernels obtained from a hierarchical partition of their domain. Formally, let T be a rooted tree such that the leaves of T are the elements of $\mathcal{X}$ and $\omega : V(T) \to \mathbb{R}_{\geq 0}$ a weight function. We refer to the tuple (T, ω) as a *hierarchy*. A hierarchy on $\mathcal{X}$ induces a similarity $k(x, y)$ for $x, y \in \mathcal{X}$ as follows. For $v \in V(T)$ let $P(v) \subseteq V(T)$ denote the vertices in T on the path from v to the root r. Then the similarity between $x, y \in \mathcal{X}$ is

$$k(x, y) = \sum_{v \in P(x) \cap P(y)} \omega(v).$$

For every strong kernel k there is a hierarchy that induces k and, vice versa, every hierarchy induces a strong kernel [2].

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

The optimal assignment kernel of Eq. (1) can be computed in linear time from the hierarchy (T, ω) of the base kernel k by histogram intersection as follows. For a node $v \in V(T)$ and a set $X \subseteq \mathcal{X}$, let X_v denote the subset of X that is contained in the subtree rooted at v. Then the optimal assignment kernel is

$$K^k_{\mathfrak{B}}(X, Y) = \sum_{v \in V(T)} \min\{|X_v|, |Y_v|\} \cdot \omega(v), \tag{2}$$

which can be seen as the histogram intersection kernel for appropriately defined histograms representing the sets X and Y under the strong base kernel k [2].

Weisfeiler-Lehman optimal assignment kernels. Color refinement, also known as 1-dimensional Weisfeiler-Lehman refinement or naïve vertex classification, is a classical heuristic for graph isomorphism testing. It iteratively refines partitions of the vertices of a graph, where the vertices in the same cell are said to have the same color. In each iteration two vertices with the same color obtain different new colors if their neighborhoods differ w.r.t. the current coloring. More recently, the approach was used to obtain kernels between graphs [1]. For the Weisfeiler-Lehman subtree kernel a graph is represented by a feature vector, where each component is associated with a color and counts the number of vertices in the graph having that color in one iteration. The Weisfeiler-Lehman subtree kernel is the dot product of such feature vectors.

We give a clear mathematical formulation of the procedure. Given a parameter h and a graph G with initial colors τ_0, a sequence $(\tau_1, \ldots, \tau_h)$ of refined colors is computed, where τ_i is obtained from τ_{i-1} by the following procedure. For every vertex $v \in V(G)$, sort the multiset of colors $\{\!\{\tau_{i-1}(u) : vu \in E(G)\}\!\}$ to obtain a unique sequence of colors and add $\tau_{i-1}(v)$ as first element. Assign a new color $\tau_i(v)$ to every vertex v by employing an injective mapping from color sequences to new colors. It was observed in [2] that color refinement applied to a set of graphs under the same injective mapping yields a hierarchy on the vertices. This hierarchy with a uniform weight function induces the strong base kernel

$$k(v, v') = \sum_{i=0}^{h} k_\delta(\tau_i(v), \tau_i(v')) \tag{3}$$

on the vertices, where k_δ denotes the Dirac kernel. This kernel measures the number of iteration required to assign different colors to the vertices and reflects the extent to which the vertices have a structurally similar neighborhood. The optimal assignment kernel with this base kernel is referred to as *Weisfeiler-Lehman optimal assignment kernel* and was shown to achieve better accuracy results in many classification experiments than the Weisfeiler-Lehman subtree kernel.

Multiple kernel learning. Multiple kernel learning refers to machine learning techniques that extend classical kernel-based support vector machines to use multiple (heterogeneous) kernels, which are combined using coefficients learned as part of the training. One such approach is EasyMKL, which is scalable to a large number of kernels and can be used to obtain a data-driven feature weighting [11]. EasyMKL combines the kernels k_i, $i \in \{1, \ldots, R\}$, to a kernel $k(x, y) = \sum_{i=1}^{R} \alpha_i k_i(x, y)$ by learning the coefficients

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

$\alpha_i \geq 0$. This is achieved by optimizing the problem

$$\max_{\boldsymbol{\alpha}:\|\boldsymbol{\alpha}\|=1} \min_{\boldsymbol{\gamma}\in\Gamma} (1-\lambda)\boldsymbol{\gamma}^\top \boldsymbol{Y} \left(\sum_{i=0}^{R} \alpha_i \boldsymbol{K}_i \right) \boldsymbol{Y}\boldsymbol{\gamma} + \lambda \|\boldsymbol{\gamma}\|_2^2 , \tag{4}$$

where $\boldsymbol{K}_i$ is the $l \times l$ kernel matrix obtained by applying k_i on the training set of cardinality l, $\boldsymbol{Y}$ the diagonal matrix with $y_{i,i}$ the class label of the ith training example and λ a hyperparameter for regularization. The set Γ is the domain of probability distributions $\boldsymbol{\gamma} \in \mathbb{R}^l_{\geq 0}$ defined over the sets of positive and negative training examples.

3 Deep Assignment Kernels

We investigate how the base kernel used to compare the parts can be learned as part of the classification problem when comparing structured objects with an assignment kernel. There are several basic difficulties with this general approach. Since we derive a similarity measure from a combinatorial problem, it is not clear how changing the base kernel will effect the value of the optimal assignment. In particular, small changes in the base kernel value may lead to entirely different optimal assignments. Solving a single assignment problem takes cubic time in general, which is not feasible for large instances. Moreover, the value of assignments does not yield a valid kernel in general.

Therefore, we consider a highly restricted class of base kernels. We learn the base kernel from the class of kernels that are induced by the same tree T. Every base kernel in this class is uniquely defined by T and the weight function ω, which we would like to learn as part of the training. Following the results summarized in Sec. 2 and considering Eq. (2) it becomes apparent that this can be achieved by multiple kernel learning. For every node $v \in V(T)$ we consider the kernel $k_v(X, Y) = \min\{|X_v|, |Y_v|\}$. Solving Eq. (4) yields coefficients α_v with $\|\boldsymbol{\alpha}\| = 1$ that can be interpreted as learned weights $\omega(v) = \alpha_v$ for the tree T to form a hierarchy. In this way, we hope to obtain strong kernels that are adaptive to the specific learning task. However, the adaption is possible to a limited extent only: The tree T determines a set of optimal solutions to the assignment problem. These solution will remain optimal under all learned weight functions, though their value may change. In case of $\alpha_v = 0$ an equivalent hierarchy is obtained by removing the node v from the tree and attaching its children to the parent of v. In this case the set of optimal solutions may be a superset of the optimal solutions obtained for the tree with uniform weights.

Deep Weisfeiler-Lehman assignment kernels. We apply the observations stated above to the Weisfeiler-Lehman optimal assignment kernel, which is based on the hierarchy generated by color refinement as detailed in Sec. 2. The hierarchy induces a similarity between the vertices, such that with each level the extent of the considered neighborhood increases. In its original version uniform weights where used that induce the vertex similarity stated in Eq. (3). Introducing weights as above, we obtain the vertex similarity

$$k(v, v') = \sum_{i=0}^{h} \omega(\tau_i(v)) \cdot k_\delta(\tau_i(v), \tau_i(v')) \tag{5}$$

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0.
Available from http://www.i6doc.com/en/.

as base kernel, where ω is a weight function for the colors of the refinement process. We refer to the assignment kernel with this base kernel as *Deep Weisfeiler-Lehman assignment kernel*. Please note that the value of the parameter h is typically determined by a grid search in a costly cross-validation process. Our approach is less dependent on the choice of this parameter since too specific features are down weighted automatically.

Feature and weight grouping. Depending on the strong kernel and the data set, the representing hierarchy and therefore also the number of weights may be very large. This is, for example, the case for the Deep Weisfeiler-Lehman assignment kernel, since the color refinement process effectively distinguishes vertices with different neighborhoods. A very large number of weights slows down the training and may lead to overfitting. To control the number of weights, we group the nodes of the hierarchy using a clustering algorithm. To this end, we represent each node v in the hierarchy by the data point $(c_1^v, c_2^v, \ldots, c_n^v)$, where n is the size of the data set and c_i^v is the number of elements of the $i$the object in the data set that are contained in the subtree rooted at v. We apply k-means clustering to these vectors and, finally, assign weights to each cluster. Therefore, all nodes in the same cluster share the same weight learned by the MKL algorithm.

4 Experimental Evaluation

We performed classification experiments using the C-SVM implementation LIBSVM and the EasyMKL implementation of MKLpy v0.2.1b0.[1] We report average prediction accuracies and standard deviations obtained by 5-fold cross-validation repeated 5 times with random fold assignment. Within each fold all hyperparameters were selected by cross-validation based on the training set. The regularization parameter C was selected from $\{0.01, 0.1, 1, 10, 100\}$ and the parameter λ from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$.

We compare the Weisfeiler-Lehman subtree kernel (WL), the Weisfeiler-Lehman optimal assignment kernel (WL-OA) and its deep variant without feature grouping (DWL-OA1) and with feature grouping (DWL-OA2), where the number of clusters was set to $k = 10$. We set the number of color refinement iterations to $h = 4$ for all experiments. We tested on widely-used graph classification benchmarks from different domains [12]. MUTAG, PTC-MR, NCI1 and NCI109 are graphs derived from small molecules, PROTEINS and D&D represent macromolecules, and REDDIT contains social network graphs. All data sets consist of two classes, the vertex and edge labels were ignored, if present.

We were not able to run DWL-OA1 on the large date sets with more than thousand objects due to memory constraints. All results are summarized in Table 1. We observed only minor differences in accuracy between the three Weisfeiler-Lehman optimal assignment kernels. DWL-OA2 performs better than DWL-OA1 which indicates the benefit of feature grouping. For the considered data sets, DWL-OA has obtained state-of-the-art accuracy results, but there is no clear evidence that learning weights via MKL improves the classification accuracy significantly. However, we have observed that a significant proportion of the learned weights is zero, which leads to compact sparse models.

[1] `https://pypi.org/project/MKLpy/`

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

Table 1: Classification accuracies and standard deviations on graph data sets representing small molecules, macromolecules and social networks.

Kernel	**Data Set**						
	MUTAG	PTC-MR	NCI1	NCI109	PROTEINS	D&D	REDDIT
WL	88.3±1.2	54.5±2.5	78.9±0.6	79.6±0.3	70.6±0.5	73.2±0.4	71.7±0.3
WL-OA	88.6±1.0	55.8±2.3	78.6±0.3	78.8±0.6	73.8±0.4	75.7±0.3	88.5±0.3
DWL-OA1	88.1±1.0	56.7±2.3	OOM	OOM	OOM	OOM	OOM
DWL-OA2	88.5±0.9	57.6±1.3	78.9±0.2	78.5±0.2	72.5±0.4	76.7±0.3	86.9±0.2

5 Conclusion

We have proposed Weisfeiler-Lehman assignment kernels which learn deep representations for graph classification. It remains future work to analyze the learned weights and their domain-specific meaning in detail. We believe that the interpretability of the weights in terms of vertex neighborhoods is a strength of the approach and can give new insights into real-world problems. Our method only allows to learn vertex similarities from a predefined restricted class of functions. In the future, we would like to study more general approaches such as learning the entire hierarchy and not just its weights.

References

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

[2] Nils M. Kriege, Pierre-Louis Giscard, and Richard Wilson. On valid optimal assignment kernels and applications to graph classification. In *Advances in Neural Information Processing Systems*, pages 1623–1631. Curran Associates, Inc., 2016.

[3] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *33rd International Conference on Machine Learning*, 2017.

[4] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in Neural Information Processing Systems*, 2018.

[5] Pinar Yanardag and S.V.N. Vishwanathan. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1365–1374. ACM, 2015.

[6] John Shawe-Taylor. Deep-er kernels. In *ICPRAM 2014 - Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods*, pages IS–9, 2014.

[7] Michele Donini and Fabio Aiolli. Learning deep kernels in the space of dot product polynomials. *Machine Learning*, pages 1–25, 2016.

[8] F. Aiolli, M. Donini, N. Navarin, and A. Sperduti. Multiple graph-kernel learning. In *2015 IEEE Symposium Series on Computational Intelligence*, pages 1607–1614, Dec 2015.

[9] Holger Fröhlich, Jörg K. Wegner, Florian Sieker, and Andreas Zell. Optimal assignment kernels for attributed molecular graphs. In *Proceedings of the 22nd International Conference on Machine learning*, pages 225–232, New York, NY, USA, 2005. ACM.

[10] Jean-Philippe Vert. The optimal assignment kernel is not positive definite. *CoRR*, abs/0801.4061, 2008.

[11] Fabio Aiolli and Michele Donini. EasyMKL: a scalable multiple kernel learning algorithm. *Neurocomputing*, 169:215 – 224, 2015.

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