A Persistent Weisfeiler-Lehman Procedure for Graph Classification

Bastian Rieck[*,†], Christian Bock[*,†], Karsten Borgwardt[*]
[*]D-BSSE, Machine Learning and Computational Biology Lab, ETH Zurich, Switzerland; [†]Equal Contributions

Introduction

Many graph classification schemes are based on the ideas of *neighbourhood aggregation*. One of the classical approaches, the Weisfeiler-Lehman (WL) kernel, still exhibits favourable and competitive classification performance values. With the increased use of graph neural networks, which are based on aggregation schemes as well, the concepts behind WL have seen renewed interest, and several new variants have been proposed.

Yet, most of the existing graph classification approaches do not sufficiently make use of its topological information. We consider topological information to represent a description of the connectivity of a graph in terms of its connected components, cycles, and cliques. This paper presents an approach that combines the WL aggregation scheme with concepts from *topological data analysis* (TDA), namely persistent homology. This makes it possible to integrate information about both the connected components and cycles into the calculation, which has a positive impact on classification performance. Our new method is efficient and we evaluate it on several benchmark classification data sets, where it exhibits favourable performance when compared with other state-of-the-art algorithms for graph classification.

Background

Persistent Homology

Given a weighted graph G, we capture its topological features on multiple scales by means of a *weight-based filtration*. A filtration is a nested sequence of subgraphs of G that represent its "growth":

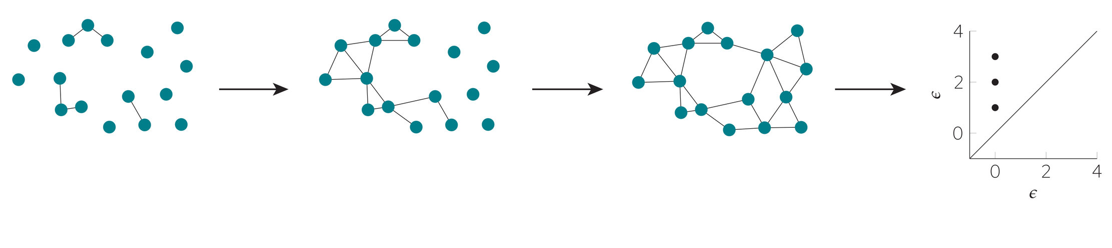
$$\emptyset \subseteq G_0 \subseteq G_1 \subseteq \cdots \subseteq G_{k-1} \subseteq G_k = G$$

During this growth process, topological features can be created (a new connected component is created by inserting a vertex), or destroyed (two connected components are joined into one).

The tuples of creation and destruction values of a node are captured as points (ϵ_c, ϵ_d) in a persistence diagram D. The *persistence*, i.e. the prominence, of a given topological feature is then calculated as:

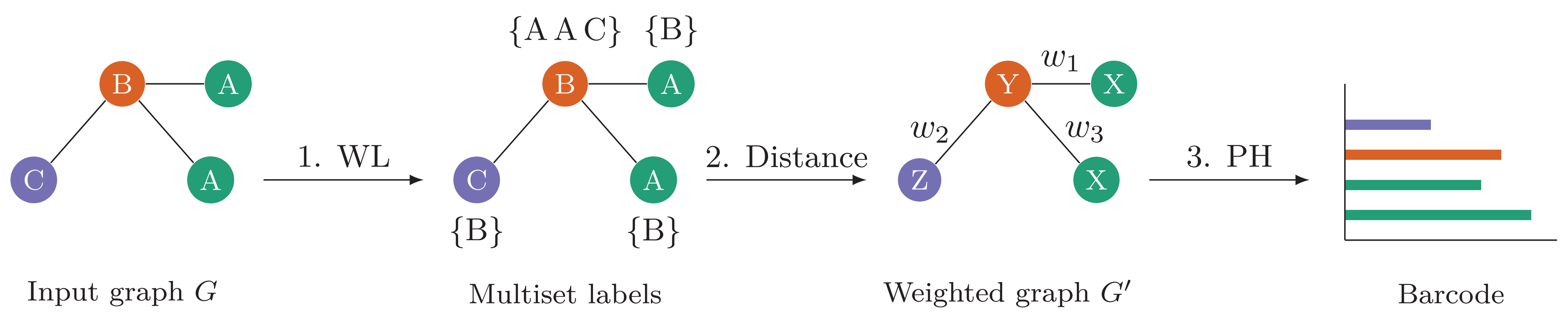
$$\mathrm{pers}(\epsilon_i, \epsilon_j) := |\epsilon_j - \epsilon_i|$$

A visual example of such a "growth process":



This calculation is highly efficient (for connected components and cycles) and known to be very stable.

Method Overview



Distances

1) **Multiset Distance**: Given two multisets A and B, we consider the vectors **x** and **y** containing the counts of the labels in the multiset. We now define:

$$\mathrm{d_M}(\mathbf{x}, \mathbf{y}) := \left(\sum_i |a_i - b_i|^p\right)^{\frac{1}{p}}$$

2) **Label Distance**: Given two adjacent vertices u and v in a graph with their multisets in iteration h as $l_u^{(h)}$ and $l_v^{(h)}$, we define the metric:

$$\mathrm{d_L}(u, v) := \left[\mathrm{l}_u^{(h-1)} \neq \mathrm{l}_v^{(h-1)}\right] + \mathrm{d_M}\left(\mathrm{l}_u^{(h)}, \mathrm{l}_v^{(h)}\right) + \tau \qquad (1)$$

Feature Representation

Persistent Weisfeiler-Lehman (P-WL)

$$\phi^{(h)}_{\text{P-WL}} := \left[\mathfrak{p}^{(h)}(l_0), \mathfrak{p}^{(h)}(l_1), \ldots\right], \text{ with } \mathfrak{p}^{(h)}(l_i) := \sum_{\mathrm{l}(v) = l_i} \mathrm{pers}(v)^p \qquad (2)$$

P-WL with Cycle Information (P-WL-C)

$$\phi^{(h)}_{\text{P-WL-C}} := \left[\mathfrak{z}^{(h)}(l_0), \mathfrak{z}^{(h)}(l_1), \ldots\right], \text{ with } \mathfrak{z}^{(h)}(l_i) := \sum_{l_i \in \mathrm{l}(u,v)} \mathrm{pers}(u, v)^p$$

P-WL-C with uniform edge weights (P-WL-UC)

The original WL can be seen as a special case of P-WL with *uniform* edge weights.
Thus, Equation (2) degenerates to a count function.
P-WL-UC is therefore the original WL augmented with cycle information.

Algorithm

Input: Graph $G = (V, E)$, number of iterations H

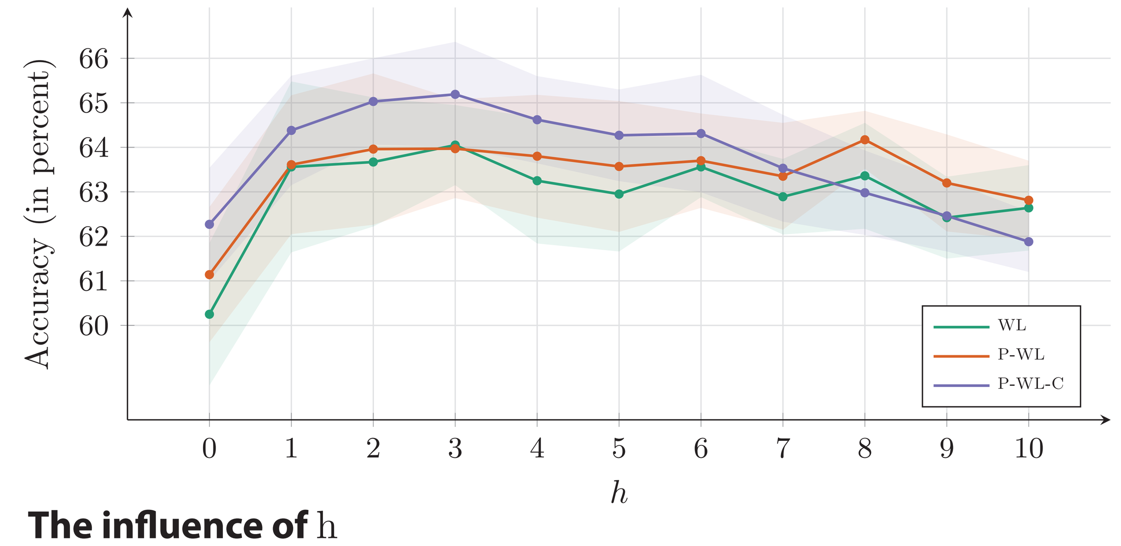
for $h \in \{0, \ldots, H-1\}$ **do**
  $\phi^{(h)}_{\text{P-WL}} := \emptyset$, $\phi^{(h)}_{\text{P-WL-C}} := \emptyset$
  // Get WL multiset labels [1]
  $L^{(h)} := \{\mathrm{l}_v^{(h)} \mid v \in V\} \leftarrow \texttt{WL\_LABELS}(G, h)$
  // Use Eq. (1) to assign each edge (u, v) its weight
  for all $(u, v) \in E$ **do**
    $\mathrm{w}(u, v) \leftarrow \mathrm{d_L}(u, v)$
  end for
  // Compress and re-assign vertex labels
  for all $v \in V$ **do**
    $\mathrm{l}_v^{(h)} \leftarrow \texttt{COMPRESS}\left(\mathrm{l}_v^{(h-1)}, \mathrm{l}_v^{(h)}\right)$
  end for
  // Calculate persistent subtree features
  for all $v \in V$ **do**
    $\phi^{(h)}_{\text{P-WL}}\left[\mathrm{l}_v^{(h)}\right] \leftarrow \phi^{(h)}_{\text{P-WL}}\left[\mathrm{l}_v^{(h)}\right] + \mathrm{pers}(v)^p$
  end for
  // Calculate persistent cycle features
  for all $(u, v) \in E$ **do**
    $\phi^{(h)}_{\text{P-WL-C}}\left[\mathrm{l}_v^{(h)}\right] \leftarrow \phi^{(h)}_{\text{P-WL-C}}\left[\mathrm{l}_v^{(h)}\right] + \mathrm{pers}(u, v)^p$
  end for
end for
return $\left[\phi^{(0)}_{\text{P-WL}}, \phi^{(0)}_{\text{P-WL-C}}, \ldots \phi^{(H-1)}_{\text{P-WL}}, \phi^{(H-1)}_{\text{P-WL-C}}\right]$

Results 1

Mean classification accuracies and standard deviations for node-labelled graphs. We used the highest available accuracy values from the respective publications for all cited algorithms (marked with *). **Bold** indicates the best mean accuracy.

	D & D	MUTAG	NCI1	NCI109	PROTEINS	PTC-MR	PTC-FR	PTC-MM	PTC-FM
V-Hist	78.32 ± 0.35	85.96 ± 0.27	64.40 ± 0.07	63.25 ± 0.12	72.33 ± 0.32	58.31 ± 0.27	$\mathbf{68.13 \pm 0.23}$	66.96 ± 0.51	57.91 ± 0.83
E-Hist	72.90 ± 0.48	85.69 ± 0.46	63.66 ± 0.11	63.27 ± 0.07	72.14 ± 0.39	55.82 ± 0.00	65.53 ± 0.00	61.61 ± 0.00	59.03 ± 0.00
RetGK*	$\mathbf{81.60 \pm 0.30}$	90.30 ± 1.10	84.50 ± 0.20		75.80 ± 0.60	62.15 ± 1.60	67.80 ± 1.10	67.90 ± 1.40	63.90 ± 1.30
WL	79.45 ± 0.38	87.26 ± 1.42	85.58 ± 0.15	84.85 ± 0.19	$\mathbf{76.11 \pm 0.64}$	63.12 ± 1.44	67.64 ± 0.74	67.28 ± 0.97	64.80 ± 0.85
Deep-WL*		82.94 ± 2.68	80.31 ± 0.46	80.32 ± 0.33	75.68 ± 0.54	60.08 ± 2.55			
P-WL	79.34 ± 0.46	86.10 ± 1.37	85.34 ± 0.14	84.78 ± 0.15	75.31 ± 0.73	63.07 ± 1.68	67.30 ± 1.50	$\mathbf{68.40 \pm 1.17}$	64.47 ± 1.84
P-WL-C	78.66 ± 0.32	$\mathbf{90.51 \pm 1.34}$	85.46 ± 0.16	$\mathbf{84.96 \pm 0.34}$	75.27 ± 0.38	$\mathbf{64.02 \pm 0.82}$	67.15 ± 1.09	$\mathbf{68.57 \pm 1.76}$	$\mathbf{65.78 \pm 1.22}$
P-WL-UC	78.50 ± 0.41	85.17 ± 0.29	$\mathbf{85.62 \pm 0.27}$	$\mathbf{85.11 \pm 0.30}$	75.86 ± 0.78	$\mathbf{63.46 \pm 1.58}$	67.02 ± 1.29	$\mathbf{68.01 \pm 1.04}$	$\mathbf{65.44 \pm 1.18}$

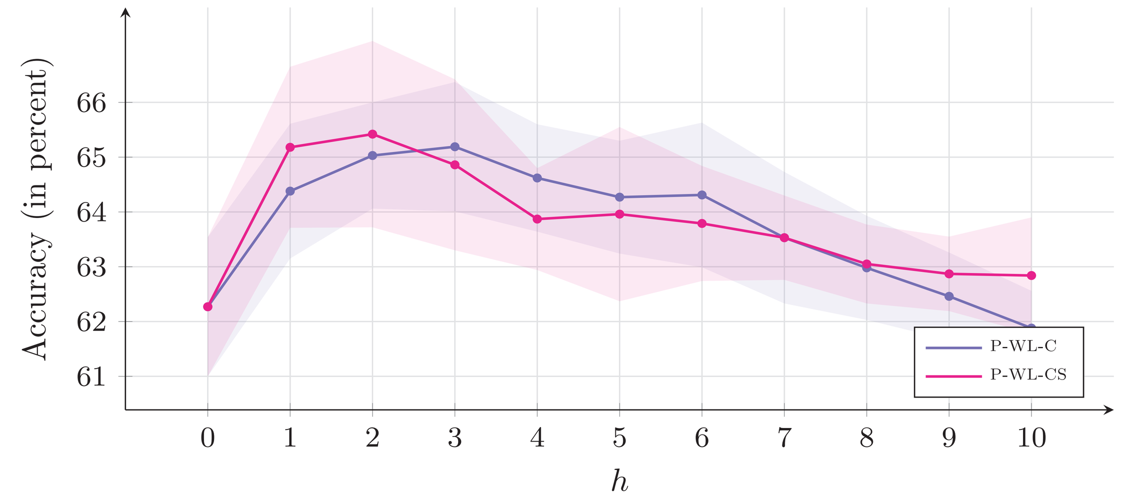
Results 2



The influence of h

We observe that persistent features (orange, violet) tend to reach higher mean accuracies at smaller values of h. In particular PWL-C, with its usage of cycle features, reaches its maximum accuracy for smaller values of h than the Weisfeiler-Lehman graph kernel.

Results 3



The effect of dissimilarity measures

Using a dissimilarity measure, e.g. the Kullback-Leibler divergence, instead of our metric, we observe larger standard deviations and initially higher maximum accuracy values that vanish in practice.

Code & Data