Graph Filtration Learning – Supplementary Material

This supplementary material contains the full proof of Lemma 1 omitted in the main work and additional information to the used datasets. It further contains details to the implementation of the models used in the experiments. For readability, all necessary definitions and results are restated and the numbering matches the original numbering.

1. Dataset details

The following table contains a selection of statistics relevant to the datasets used in our experiments.

Datasets	REDDIT-BINARY	REDDIT-MULTI-5K	IMDB-BINARY	IMDB-MULTI	PROTEINS	NCI1
# graphs	2000	4999	1000	1500	1113	4110
# classes	2	5	2	3	2	2
∅ nodes	429.6	508.5	19.8	13.0	39.1	29.9
∅ edges	497.8	594.9	96.5	65.9	72.8	32.3
# labels	n/a	n/a	n/a	n/a	3	37

2. Proof of Lemma 1

Definition 1 (Learnable vertex filter function)**.** Let $\mathbb{V}$ be a vertex domain, $\mathbb{K}$ the set of possible simplicial complexes over $\mathbb{V}$ and let

$$f : \mathbb{R} \times \mathbb{K} \times \mathbb{V} \to \mathbb{R} \qquad (\theta, K, v) \mapsto f(\theta, K, v)$$

be differentiable in θ for $K \in \mathbb{K}, v \in \mathbb{V}$. Then, we call f a *learnable vertex filter function* with parameter θ.

Definition 2 (Barcode coordinate function)**.** Let $s : \mathbb{R}^2 \to \mathbb{R}$ be a differentiable function that vanishes on the diagonal of $\mathbb{R}^2$. Then

$$\mathcal{V} : \mathbb{B} \to \mathbb{R} \qquad \mathcal{B} \mapsto \sum_{(b,d)\in\mathcal{B}} s(b, d)$$

is called *barcode coordinate function*.

Lemma 1. *Let K be a* finite *simplicial complex with vertex set $V = \{v_1, \dots, v_n\}$, $f : \mathbb{R} \times \mathbb{K} \times \mathbb{V} \to \mathbb{R}$ be a learnable vertex filter function as in Definition 1 and $\mathcal{V}$ a barcode coordinate function as in Definition 2. If, for $\theta_0 \in \mathbb{R}$, it holds that the pairwise vertex filter values are distinct, i.e.,*

$$f(\theta_0, K, v_i) \neq f(\theta_0, K, v_j) \quad \textit{for} \quad 1 \leq i < j \leq n$$

then the mapping

$$\theta \mapsto \mathcal{V}\big(\mathsf{ph}_k^f(\theta, K)\big) \tag{1}$$

is differentiable *at* θ_0.

Proof. For notational convenience, let $y_i = f(\theta_0, K, v_i) = f(\theta_0, v_i)$. Also, let π the sorting permutation of $\big(y_i(\theta_0)\big)_{i=1}^n$, i.e., $y_{\pi(1)}(\theta_0) < y_{\pi(2)}(\theta_0) < \cdots < y_{\pi(n)}(\theta_0)$. By assumption the pairwise filter values are distinct, thus there is a neighborhood around θ_0 such that the ordering of the filtration values is not modified by changes of θ within this neighborhood, i.e.,

$$\exists \varepsilon > 0 \forall h \in \mathbb{R} : |h| < \varepsilon \Rightarrow y_i(\theta_0 + h) \neq y_j(\theta_0 + h) \tag{2}$$

and

$$y_{\pi(1)}(\theta_0 + h) < y_{\pi(2)}(\theta_0 + h) < \cdots < y_{\pi(n)}(\theta_0 + h) \ . \tag{3}$$

This implies that a sufficiently small change, h, of θ_0 does not change the induced filtrations. Formally,

$$\big(K_i^f(\theta_0)\big)_{i=0}^n = \big(K_i^f(\theta_0+h)\big)_{i=0}^n \quad \text{with} \quad 0 \leq i \leq n \ . \tag{4}$$

Importantly, this means that

$$\mu_k^{i,j}(\theta_0) = \mu_k^{i,j}(\theta_0+h) \quad \text{for} \quad 1 \leq i < j \leq n \ . \tag{5}$$

We next show that the derivative of Eq. (1) with respect to h exists. By assumption, s is differentiable and thus $s\big(f(\cdot, v_i), s(f(\cdot, v_j)\big)$ is differentiable. Now consider

$$\begin{aligned}
&\lim_{|h|\to 0} \frac{\mathcal{V}(\mathsf{ph}_k^f(K,\theta_0)) - \mathcal{V}(\mathsf{ph}_k^f(K,\theta_0+h))}{h} \\
&= \lim_{|h|\to 0} \frac{\sum\limits_{i<j} \mu_k^{i,j}(\theta_0)\cdot s\big(y_{\pi(i)}(\theta_0), y_{\pi(j)}(\theta_0)\big) - \sum\limits_{i<j} \mu_k^{i,j}(\theta_0+h)\cdot s\big(y_{\pi(i)}(\theta_0+h), y_{\pi(j)}(\theta_0+h)\big)}{h} \\
&= \lim_{|h|\to 0} \frac{\sum\limits_{i<j} \mu_k^{i,j}(\theta_0)\cdot \big[s\big(y_{\pi(i)}(\theta_0), y_{\pi(j)}(\theta_0)\big) - s\big(y_{\pi(i)}(\theta_0+h), y_{\pi(j)}(\theta_0+h)\big)\big]}{h} = && \text{(by Eq. (5))} \\
&= \sum_{i<j} \mu_k^{i,j}(\theta_0)\cdot \lim_{|h|\to 0} \frac{s\big(y_{\pi(i)}(\theta_0), y_{\pi(j)}(\theta_0)\big) - s\big(y_{\pi(i)}(\theta_0+h), y_{\pi(j)}(\theta_0+h)\big)}{h} \\
&= \sum_{i<j} \mu_k^{i,j}(\theta_0)\cdot \lim_{|h|\to 0} \frac{s\big(f(\theta_0, v_{\pi(i)}), f(\theta_0, v_{\pi(j)})\big) - s\big(f(\theta_0+h, v_{\pi(i)}), f(\theta_0+h, v_{\pi(j)})\big)}{h} \\
&= \sum_{i<j} \mu_k^{i,j}(\theta_0)\cdot \frac{\partial s\big(f(\theta, v_{\pi(i)}), f\big(\theta, v_{\pi(j)}\big)\big)}{\partial \theta}(\theta_0) \ .
\end{aligned}$$

This concludes the proof, since the derivative within the summation exists by assumption. □

3. Architectural details

As mentioned in Section 5 (Experiments), we implement the learnable vertex filter function $f(\theta, K, v_i)$ using a single `GIN-`ε layer from (Xu et al., 2019) (with ε set as a learnable parameter). The internal architecture is as follows:

```
Embedding[n,64]-FC[64,64]-BatchNorm-LeakyReLU-FC(64,64).
```

Here, `n` denotes the dimension of the node attributes. For example, if initial node features are based on the degree function and the maximum degree over all graphs is 200, then `n=200+1`. In other words, `n` is the *number of embedding vectors* in $\mathbb{R}^{64}$ used to represent node degrees.

The multi-layer perceptron (`MLP`) mapping the output of the `GIN` layer to a real-valued node filtration value is parameterized as:

```
Embedding[64,64]-FC[64,64]-BatchNorm-LeakyReLU-FC(64,1)-Sigmoid.
```

As classifier, we use a simple `MLP` of the form

```
FC[300,64]-ReLU-FC[64,#classes].
```

Here, the input dimensionality is 300, as each barcode is represented by a 100-dimensional vector.

References

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *ICLR*, 2019.