Graph Learning: Generalisation and Expressiveness

Floris Geerts (University of Antwerp, Belgium)

The plan is to

- introduce graph learning and generalisation;
- recall expressiveness of graph learning methods; and
- combine the two.

Main source: WL meet VC by Christopher Morris, Floris Geerts, Jan Tönshoff & Martin Grohe. In Proc. of ICML 2023, pp 25275-25302.
https://proceedings.mlr.press/v202/morris23a/morris23a.pdf

A bit of graph learning theory

- Let $\mathcal{G}$ be the set of all *graphs* and let $\mathbb{Y}$ be a set of *labels*.
- We are given some *training data* $\mathcal{T}$, i.e., elements $(G_1, y_1), \ldots, (G_m, y_m)$ in $\mathcal{G} \times \mathbb{Y}$.
- We are given some class $\mathcal{H}$ of *graph classifiers* $h : \mathcal{G} \to \mathbb{Y}$, or more generally, *graph embedding methods*.

Learning=Empirical Risk Minimisation (ERM)

Find the **best graph classifier** from $\mathcal{H}$ for the training data $\mathcal{T}$, that is, return

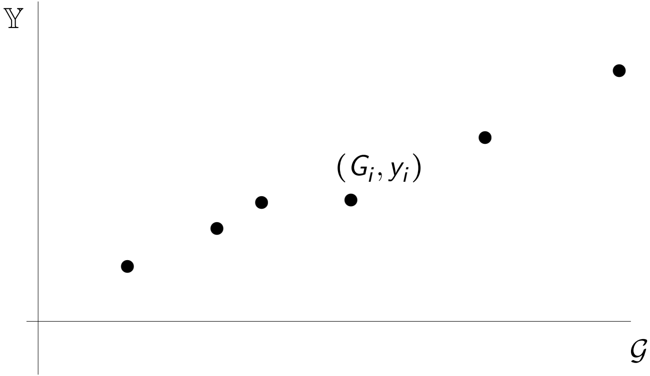
$$h^\star_{\mathcal{T}} := \arg\min_{h \in \mathcal{H}} L_{\mathcal{T}}(h)$$

with $L_{\mathcal{T}}(\cdot)$ the *empirical loss function* given by

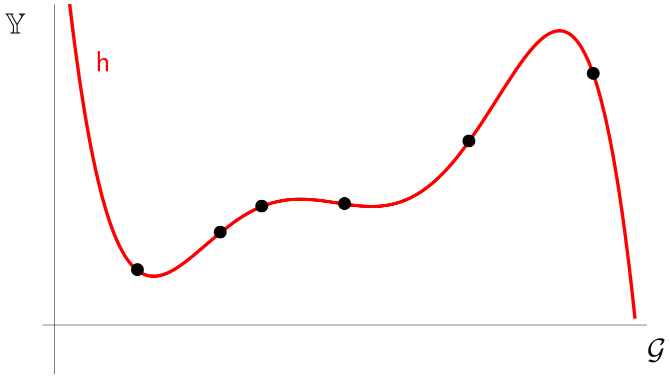
$$L_{\mathcal{T}}(h) := \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}[h(G_i) \neq y_i].$$

See also: Theory of graph neural networks: Representation and learning by Stefanie Jegelka. In Int. Cong. Math. 2022, Vol. 7, pp. 5450–5476. `https://doi.org/10.4171/icm2022/162`

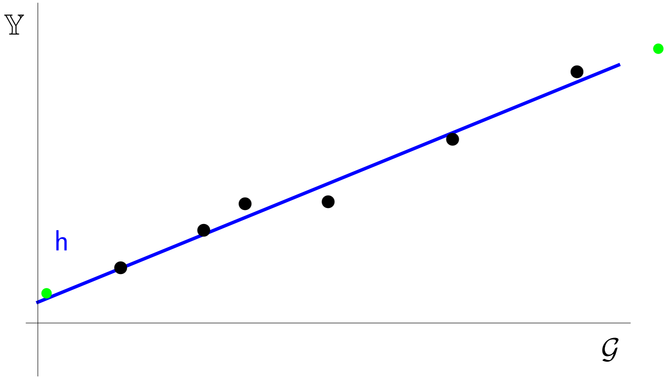
- Let assume we have some training data $\mathcal{T}$:



- The red hypothesis h makes us very happy: $L_{\mathcal{T}}(h) = 0$!



- But perhaps a bit of error is fine...



- Just making predictions for elements in the training data is not so interesting.
- One is more interested in predicting the labels of graphs that are *not part* of the training data.
- Let us assume a *distribution* $\mathcal{D}$ over the product space $\mathcal{G} \times \mathbb{Y}$.

Risk Minimisation

Find the **best graph classifier** from $\mathcal{H}$ over *all* input elements, that is

$$\hat{h}_{\mathcal{D}} := \arg\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$$

with $L_{\mathcal{D}}(\cdot)$ the *population risk* (expected loss) given by

$$L_{\mathcal{D}}(h) := \mathrm{Prob}_{(G,y)\sim\mathcal{D}}[h(G) \neq y].$$

Caveat: We don't know $\mathcal{D}$.

What is generalisation?

- A class $\mathcal{H}$ has *good generalisation* if the empirical risk classifier $h^\star_{\mathcal{T}}$ approximates the expected risk classifier $\hat{h}_{\mathcal{D}}$ well, and this with a small number of training data.

We are interested in bounding the **generalisation error**, defined as

$$L_{\mathcal{D}}(h) - L_{\mathcal{T}}(h),$$

for $h \in \mathcal{H}$ in terms of e.g., training size m or some *complexity* measure of the underlying hypothesis class $\mathcal{H}$.

Example complexity measures are the Vapnik-Chervonenkis (VC) dimension, Rademacher complexity, robustness, ..

VC dimension

- For simplicity, assume binary classification from now on, i.e., $\mathbb{Y} = \{0, 1\}$.
- A set $G_1, \ldots, G_d$ is *shattered by* a class $\mathcal{H}$ of graph classifiers, if for any labeling $y_1, \ldots, y_d$, there is a graph classifier $h \in \mathcal{H}$ such that $h(G_1) = y_1, \ldots, h(G_d) = y_d$.
- VC dimension of $\mathcal{H}$ is *maximal* number of graphs that can be shattered.

VC dimension & generalisation error

Theorem (Vapnik&Chervonenkis-1964)

For $\delta > 0$, with probability $1 - \delta$, for all $h \in \mathcal{H}$:

$$L_{\mathcal{D}}(h) - L_{\mathcal{T}}(h) \leq \sqrt{\frac{2d \log \frac{em}{d}}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}},$$

were d is the VC dimension of $\mathcal{H}$.

- Large VC dimension d implies need for *large* training set to reduce overfitting.
- Note $d \leq em$ for this bound to make sense.

See also: The uniform convergence of frequencies of the appearance of events to their probabilities by Vladimir Vapnik and Alexey Cervonenkis. In Dokl. Akad. Nauk SSSR, 181, 4, 1968. English version appeared in Theory of Probability & Its Applications, 16 (2), pp 264–280, 1971. https://doi.org/10.1137/1116025

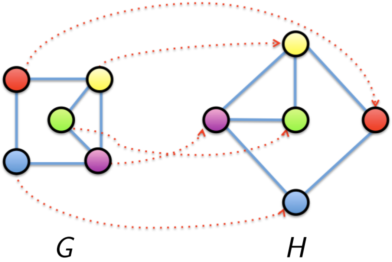
Questions

- Can we say more about VC dimension of graph embedding methods?
- Can we connect this to expressiveness of these methods?

- Let $G = (V(G), E(G))$ and $H = (V(H), E(H))$ be two graphs. An *isomorphism* from G to H is an **edge-preserving vertex bijection**.
- That is, a bijection $f : V(G) \to V(H)$ such that

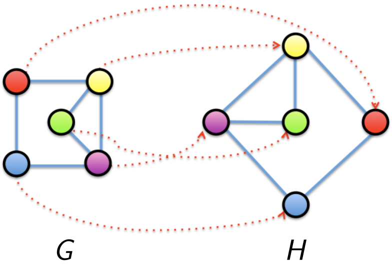
$$(v, w) \in E(G) \Longleftrightarrow (f(v), f(w)) \in E(H)$$

holds. We write $G \cong H$ if such an isomorphism exists, and say that G and H are *isomorphic*.



Decide whether two graphs are isomorphic.

- Complexity: open
- Quasi-polynomial algorithm $n^{poly(\log(n))}$ (Babai 2015/2017)



- In graph learning, as mentioned, the hypothesis class $\mathcal{H}$ consists of *graph classifiers* or, more generally, *embedding methods* $h : \mathcal{G} \to \mathbb{Y}$; Similar notions are in place for vertex and k-tuple of vertex embeddings used in e.g., link prediction.
- A crucial property of classes $\mathcal{H}$ used in graph learning is that whenever two graphs G and H are *isomorphic*, i.e., $G \cong H$, then $h(G) = h(H)$. That is, $\mathcal{H}$ consists of *invariant* embeddings.
- Indeed, one does not want to learn things that depend on the graph representation, e.g., on the order of vertices when building an adjacency matrix of a graph.

Distinguishing power

- Measured in terms of which pairs of inputs (graphs) can be distinguished/separated by elements in $\mathcal{H}$.

- We define $\rho(\mathcal{H}) := \big\{(G, H) \in \mathcal{G} \times \mathcal{G} \mid \exists h \in \mathcal{H} \text{ s.t. } h(G) \neq h(H)\big\}$.

- Hypothesis class $\mathcal{H}$ is *more expressive* than $\mathcal{H}'$ if $\rho(\mathcal{H}') \subseteq \rho(\mathcal{H})$.
- For any invariant $\mathcal{H}$, $\rho(\mathcal{H}) \subseteq \rho(\text{ISO})$ where ISO refers to graph isomorphism test, e.g., assigning the the isomorphism type to each graph. In this case, $\rho(\mathcal{H})$ consists of all pairs of non-isomorphic graphs.
- In the machine learning literature, $\rho(\mathcal{H})$ is well-understood for various classes of graph learning methods.

A first observation

- The VC dimension of a class $\mathcal{H}$ is bounded by the distinguishing power of the class.

- Indeed, let B the maximal degree of $\rho(\mathcal{H})$ (viewed as a graph); This implies that that there is no graph G that can be distinguished from more than B graphs $G_1, \ldots, G_B$ by elements in $\mathcal{H}$.

Proposition

We have $\mathrm{VCD}(\mathcal{H}) \leq B + 1$

- Indeed, it is impossible shatter more that $B + 1$ graphs using elements from $\mathcal{H}$.
- We will use the notation $\mathrm{VCD}_{\mathcal{X}}(\mathcal{H})$ for the VC dimension of $\mathcal{H}$ *restricted* to inputs in $\mathcal{X} \subseteq \mathcal{G}$.

Let us zoom in into some specific class $\mathcal{H}$: Message-Passing Neural Networks (MPNNs)

GNN(d, L): L-layered Graph Neural Networks of width d

- Vertex level:

$$\underbrace{F^{(0)}(G, v)}_{\in \mathbb{R}^d} \quad F^{(t)}(G, v) = \sigma\left(\underbrace{W_1^{(t)}}_{d\times d} F^{(t-1)}(G, v) + \underbrace{W_2^{(t)}}_{d\times d} \sum_{w \in N(G,v)} F^{(t-1)}(G, w)\right) \in \mathbb{R}^d$$

- Graph level:

$$F(G) = \sigma\Big(W \sum_v F^{(L)}(G, v) + b\Big)$$

MPNN(d, L): L-layered Message-Passing Neural Network of width d.

- Vertex level:

$$F^{(t)}(G,v) = \mathrm{UPD}^{(t)}\Big(F^{(t-1)}(G,v), \mathrm{AGG}^{(t)}\big(\{\!\{F^{(t-1)}(G,w) \mid w \in N(G,v)\}\!\}\big)\Big)$$

- Graph level:

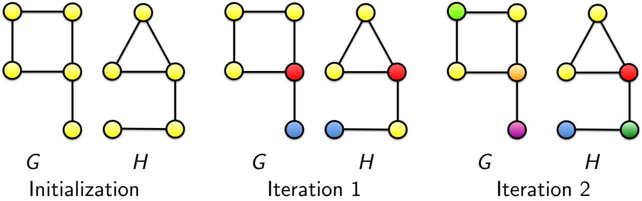
$$F(G) = \mathrm{READOUT}\big(\{\!\{F^{(L)}(G,v) \mid v \in V\}\!\}\big)$$

Heuristic for *graph isomorphism testing*

1-dim. Weisfeiler-Leman algorithm

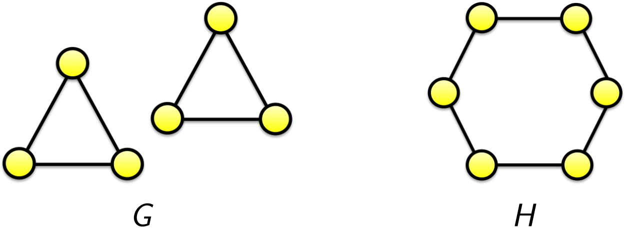
- Iteration: Two vertices get *identical colours* iff their *coloured neighbourhoods are identical*

Two graphs are distinguished by 1-WL if they have different *colour histograms*.



- One side graph isomorphism test. If histograms differ, then non-isomorphic.

1-dim. Weisfeiler-Leman algorithm



Relationship between 1-WL and GNNs

Theorem

MPNNs are bounded in expressive power by 1-WL, that is, $\rho(\mathit{MPNNs}) \subseteq \rho(\mathit{1\text{-}WL})$.

Since GNNs $\subseteq$ MPNNs, also GNNs are bounded by 1-WL.

Theorem

There exists a GNN architecture *and corresponding* weights *such that it has the* same power *as the* 1-WL. *Hence,* $\rho(\mathit{1\text{-}WL}) \subseteq \rho(\mathit{GNNs})$.

As a consequence, $\rho(\text{1-WL}) = \rho(\mathit{GNNs}) = \rho(\mathit{MPNNs})$.
Moreover, L iterations of 1-WL corresponds to L layers in the GNNs.

VC Dimension of GNNs when fixing graph size

- Let $\mathcal{G}_{n,d}$ be a set of graphs order n with d-dimensional boolean vertex features.
- Let $m_{n,d,L}$ be the number 1-WL-distinguishable graphs in $\mathcal{G}_{n,d}$ after L iterations of 1-WL.

Theorem

For all n, d, and $L > 0$, all $m_{n,d,L}$ 1-WL-distinguishable graphs of order n with d-dimensional boolean features can be shattered by sufficiently wide L-layer GNNs using piecewise linear activation functions. Hence,

$$VCD_{\mathcal{G}_{d,n}}\big(GNN(L)\big) = m_{n,d,L}.$$

- Sufficiently wide means $d \in \mathcal{O}(nm_{n,d,L})$.[1]
- Without restrictions on width and size on graphs, VC dimension of GNNs and MPNNs is ∞

[1] Can be improved by recent result. On dimensionality of feature vectors in MPNNs by César Bravo, Alexander Kozachinskiy & Cristobal Rojas. In Proc. ICML 2024. `https://openreview.net/forum?id=UjDp4Wkq2V`

VC Dimension of GNNs: Uniform case – Bounded bitlength

What if we *restrict width*, but arbitrary size graphs? Also here, VC dimension is ∞. Based on result for GNNs whose weights have *fixed bitlength b*.

Theorem

There exists a family $\mathcal{F}_b$ of simple 1-layer GNNs of width one and bitlength $\mathcal{O}(b)$ using piece-wise linear activation functions such that its VC dimension is exactly b.

Letting $b \to \infty$ results in infinite VC dimension, even for width one GNNs but unbounded bitlength.

VC Dimension of GNNs: Color complexity

- We now consider the class $\mathcal{G}_{d,\leq u}$ consisting of graphs having d-dimensional features and color complexity at most u
- Color complexity = number of colors used by 1-WL.

Theorem

Assume d and L in $\mathbb{N}$, and GNNs in $\mathsf{GNN}_{\mathsf{slp}}(d, L)$ using piece-wise polynomial activation functions with $p > 0$ pieces and degree $\delta \geq 0$. Let $P = d(2dL + L + 1) + 1$ be the number of parameters in the GNNs. For all u in $\mathbb{N}$,

$$VCD_{\mathcal{G}_{d,\leq u}}(GNN_{slp}(d, L)) \leq \begin{cases} \mathcal{O}(LP\log(puP)) & \text{if } \delta = 1, \\ \mathcal{O}(LP\log(puP) + L^2 P\log(\delta)) & \text{if } \delta > 1. \end{cases}$$

- Dependency on u cannot be improved: tight w.r.t. color complexity.

Commentary

- Most of the results extend easily to higher-order MPNNs.
- According to theory, VC dimension increases with expressive power of $\mathcal{H}$.
- If our domain of graphs has low 1-WL complexity, less training data is needed to get good generalisation. (Think of regular graphs!)
- This may need more investigation.

End of story?

Margin-based bounds (very rough slide)

- It was experimentally shown that *adding power not always results in worse generalisation*.
- Generalisation error bounds exist in terms of VC dimension and *margin*, the latter being the minimal distance to decision boundaries.
- The larger the margin, the lower generalisation error.
- So, two classes with same VC dimension may behave differently depending on margins obtained.

We can thus obtain a more fine-grained view of VC dimension in the graph setting.

Weisfeiler-Leman at the margin: When more expressivity matters by Billy Joe Franks, Christopher Morris, Ameya Velingker & Floris Geerts. In Proc. of ICML, 2024. `https://openreview.net/forum?id=HTNgNt8CTJ`

Towards Bridging Generalization and Expressivity of Graph Neural Networks. Shouheng Li, Dongwoo Kim, Qing Wang, Floris Geerts. Proceedings of 13th International Conference on Learning Representations (ICLR), 2025

Other techniques/questions

- Robustness framework of Xu & Manor (2010). Ongoing work to bound covering numbers of graph spaces relative to graph learning method and metric.
- Analysing the Graph Neural Tangent Kernel in order to obtain *conditional* expressiveness results.
- When GNNs are combined, stacked, etc, i.e., when we have an *algebra of GNNs*, how does VC dimension change under such operations?

For robustness: Robustness and generalization by Huan Xu & Shie Mannor. In Mach. Learn, 86, pp. 391â423 (2012). `https://doi.org/10.1007/s10994-011-5268-1`

Covered Forest: Fine-grained generalization analysis of graph neural networks. Antonis Vasileiou, Ben Finkelshtein, Floris Geerts, Ron Levie, Christopher Morris, under review, 2025.