Weisfeiler and Leman Consider Sampling

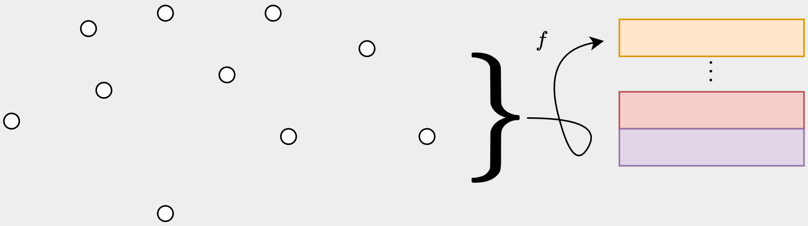
Blessings of Dimensionality

Abdullah Naeem Malik

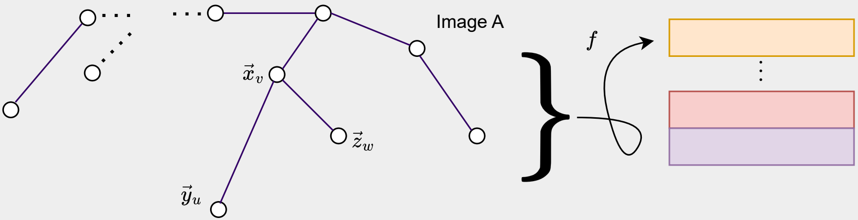
Department of Science and Math
Tallahassee State College

October 25, 2024

Neural Networks

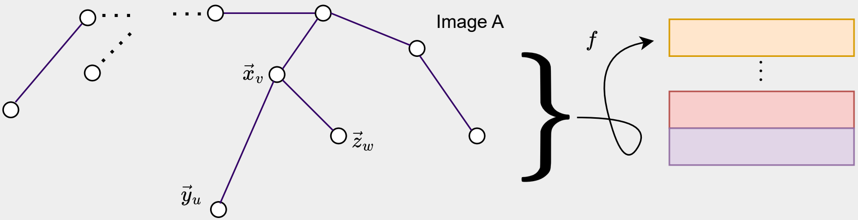


Message passing in Graph Neural Networks



$$x_v^{(k+1)} = \text{COMBINE}\left(x_v^{(k)}, \text{AGGREGATE}^{(k+1)}\left(\left\{x_u^{(k)} : u \in N(v)\right\}\right)\right)$$

Message passing in Graph Neural Networks



$$x_v^{(k+1)} = \text{COMBINE}\left(x_v^{(k)}, \text{AGGREGATE}^{(k+1)}\left(\left\{x_u^{(k)} : u \in N(v)\right\}\right)\right)$$

$F^{out} = \psi \circ ((P(A))\,F)$. Here, $P(A) = AW + I$ and $\psi : \mathbb{R} \longrightarrow \mathbb{R}$

. In general, $P(A) = I + AW_1 + A^2W_2 + ... + A^nW_n$

Message passing in Graph Neural Networks

How powerful are graph neural networks?[MBHSL19, XHLJ18]

Theorem

Let G_1 and G_2 be any two non-isomorphic graphs. If a graph neural network $f : \mathcal{G} \longrightarrow \mathbb{R}^d$ maps G_1 and G_2 to different embeddings, the Weisfeiler-Leman graph isomorphism test also decides G_1 and G_2 are not isomorphic. Converse holds if COMBINE and AGGREGATE are injective[XHLJ18].

Weisfeiler-Leman Algorithm[WL68]

Definition

A **coloring** of a graph $G = (V, E)$ is a function $c : V \longrightarrow \mathbb{N}$. A (perfect) **hashing** is any injective function.

Weisfeiler-Leman Algorithm[WL68]

Definition

A **coloring** of a graph $G = (V, E)$ is a function $c : V \longrightarrow \mathbb{N}$. A (perfect) **hashing** is any injective function.

Algorithm Weisfeiler-Leman (WL) or Naive vertex refinement[WL68]

1: **Input**: (V, E, X_V) Here, $x_v \in \mathbb{Z}_2^d$
2: $c(v) = c^{(0)}(v) \longleftarrow hash(x_v)$
3: **while** $c^{(t)}(v) = c^{(t+1)}(v)\ \forall v \in V$ **do**
4: $\quad c^{(t+1)}(v) \longleftarrow \text{hash}\left(c^{(t)}(v), \{\{c^{(t)}(w) : w \in N(v)\}\}\right)$
5: **Output:** $c^{(T)}(v)\ \forall v \in V$

Weisfeiler-Leman Algorithm[WL68]

Definition

A **coloring** of a graph $G = (V, E)$ is a function $c : V \longrightarrow \mathbb{N}$. A (perfect) **hashing** is any injective function.

Algorithm Weisfeiler-Leman (WL) or Naive vertex refinement[WL68]

Input: (V, E, X_V) Here, $x_v \in \mathbb{Z}_2^d$

$c(v) = c^{(0)}(v) \longleftarrow hash(x_v)$

while $c^{(t)}(v) = c^{(t+1)}(v)\ \forall v \in V$ **do**

  $c^{(t+1)}(v) \longleftarrow \text{hash}\left(c^{(t)}(v), \{\{c^{(t)}(w) : w \in N(v)\}\}\right)$

Output: $c^{(T)}(v)\ \forall v \in V$

Secretly message passing!

Weisfeiler-Leman Algorithm[WL68]

Definition

A **coloring** of a graph $G = (V, E)$ is a function $c : V \longrightarrow \mathbb{N}$. A (perfect) **hashing** is any injective function.

Algorithm Weisfeiler-Leman (WL) or Naive vertex refinement[WL68]

Input: (V, E, X_V) — Here, $x_v \in \mathbb{Z}_2^d$
$c(v) = c^{(0)}(v) \longleftarrow hash(x_v)$
while $c^{(t)}(v) = c^{(t+1)}(v)\ \forall v \in V$ **do**
  $c^{(t+1)}(v) \longleftarrow \text{hash}\left(c^{(t)}(v), \{\{c^{(t)}(w) : w \in N(v)\}\}\right)$
Output: $c^{(T)}(v)\ \forall v \in V$

Secretly message passing!
But not belief propagation

Message Passing on Simplicial Complexes

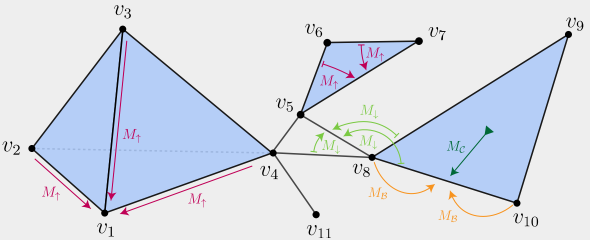


Image from [BFW+21]

Simplicial WL [BFW+21]: $c^{t+1}(\sigma)$ =hash of coloring at t of σ and..

Message Passing on Simplicial Complexes

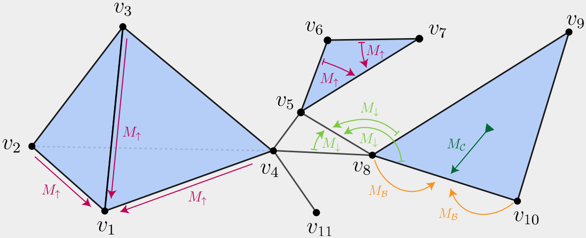


Image from [BFW$^+$21]

Simplicial WL [BFW$^+$21]: $c^{t+1}(\sigma) =$hash of coloring at t of σ and.. coloring of its upper-neighbors, lower-neighbors, boundaries and coboundaries.

Message Passing on Simplicial Complexes

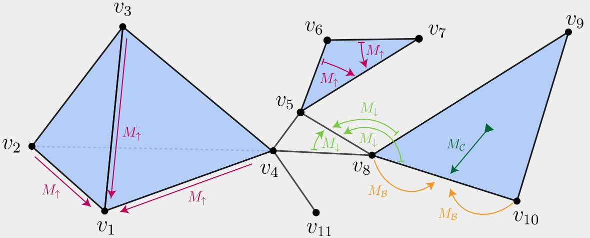


Image from [BFW+21]

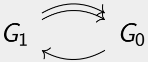
Simplicial WL [BFW+21]: $c^{t+1}(\sigma)$ =hash of coloring at t of σ and.. coloring of its upper-neighbors, lower-neighbors, boundaries and coboundaries.

SWL is strictly stronger than 3-WL[BFW+21]

Directed Cliques

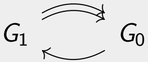
$G_1 \rightrightarrows G_0$ **Directed Graph**

Directed Cliques

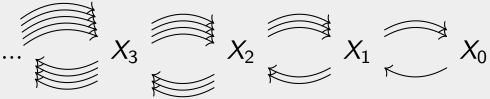


Directed Graph

Directed Cliques

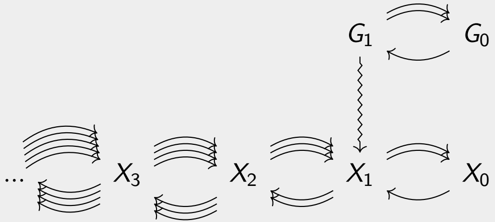


Directed Graph



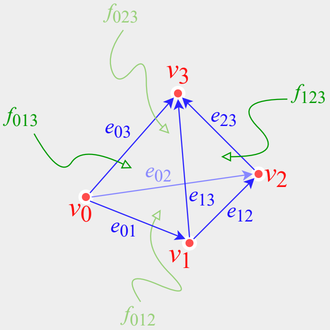
Simplicial Set

Directed Cliques



Directed Graph

Simplicial Set



$\nu_0, \nu_1, \nu_2, \nu_3 \in X_0$

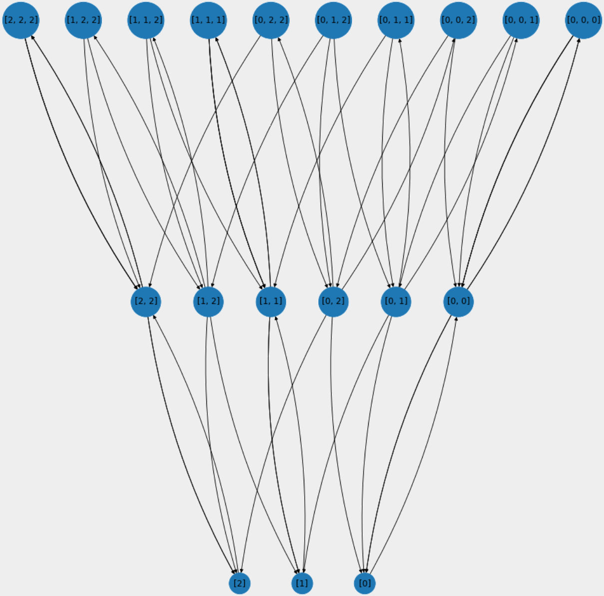
$e_{01}, e_{02}, e_{03}, e_{12}, e_{13}, e_{23} \in X_1$

$f_{012}, f_{013}, f_{023}, f_{123} \in X_2$

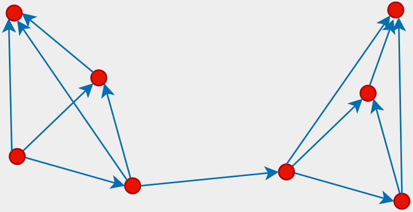
$g_{0123} \in X_3$

Implementation

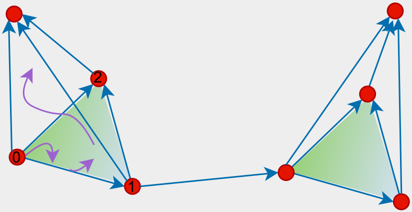
Graph $\rightsquigarrow$ $\mathcal{S}$Set $\rightsquigarrow$ Adj Graph

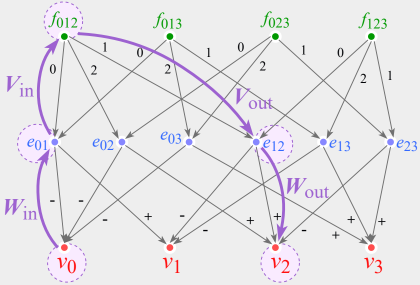


CinchNET



CinchNET





$$\sum_{j=1}^{k} \left(D_d^{-1/2} A_d D_d^{-1/2}\right)^j W_j \left(\sum_{i=1}^{k} \left(D_s^{-1/2} A_s D_s^{-1/2}\right)^i W_i F\right)$$

To pursue

- Can we rigorously prove that message passing polynomial $P(A) \simeq$ higher order correlations?

To pursue

- Can we rigorously prove that message passing polynomial $P(A) \simeq$ higher order correlations?
  - $P(A) = AW$ might help approximate covariance matrix

To pursue

- Can we rigorously prove that message passing polynomial $P(A) \simeq$ higher order correlations?
  - $P(A) = AW$ might help approximate covariance matrix
- Do higher order WL tests give us a variant of junction tree algorithm that does not eliminate cycles?

To pursue

- Can we rigorously prove that message passing polynomial $P(A) \simeq$ higher order correlations?
  - $P(A) = AW$ might help approximate covariance matrix
- Do higher order WL tests give us a variant of junction tree algorithm that does not eliminate cycles?
  - Edges, seen as boundaries of 2-simplices, can have their marginal probabilities determined.

References I

Cristian Bodnar, Fabrizio Frasca, Yuguang Wang, Nina Otter, Guido F Montufar, Pietro Lio, and Michael Bronstein, *Weisfeiler and lehman go topological: Message passing simplicial networks*, International Conference on Machine Learning, PMLR, 2021, pp. 1026–1037.

Jin-Yi Cai, Martin Fürer, and Neil Immerman, *An optimal lower bound on the number of variables for graph identification*, Combinatorica **12** (1992), no. 4, 389–410.

Ningyuan Teresa Huang and Soledad Villar, *A short tutorial on the weisfeiler-lehman test and its variants*, ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2021, pp. 8533–8537.

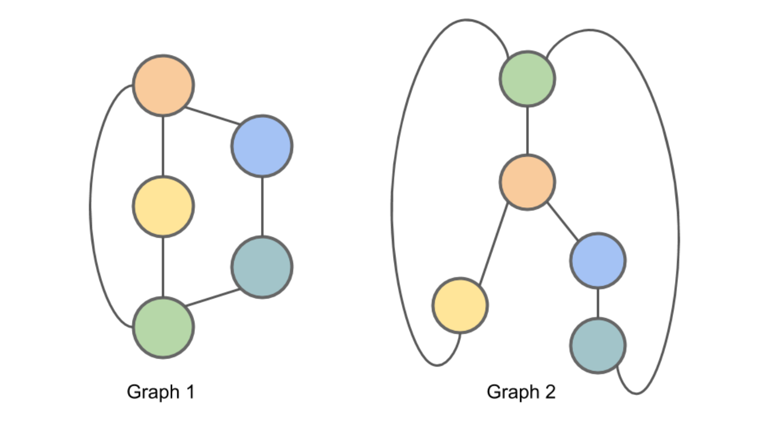
Will Hamilton, Zhitao Ying, and Jure Leskovec, *Inductive representation learning on large graphs*, Advances in neural information processing systems **30** (2017).

References II

- Thomas N Kipf and Max Welling, *Semi-supervised classification with graph convolutional networks*, IEEE Transactions on Neural Networks **20** (2008), no. 1, 61–80.
- Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman, *Provably powerful graph networks*, Advances in neural information processing systems **32** (2019).
- Martin Mladenov Pascal Schweitzer Martin Grohe, Kristian Kersting, *Color refinement and its applications*, An Introduction to Lifted Probabilistic Inference (Sriraam Natarajan Davide Poole Guy Van den Broeck, Kristian Kersting, ed.), MIT Press, 2021.
- Boris Weisfeiler and Andrei Leman, *The reduction of a graph to canonical form and the algebra which appears therein*, nti, Series **2** (1968), no. 9, 12–16.
- Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka, *How powerful are graph neural networks?*, arXiv preprint arXiv:1810.00826 (2018).

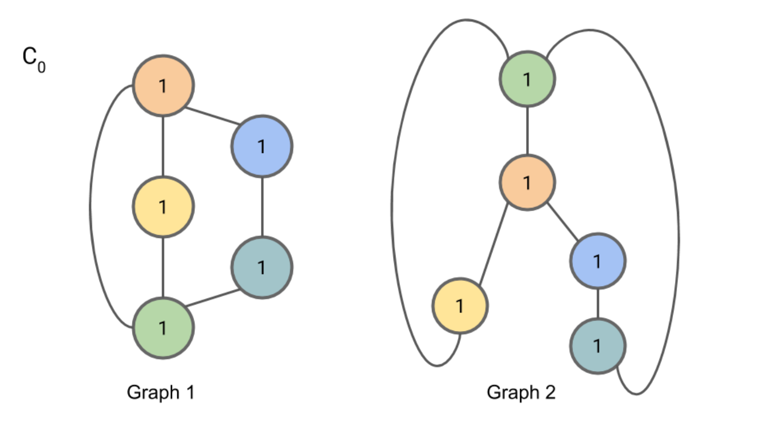
Thank you!

Weisfeiler-Leman Algorithm Example



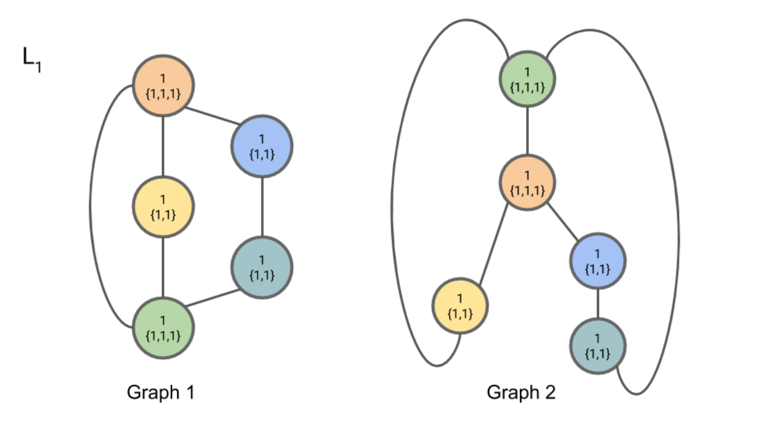
Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Leman Algorithm Example



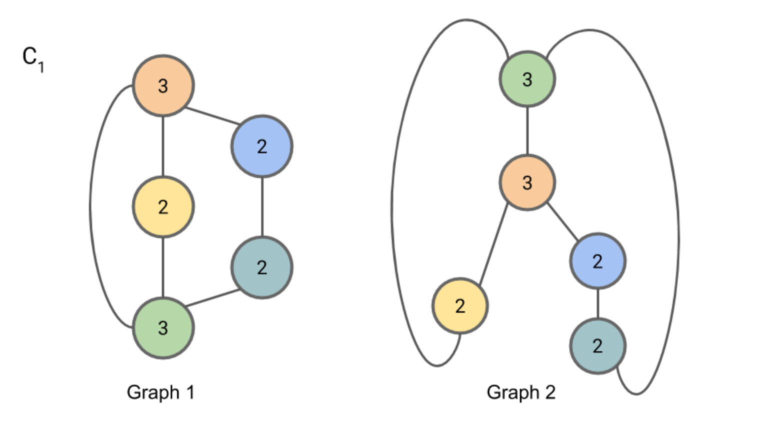
Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Leman Algorithm Example



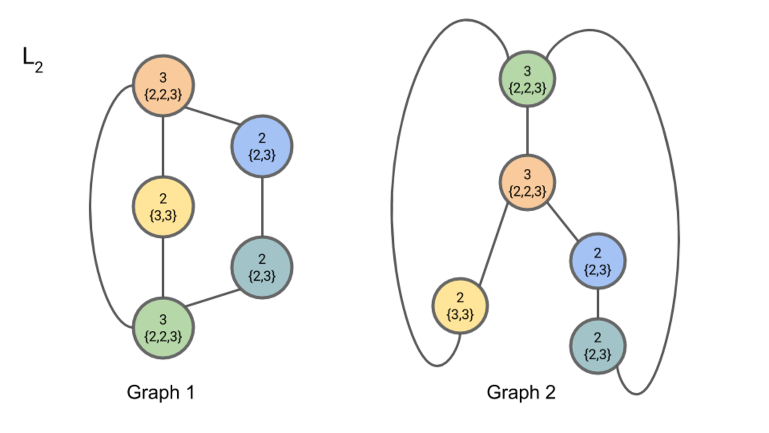
Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Leman Algorithm Example



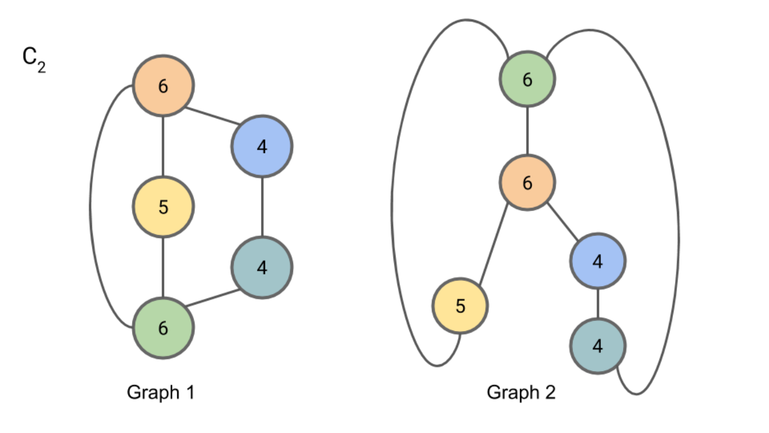
Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Leman Algorithm Example



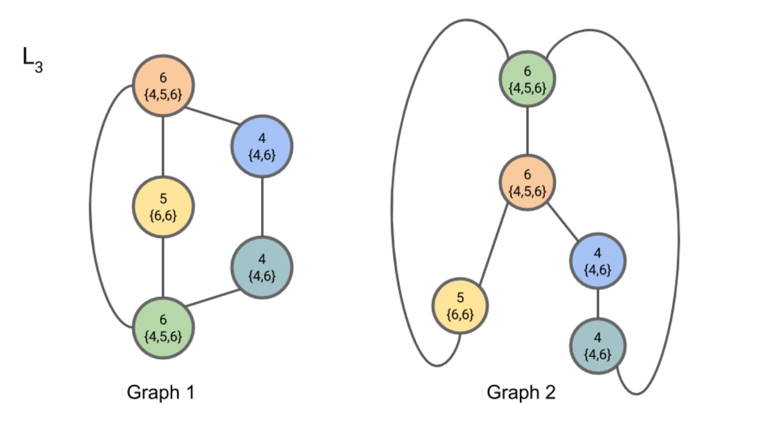
Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Leman Algorithm Example



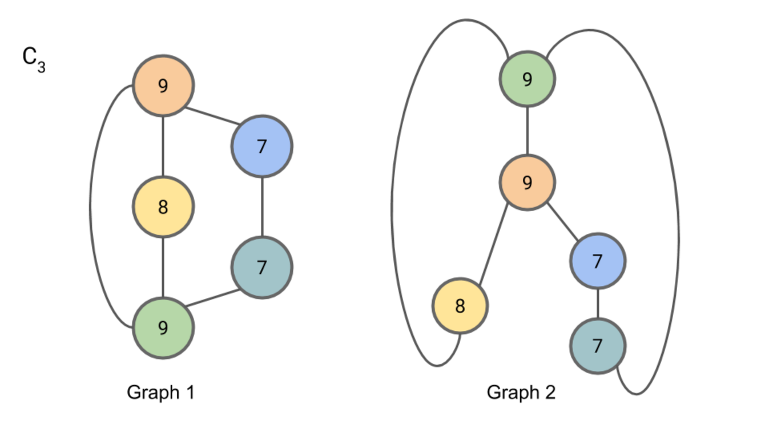
Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Leman Algorithm Example



Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Weisfeiler-Leman Algorithm Example



Source:
https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

Examples of Message Passing

AGGREGATE	COMBINE	Ref
$\text{MAX}\left(\left\{\sigma\left(W_1.x_u^{(k)}\right)\right\}, u \in N(v)\right)$	$W_2.\left[x_v^{(k)}, a_v^{(k+1)}\right]$	GraphSAGE[HYL17]
$W_1.\text{MEAN}\left(x_u^{(k)}, u \in N(v) \cup \{v\}\right)$	$\sigma\left(\left\{W_2.a_v^{(k+1)}\right\}\right)$	GCN[KW08]

k-Weisfeiler-Leman Algorithm

Algorithm k-Weisfeiler-Leman (k-WL)

1: **Input**: (V, E, X_V) Here, $x_v \in \mathbb{Z}_2^d$
2: $c\left(\overrightarrow{v}\right) = c^{(0)}\left(\overrightarrow{v}\right) \longleftarrow hash\left(x_{\overrightarrow{v}}\right)$
3: **while** $c^{(t)}\left(\overrightarrow{v}\right) = c^{(t+1)}\left(\overrightarrow{v}\right) \forall \overrightarrow{v} \in V^k$ **do**
4: $\quad c_i^{(t+1)}\left(\overrightarrow{v}\right) \longleftarrow \{\!\{c^{(t)}\left(\overrightarrow{w}\right) : w \in N_i\left(\overrightarrow{v}\right)\}\!\} \forall \overrightarrow{v} \in V^k$
5: $\quad c^{(t+1)}\left(\overrightarrow{v}\right) \longleftarrow \text{hash}\left(c^{(t)}\left(\overrightarrow{v}\right), c_1^{(t+1)}\left(\overrightarrow{v}\right), \dots, c_k^{(t+1)}\left(\overrightarrow{v}\right)\right) \forall \overrightarrow{v} \in V^k$
6: **Output:** $c^{(T)}\left(\overrightarrow{v}\right) \forall \overrightarrow{v} \in V^k$

---

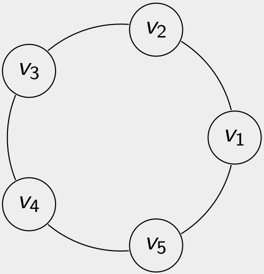
where $\text{hash}(x_{\overrightarrow{v}}) = \text{hash}(x_{\overrightarrow{w}})$ iff **(a)** $x_{v_i} = x_{w_i}$ and **(b)** $(v_i, v_j) \in E$ iff $(w_i, w_j) \in E$.
N.B.: $N_i\left(\overrightarrow{v}\right) = \{(v_1, \dots, v_{i-1}, u, v_{i+1}, \dots, v_k) : u \in V\}$.

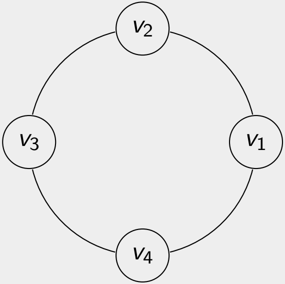
Weisfeiler-Leman Hierarchies

If two graphs have different colorings, then the graphs are not isomorphic. Test is inconclusive if coloring of graphs is the same[MBHSL19]

Weisfeiler-Leman Hierarchies

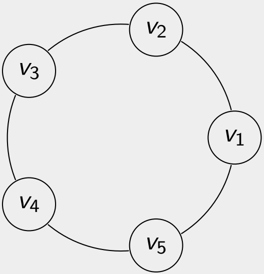
If two graphs have different colorings, then the graphs are not isomorphic. Test is inconclusive if coloring of graphs is the same[MBHSL19]
WL Test fails to distinguish the following graphs, whereas 5-WL does not:

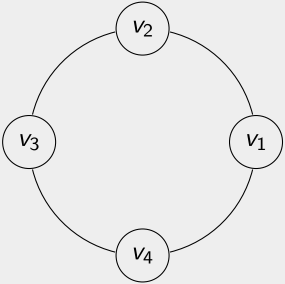




Weisfeiler-Leman Hierarchies

If two graphs have different colorings, then the graphs are not isomorphic. Test is inconclusive if coloring of graphs is the same[MBHSL19]
WL Test fails to distinguish the following graphs, whereas 5-WL does not:

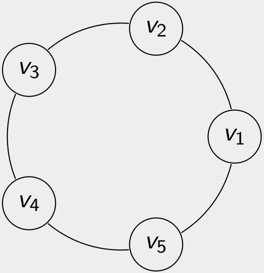


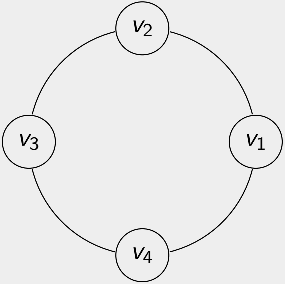


k-WL is strictly weaker than $(k+1)$-WL[HV21]

Weisfeiler-Leman Hierarchies

If two graphs have different colorings, then the graphs are not isomorphic. Test is inconclusive if coloring of graphs is the same[MBHSL19]
WL Test fails to distinguish the following graphs, whereas 5-WL does not:

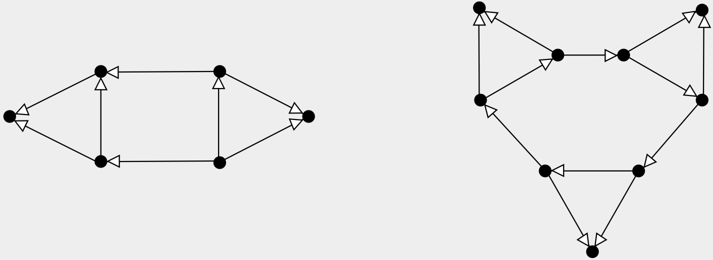




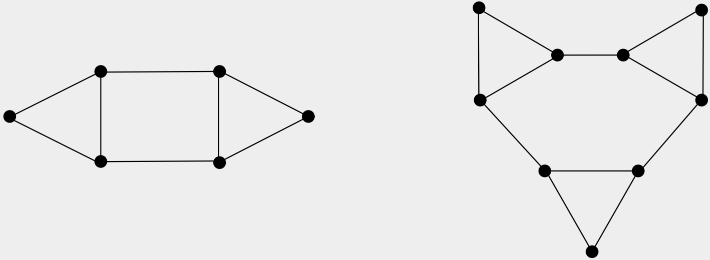
> k-WL is strictly weaker than $(k+1)$-WL[HV21]

However.. for every k, there is an infinite family of graphs for which the k-WL test fails[CFI92]

Simplicial Set Weisfeiler-Leman Algorithm



Simplicial Set Weisfeiler-Leman Algorithm



Directed WL Test

Directed WL Test

Algorithm Directed Weisfeiler-Leman (DWL)[MG21]

1: **Input**: (V, E, X_V) Here, $x_v \in \mathbb{Z}_2^d$

2: $c(v) = c^{(0)}(v) \longleftarrow hash(x_v)$

3: **while** $c^{(t)}(v) = c^{(t+1)}(v)\ \forall v \in V$ **do**

4: $c^{(t+1)}(v) \longleftarrow \text{hash}\begin{pmatrix} c^{(t)}(v), \{\{c^{(t)}(w) : w \in N_{\text{in}}(v)\}\}, \\ \{\{c^{(t)}(u) : w \in N_{\text{out}}(u)\}\} \end{pmatrix}$

5: **Output:** $c^{(T)}(v)\ \forall v \in V$

DWL is strictly stronger than WL[BFW$^+$21]

Summary of WL Zoo

Summary:

$$
\begin{array}{ccc}
DWL & \sqsubset & WL \\
 & & \sqcup \\
 & & k\text{-WL} \\
 & & \sqcup \\
 & & SWL
\end{array}
$$

Summary of WL Zoo

Simplicial Set WL: $c^{t+1}(\sigma) =$hash of coloring at t of σ and coloring of its i-th boundaries, i-th coboundaries, i-th upper-neighbors and i-th lower-neighbors for $0 \leq i \leq k$ whenever $\sigma \in X_k$

Summary:

$$\begin{array}{ccc} DWL & \sqsubset & WL \\ \sqcup & & \sqcup \\ ? & & k\text{-WL} \\ \sqcup & & \sqcup \\ SSWL & \sqsubset & SWL \end{array}$$