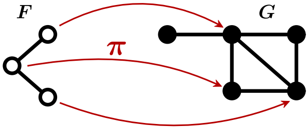
ICML | 2020

Graph Homomorphism Convolution

Universality Results of Graph Homomorphism Densities



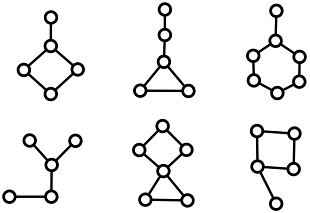
Hoang NT and Takanori Maehara

RIKEN Center for Advanced Intelligence Project

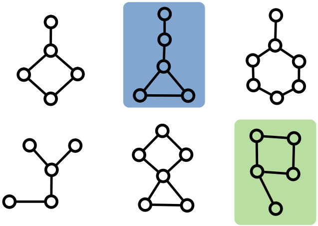
July 15, 2020

Outline

Introduction

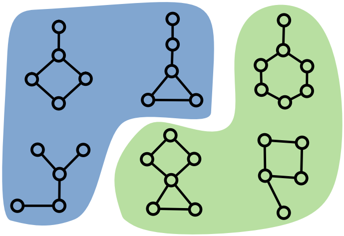


Graph dataset:
$\mathcal{G} := \{G_i, x_i, y_i\}_{i=1}^{n}$,
$G_i = (V(G_i), E(G_i))$ is simple.



Graph dataset:
$\mathcal{G} := \{G_i, x_i, y_i\}_{i=1}^{n}$,
$G_i = (V(G_i), E(G_i))$ is simple.

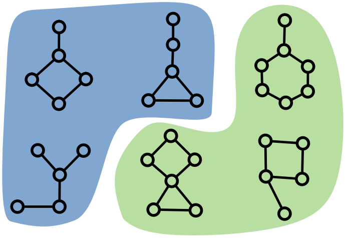
Training set: $\mathcal{G}_{\text{train}} \subset \mathcal{G}$



Graph dataset:
$\mathcal{G} := \{G_i, x_i, y_i\}_{i=1}^{n}$,
$G_i = (V(G_i), E(G_i))$ is simple.

Training set: $\mathcal{G}_{\text{train}} \subset \mathcal{G}$

Find a general hypothesis
$h(G_i, x_i) \approx y_i$.



Graph dataset:
$\mathcal{G} := \{G_i, x_i, y_i\}_{i=1}^{n}$,
$G_i = (V(G_i), E(G_i))$ is simple.

Training set: $\mathcal{G}_{\text{train}} \subset \mathcal{G}$

Find a general hypothesis
$h(G_i, x_i) \approx y_i$.

Applications: Molecule classification, social network classification, point-cloud classification, etc.

! *Common approach*: $\rho : (G, x) \mapsto \rho(G, x) \in \mathbb{R}^d$.

A key property of ρ: Invariant or equivariant to some permutation groups (e.g. invariant to graph isomorphism).

❗ A key property of ρ: Invariant or equivariant to some permutation groups (e.g. invariant to graph isomorphism).

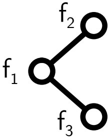
Kernels (2000s)	**Neural Networks (2010s)**
Based on counting or graph distances	Based on message passing or feature aggregation
Subgraph kernel [GFW03] RW kernel [BOS+05] Tree kernel [CD02, MV09] WL kernel [SSL+11]	GIN [XHLJ19] GraphSAGE [HYL17] SplineCNN [FLWM18] GNT kernel [DHP+19]
Invariant by definition.	Invariant by summation of feature aggregations.

A key property of ρ: Invariant or equivariant to some permutation groups (e.g. invariant to graph isomorphism).

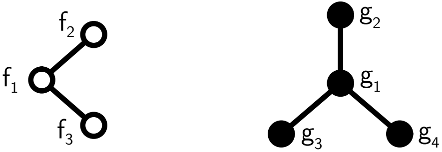
Kernels (2000s)	**Neural Networks (2010s)**
Based on counting or graph distances	Based on message passing or feature aggregation
Subgraph kernel [GFW03] RW kernel [BOS+05] Tree kernel [CD02, MV09] WL kernel [SSL+11]	GIN [XHLJ19] GraphSAGE [HYL17] SplineCNN [FLWM18] GNT kernel [DHP+19]
Invariant by definition.	Invariant by summation of feature aggregations.

We study *graph homomorphism* count vectors as ρ.

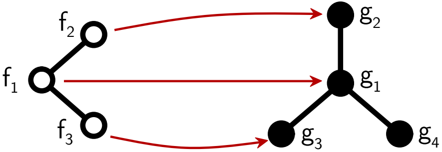
$\boldsymbol{F}$

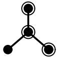


$$\boldsymbol{F} \longmapsto \boldsymbol{G}$$

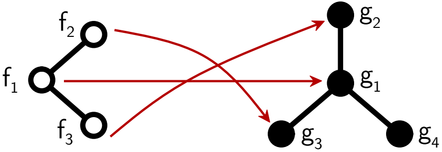


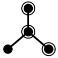
$$\boldsymbol{F} \longmapsto \boldsymbol{G}$$



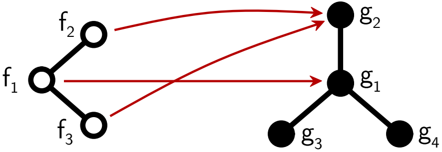
π	f_1	f_2	f_3
	g_1	g_2	g_3

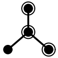
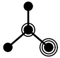
$$F \longmapsto G$$



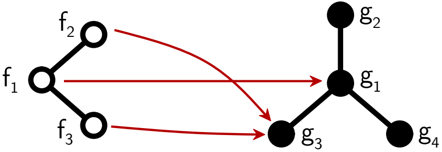
π	f_1	f_2	f_3	
	g_1 g_1	g_2 $\cdots$	g_3 $\cdots$	**6**

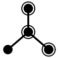
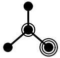
$\boldsymbol{F} \longmapsto \boldsymbol{G}$

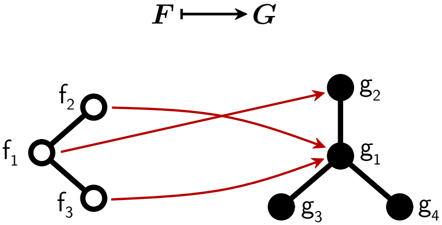


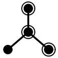
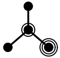
π	f_1	f_2	f_3	
	g_1 g_1	g_2 $\cdots$	g_3 $\cdots$	**6**
	g_1	g_2	g_2	

$$\boldsymbol{F} \longmapsto \boldsymbol{G}$$

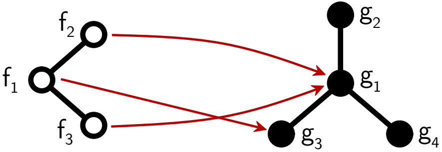


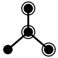
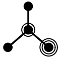
π	f_1	f_2	f_3	
	g_1 g_1	g_2 $\cdots$	g_3 $\cdots$	**6**
	g_1 g_1	g_2 $\cdots$	g_2 $\cdots$	**3**



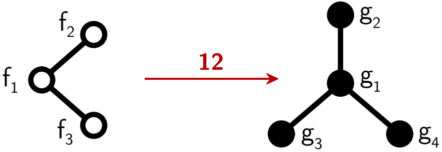
π	f_1	f_2	f_3	
	g_1 g_1	g_2 $\cdots$	g_3 $\cdots$	**6**
	g_1 g_1	g_2 $\cdots$	g_2 $\cdots$	**3**
	g_2	g_1	g_1	

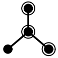
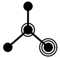
$F \longmapsto G$

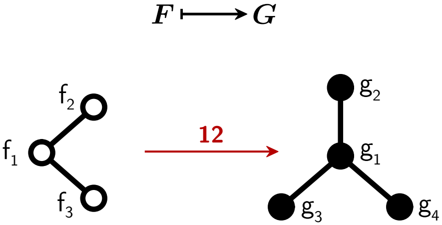


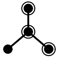
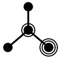
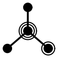
π	f_1	f_2	f_3	
	g_1 g_1	g_2 $\cdots$	g_3 $\cdots$	**6**
	g_1 g_1	g_2 $\cdots$	g_2 $\cdots$	**3**
	g_2 $\cdots$	g_1 g_1	g_1 g_1	**3**

$F \longmapsto G$



π	f_1	f_2	f_3	
	g_1 g_1	g_2 $\cdots$	g_3 $\cdots$	**6**
	g_1 g_1	g_2 $\cdots$	g_2 $\cdots$	**3**
	g_2 $\cdots$	g_1 g_1	g_1 g_1	**3**



π	f_1	f_2	f_3	
	g_1 g_1	g_2 $\cdots$	g_3 $\cdots$	**6**
	g_1 g_1	g_2 $\cdots$	g_2 $\cdots$	**3**
	g_2 $\cdots$	g_1 g_1	g_1 g_1	**3**

The set of all *connectivity preserving* maps is denoted $\mathrm{Hom}(F, G)$.
Here we have $\mathrm{hom}(F, G) := |\mathrm{Hom}(F, G)| = 12$.

$$\hom(F,G) := \sum_{\pi : V(F) \mapsto V(G)} \prod_{(u,v) \in E(F)} 1[(\pi(u), \pi(v)) \in E(G)]$$

$$\mathrm{hom}(F, G) := \sum_{\pi : V(F) \mapsto V(G)} \prod_{(u,v) \in E(F)} 1[(\pi(u), \pi(v)) \in E(G)]$$

All possible maps

$$\hom(F, G) := \underbrace{\sum_{\pi : V(F) \mapsto V(G)}}_{\text{All possible maps}} \prod_{(u,v) \in E(F)} \underbrace{1[(\pi(u), \pi(v)) \in E(G)]}_{\text{Iverson bracket}}$$

$$\mathrm{hom}(F,G) := \underbrace{\sum_{\pi : V(F) \mapsto V(G)}}_{\text{All possible maps}} \overbrace{\prod_{(u,v)\in E(F)}}^{\text{Return 1 for each valid map}} \underbrace{1[(\pi(u),\pi(v)) \in E(G)]}_{\text{Iverson bracket}}$$

Graph Homomorphism | Example – Definition

Return 1 for each valid map

$$\mathsf{hom}(F, G) := \sum_{\pi : V(F) \mapsto V(G)} \prod_{(u,v) \in E(F)} 1[(\pi(u), \pi(v)) \in E(G)]$$

All possible maps

Iverson bracket

For some set of graphs $\mathcal{F}$, the graph *embedding* is given by:

$$G \overset{\mathcal{F}}{\approx} \mathsf{hom}(\mathcal{F}, G) := [\mathsf{hom}(F, G) : F \in \mathcal{F}]$$

Universality Results

Theorem 1 ([Lov67])

Two graphs G_1 and G_2 are isomorphic if and only if hom$(F, G_1) =$ hom(F, G_2) for all simple graphs F. If $|V(G_1)|, |V(G_2)| \leq n$ then we only have to examine F with $|V(F)| \leq n$.

Theorem 1 ([Lov67])

Two graphs G_1 and G_2 are isomorphic if and only if hom$(F, G_1) =$ hom(F, G_2) for all simple graphs F. If $|V(G_1)|, |V(G_2)| \leq n$ then we only have to examine F with $|V(F)| \leq n$.

! This hinted that we can build universal approximators for isomorphism-invariant functions on G via hom$(\mathcal{F}, G)$.

Theorem 1 ([Lov67])

Two graphs G_1 and G_2 are isomorphic if and only if hom$(F, G_1) =$ hom(F, G_2) for all simple graphs F. If $|V(G_1)|, |V(G_2)| \leq n$ then we only have to examine F with $|V(F)| \leq n$.

❗ This hinted that we can build universal approximators for isomorphism-invariant functions on G via hom$(\mathcal{F}, G)$.

❗ Existing universal approximators for graph functions exhibit high dimensionality (high tensor order) [MFSL19, KP19].

Note that homomorphism vector $\hom(\mathcal{F}, G)$ also exhibits high-dimensionality if $\mathcal{F}$ is a set of all simple graphs up to size $|V(G)|$, which is similar to previous research.

Note that homomorphism vector $\hom(\mathcal{F}, G)$ also exhibits high-dimensionality if $\mathcal{F}$ is a set of all simple graphs up to size $|V(G)|$, which is similar to previous research.

- [XHLJ19] proved 1-dim WL-test is implementable by GIN but required $O(|V(G)|)$ feature dimension.

Note that homomorphism vector hom$(\mathcal{F}, G)$ also exhibits high-dimensionality if $\mathcal{F}$ is a set of all simple graphs up to size $|V(G)|$, which is similar to previous research.

- [XHLJ19] proved 1-dim WL-test is implementable by GIN but required $O(|V(G)|)$ feature dimension.
- The line of work by [MFSL19], [KP19] and other teams typically requires tensorization of $O(n^{n^4})$ for $n = |V(G)|$.

Note that homomorphism vector $\mathrm{hom}(\mathcal{F}, G)$ also exhibits high-dimensionality if $\mathcal{F}$ is a set of all simple graphs up to size $|V(G)|$, which is similar to previous research.

- [XHLJ19] proved 1-dim WL-test is implementable by GIN but required $O(|V(G)|)$ feature dimension.
- The line of work by [MFSL19], [KP19] and other teams typically requires tensorization of $O(n^{n^4})$ for $n = |V(G)|$.

❤ Good news is we have the set $\mathcal{F}$ to "parameterize":

Note that homomorphism vector hom$(\mathcal{F}, G)$ also exhibits high-dimensionality if $\mathcal{F}$ is a set of all simple graphs up to size $|V(G)|$, which is similar to previous research.

- [XHLJ19] proved 1-dim WL-test is implementable by GIN but required $O(|V(G)|)$ feature dimension.
- The line of work by [MFSL19], [KP19] and other teams typically requires tensorization of $O(n^{n^4})$ for $n = |V(G)|$.

❤ Good news is we have the set $\mathcal{F}$ to "parameterize":

- Some class of $\mathcal{F}$ can be computed efficiently [DST02]

Note that homomorphism vector $\hom(\mathcal{F}, G)$ also exhibits high-dimensionality if $\mathcal{F}$ is a set of all simple graphs up to size $|V(G)|$, which is similar to previous research.

- [XHLJ19] proved 1-dim WL-test is implementable by GIN but required $O(|V(G)|)$ feature dimension.
- The line of work by [MFSL19], [KP19] and other teams typically requires tensorization of $O(n^{n^4})$ for $n = |V(G)|$.

♥ Good news is we have the set $\mathcal{F}$ to "parameterize":

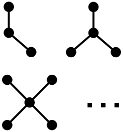
- Some class of $\mathcal{F}$ can be computed efficiently [DST02]
- Restricting $\mathcal{F}$ can lead to different class of universal approximators (this work)

Definition 2 ([BCGR19])

A function $f : \mathcal{G} \mapsto \mathbb{R}$ is $\mathcal{F}$-invariant if $f(G_1) = f(G_2)$ for G_1, G_2 satisfy hom$(\mathcal{F}, G_1) =$ hom$(\mathcal{F}, G_2)$ (termed $\mathcal{F}$-indistinguishable).

Definition 2 ([BCGR19])

A function $f : \mathcal{G} \mapsto \mathbb{R}$ is $\mathcal{F}$-invariant if $f(G_1) = f(G_2)$ for G_1, G_2 satisfy $\hom(\mathcal{F}, G_1) = \hom(\mathcal{F}, G_2)$ (termed $\mathcal{F}$-indistinguishable).

$\mathcal{F}$	$\mathcal{F}$-indistinguishable
	graphs with the same number of vertices
	graphs with the same number of edges
	graphs with the same degree sequence
k-**treewidth**	same results for k-dim WL test [DGR18]

Theorem 3 (Bounded Graphs)

Let f be an $\mathcal{F}$-invariant function. For any positive integer N, there exists a degree N polynomial h_N of $\hom(\mathcal{F}, G)$ such that $f(G) \approx h_N(G)$ for all G with $|V(G)| \leq N$.

Theorem 3 (Bounded Graphs)

Let f be an $\mathcal{F}$-invariant function. For any positive integer N, there exists a degree N polynomial h_N of $\hom(\mathcal{F}, G)$ such that $f(G) \approx h_N(G)$ for all G with $|V(G)| \leq N$.

Theorem 4 (Unbounded Graphs)

Let f be a continuous $\mathcal{F}$-invariant function. There exists a degree N polynomial h_N of $\hom(F, G)$ ($F \in \mathcal{F}$) such that $f(G) \approx h_N(G)$ for all G.

Theorem 3 (Bounded Graphs)

Let f be an $\mathcal{F}$-invariant function. For any positive integer N, there exists a degree N polynomial h_N of $\hom(\mathcal{F}, G)$ such that $f(G) \approx h_N(G)$ for all G with $|V(G)| \leq N$.

Theorem 4 (Unbounded Graphs)

Let f be a continuous $\mathcal{F}$-invariant function. There exists a degree N polynomial h_N of $\hom(F, G)$ ($F \in \mathcal{F}$) such that $f(G) \approx h_N(G)$ for all G.

! The unbounded case only holds for *continous* functions f or in the case G is a *graphon*.

Theorem 3 (Bounded Graphs)

Let f be an $\mathcal{F}$-invariant function. For any positive integer N, there exists a degree N polynomial h_N of $\hom(\mathcal{F}, G)$ such that $f(G) \approx h_N(G)$ for all G with $|V(G)| \leq N$.

Theorem 4 (Unbounded Graphs)

Let f be a continuous $\mathcal{F}$-invariant function. There exists a degree N polynomial h_N of $\hom(F, G)$ ($F \in \mathcal{F}$) such that $f(G) \approx h_N(G)$ for all G.

❗ The unbounded case only holds for *continous* functions f or in the case G is a *graphon*.

❗ The main idea for our proofs is to unify these theorems under the Stone-Weierstrass Theorem.

Experimental Results

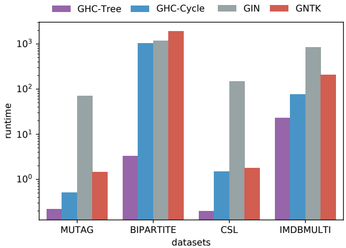
Synthetic graph datasets often used for theoretical studies of graph classification models:

- Circular Skip Links: 150 graphs with 10 isomorphic groups.
- Bipartite: Our synthesized dataset in which random bipartite and random ER graphs are to be distinguished.
- Paulus25: We show the results for strongly regular and co-spectral graphs as a challenge for the future work.

METHODS	CSL	BIPARTITE	PAULUS25
Practical models			
GIN	10.00 ± 0.00	55.75 ± 7.91	7.14 ± 0.00
GNTK	10.00 ± 0.00	58.03 ± 6.84	7.14 ± 0.00
Theory models			
Ring-GNN	$10\sim80 \pm 15.7$	-	-
GHC-Tree	10.00 ± 0.00	52.68 ± 7.15	7.14 ± 0.00
GHC-Cycle	$\mathbf{100.0 \pm 0.00}$	$\mathbf{100.0 \pm 0.00}$	7.14 ± 0.00

We experiment with data from the TU-Dortmund graph kernel benchmark dataset[1].

Methods	MUTAG	IMDB-BIN	IMDB-MUL
Practical models			
GNTK	89.46 ± 7.03	75.61 ± 3.98	51.91 ± 3.56
GIN	89.40 ± 5.60	70.70 ± 1.10	43.20 ± 2.00
PATCHY-SAN	89.92 ± 4.50	71.00 ± 2.20	45.20 ± 2.80
WL kernel	90.40 ± 5.70	73.80 ± 3.90	50.90 ± 3.80
Theory models			
Ring-GNN	-	73.00 ± 5.40	48.20 ± 2.70
GHC-Tree	89.28 ± 8.26	72.10 ± 2.62	48.60 ± 4.40
GHC-Cycles	87.81 ± 7.46	70.93 ± 4.54	47.41 ± 3.67



Since homomorphism vectors are built from $\mathcal{F}$, this method also gives us the ability to look at which sub-structure is important. For these datasets, single vertex, edge, triangle, and star-4 are the top important graphs.

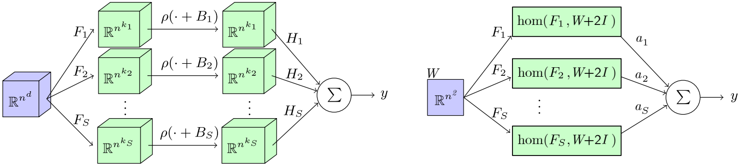
[1] http://graphlearning.io/

Conclusion

Extra Details

In this presentation, few following points are not mentioned:

- Weighted Homomorphism Number: hom$(F, (G, x))$
- Universality results for Weighted Homomorphism
- Invariant/Equivariant Universality results[2]



(a) Tensorized Graph Neural Network (b) Graph Homomorphism Model

[2] `https://arxiv.org/abs/1910.03802`

Remarks and Future Directions

In this work:

- Proposed the study of homomorphism numbers as embeddings for graphs. Our results not only show its general universality, but also the concept of $\mathcal{F}$-invariant universality.
- Proved the universality in unbounded graphs with continuous assumption, and adapted the *graphon* model to prove the universality in general function case.
- Demonstrate the practicality of homomorphism numbers in the graph classification task.

Future:

- To what degree homomorphism vectors generalize other graph kernels?
- Develop learning theory for graphs based on the homomorphism densities.

Thank you for listening!

hoang.nguyen.rh@riken.jp
/gear/graph-homomorphism-network (code + slide)
arxiv.org/pdf/2005.01214.pdf

Jan Böker, Yijia Chen, Martin Grohe, and Gaurav Rattan.
The complexity of homomorphism indistinguishability.
In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel.
Protein function prediction via graph kernels.
Bioinformatics, 21(suppl_1):i47–i56, 2005.

Michael Collins and Nigel Duffy.
Convolution kernels for natural language.
In *Advances in Neural Information Processing Systems*, pages 625–632, 2002.

Holger Dell, Martin Grohe, and Gaurav Rattan.
Lov\'asz meets weisfeiler and leman.
arXiv preprint arXiv:1802.08876, 2018.

Simon S. Du, Kangcheng Hou, Barnabás Póczos, Ruslan Salakhutdinov, Ruosong Wang, and Keyulu Xu.
Graph neural tangent kernel: Fusing graph neural networks with graph kernels.
CoRR, abs/1905.13192, 2019.

Josep Díaz, Maria Serna, and Dimitrios M Thilikos.
Counting h-colorings of partial k-trees.
Theoretical Computer Science, 281(1-2):291–309, 2002.

Matthias Fey, Jan Eric Lenssen, Frank Weichert, and Heinrich Müller.
SplineCNN: Fast geometric deep learning with continuous B-spline kernels.
In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Thomas Gärtner, Peter Flach, and Stefan Wrobel.
On graph kernels: Hardness results and efficient alternatives.
In *Learning theory and kernel machines*, pages 129–143. Springer, 2003.

Will Hamilton, Zhitao Ying, and Jure Leskovec.
Inductive representation learning on large graphs.
In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

Nicolas Keriven and Gabriel Peyré.
Universal invariant and equivariant graph neural networks.
arXiv preprint arXiv:1905.04943, 2019.

László Lovász.
Operations with structures.
Acta Mathematica Hungarica, 18(3-4):321–328, 1967.

Haggai Maron, Ethan Fetaya, Nimrod Segol, and Yaron Lipman.
On the universality of invariant networks.
arXiv preprint arXiv:1901.09342, 2019.

Pierre Mahé and Jean-Philippe Vert.
Graph kernels based on tree patterns for molecules.
Machine learning, 75(1):3–35, 2009.

Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt.
Weisfeiler-lehman graph kernels.
Journal of Machine Learning Research, 12(Sep):2539–2561, 2011.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka.
How powerful are graph neural networks?
International Conference on Learning Representations, 2019.