Graph neural networks for combinatorial optimization problems

Soledad Villar

based on work with Afonso Bandeira, Joan Bruna, Zhengdao Chen, Lei Chen, Alex Nowak, Weichi Yao

Center for Data Science
Courant Institute of Mathematical Sciences

NEW YORK UNIVERSITY

Using Physical Insights for Machine Learning
IPAM, UCLA, November 21 2019

CNNs: state of the art in image processing

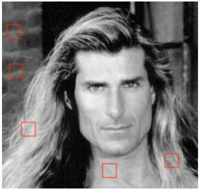
CNNs exploit local invariances of data (compositionality).

LeCun, Bengio, Hinton. 2015

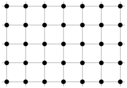
Bronstein, Bruna, LeCun, Szlam, Vandergheynst. 2016

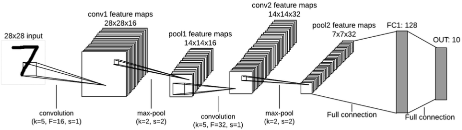
CNNs: state of the art in image processing

CNNs exploit local invariances of data (compositionality).



- locality
- stationarity
- local stationarity
- multi scale features





Data: low dimensional structure in high dimensional space
exploiting of symmetries allows for breaking of curse of dimension.

LeCun, Bengio, Hinton. 2015

Bronstein, Bruna, LeCun, Szlam, Vandergheynst. 2016

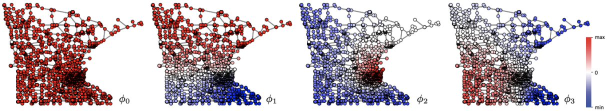
Graphs: beyond Euclidean data

What about other types of data? manifolds, molecules, natural language

What are natural structures? graphs are a general abstraction

What are the relevant invariances? label permutations

What are natural neural network structures?



Bronstein, Bruna, LeCun, Szlam, Vandergheynst. 2016

Graph neural networks

- Computational chemistry
  - drug discovery
- Social networks
  - community detection
  - identifying fake news
- Natural language processing
- Computer vision
- ...

github.com/thunlp/GNNPapers

Must-read papers on GNN

GNN: graph neural network

Contributed by Jie Zhou, Ganqu Cui, Zhengyan Zhang and Yushi Bai.

Content

1. Survey	
2. Models	
2.1 Basic Models	2.2 Graph Types
2.3 Pooling Methods	2.4 Analysis
2.5 Efficiency	
3. Applications	
3.1 Physics	3.2 Chemistry and Biology
3.3 Knowledge Graph	3.4 Recommender Systems
3.5 Computer Vision	3.6 Natural Language Processing
3.7 Generation	3.8 Combinatorial Optimization
3.9 Adversarial Attack	3.10 Graph Clustering
3.11 Graph Classification	3.12 Reinforcement Learning
3.13 Traffic Network	3.14 Few-shot and Zero-shot Learning
3.15 Program Representation	3.16 Social Network

Types of GNNs

- Graph convolutional networks
  - Spectral
  - Spatial (message passing neural networks)
- Graph autoencoders
- Spatial-temporal graph neural networks

Types of GNNs

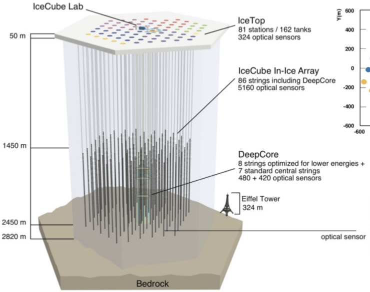
- Graph convolutional networks
  - Spectral
  - Spatial (message passing neural networks)
- Graph autoencoders
- Spatial-temporal graph neural networks

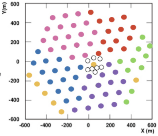
Input: G with A adjacency matrix, X node features, X^e edge features.

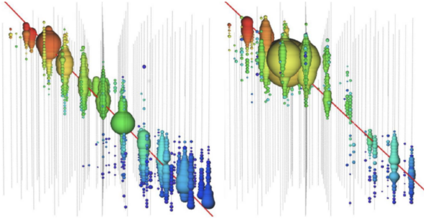
Output: $f(G)$ graph embedding, or solution to optimization problem.

Fundamental property: $f(\pi \cdot G) = \pi \cdot f(G)$

Example: Classifier for IceCube neutrino observatory data







Task:
Neutrino detection (classification neutrino/background)

Data:
Simulated data (neutrino/background)
Simulated IceCube detector

Graph Convolutional Network:
Vertices: sensors
Edges: learned function of the sensors' spatial coordinates

GCN outperforms baseline physical model and 3D CNN.

Choma, Monti, Gerhardt, Palczewski, Ronaghi, Prabhat, Bhimji, Bronstein, Klein, Bruna, IceCube collaboration

This talk

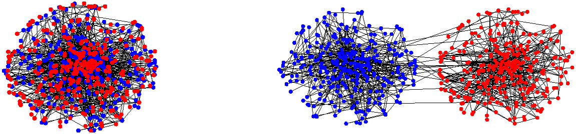
- Motivation / background on different types of GNN (arguably inspired from methods in statistical physics).
- A “natural” way to compare their expressive power.
- Open problems.

Spectral GNN. Motivating example from statistical physics

Clustering the stochastic block model

$A \sim SBM(p, q, n, 2)$ (two equal-sized communities):

$$\mathbb{P}(A_{ij} = 1) = \begin{cases} p & \text{if } i, j \text{ in the same community} \\ q & \text{if } i, j \text{ in different communities} \end{cases}$$

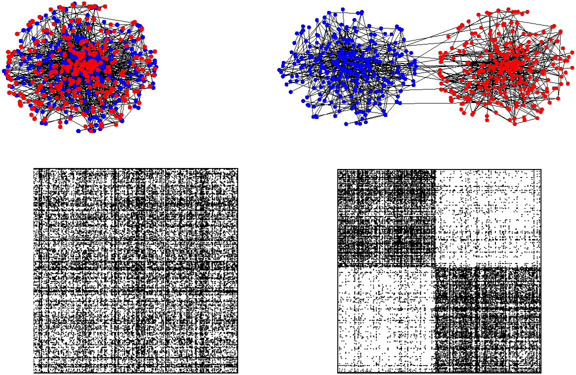


Spectral GNN. Motivating example from statistical physics

Clustering the stochastic block model

$A \sim SBM(p, q, n, 2)$ (two equal-sized communities):

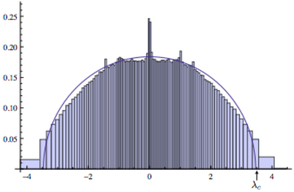
$$\mathbb{P}(A_{ij} = 1) = \begin{cases} p & \text{if } i, j \text{ in the same community} \\ q & \text{if } i, j \text{ in different communities} \end{cases}$$



Clustering the stochastic block model

$A \sim SBM(a/n, b/n, n, 2)$ sparse.
Statistical threshold for detection: $(a-b)^2 > 2(a+b)$.

Spectrum doesn't concentrate (high degree vertices dominate it)
Laplacian is not useful for clustering



Other methods succeed. Example: semidefinite programming.

Krzakala, Moore, Mossel, Neeman, Sly, Zdeborová, Zhang, 2013

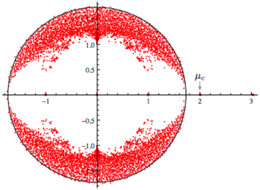
Deshpande, Abbe, Montanari, 2014

Abbe, Bandeira, Hall, 2014

Spectral redemption

Consider the non-backtracking operator (from linearized BP)

$$B_{(i\to j)(i'\to j')} = \begin{cases} 1 \text{ if } j = i' \text{ and } j' \neq i \\ 0 \text{ otherwise} \end{cases}$$



Second eigenvector of B reveals clustering structure

Krzakala, Moore, Mossel, Neeman, Sly, Zdeborová, Zhang, 2013

Bordenave, Lelarge, Massoulie, 2015

Bethe Hessian

$$BH(r) = (r^2 - 1)I - rA + D$$

Fixed points of BP $\longleftrightarrow$ Stationary points of Bethe free energy

Spectrum reveals clustering structure again.

Pitfalls: highly dependent in the model, hard to derive.

What if data doesn't come from a nice model?
Goal: Combine graph operators $I, D, A, \ldots$ to generate robust "data-driven spectral methods" for problems in graphs

---

Saade, Krzakala, Zdeborová, 2014

Graph neural networks

sGNN($\mathcal{M}$)

Power method: $v^{t+1} = Mv^t \qquad t = 1, \ldots, T.$

Scarselli, Tsoi, Hagenbuchner, Monfardini, 2009

Chen, Li, Bruna, 2017

Graph neural networks

sGNN($\mathcal{M}$)

Power method: $v^{t+1} = Mv^t \qquad t = 1, \ldots, T.$

A adjacency matrix. Set $\mathcal{M} = \{I_n, D, A, \min(1, A^2), \ldots, \min(1, A^{2^J})\}$,

$$v^{t+1} = \left(\sum_{M \in \mathcal{M}} M v^t \theta_M \right),$$

with $v^t \in \mathbb{R}^{n \times d_t}$,
$\Theta = \{\theta_1^t, \ldots, \theta_{|\mathcal{M}|}^t\}_t$, $\theta_M^t \in \mathbb{R}^{d_t \times d_{t+1}}$ trainable parameters.

Scarselli, Tsoi, Hagenbuchner, Monfardini, 2009

Chen, Li, Bruna, 2017

Graph neural networks

sGNN($\mathcal{M}$)

Power method: $v^{t+1} = Mv^t \qquad t = 1, \ldots, T.$

A adjacency matrix. Set $\mathcal{M} = \{I_n, D, A, \min(1, A^2), \ldots, \min(1, A^{2^J})\}$,

$$v_l^{t+1} = \quad \left(\sum_{M \in \mathcal{M}} M v^t \theta_{M,l} \right) \,, \; l = 1, \ldots, d_{t+1}$$

with $v^t \in \mathbb{R}^{n \times d_t}$,
$\Theta = \{\theta_1^t, \ldots, \theta_{|\mathcal{M}|}^t\}_t$, $\theta_M^t \in \mathbb{R}^{d_t \times d_{t+1}}$ trainable parameters.

Scarselli, Tsoi, Hagenbuchner, Monfardini, 2009

Chen, Li, Bruna, 2017

Graph neural networks

sGNN($\mathcal{M}$)

Power method: $v^{t+1} = Mv^t \qquad t = 1, \ldots, T.$

A adjacency matrix. Set $\mathcal{M} = \{I_n, D, A, \min(1, A^2), \ldots, \min(1, A^{2^J})\}$,

$$v_l^{t+1} = \left(\sum_{M \in \mathcal{M}} M v^t \theta_{M,l}{}^t \right) , \; l = 1, \ldots, d_{t+1}$$

with $v^t \in \mathbb{R}^{n \times d_t}$,
$\Theta = \{\theta_1^t, \ldots, \theta_{|\mathcal{M}|}^t\}_t$, $\theta_M^t \in \mathbb{R}^{d_t \times d_{t+1}}$ trainable parameters.

Scarselli, Tsoi, Hagenbuchner, Monfardini, 2009
Chen, Li, Bruna, 2017

Graph neural networks

sGNN($\mathcal{M}$)

Power method: $v^{t+1} = Mv^t \qquad t = 1, \ldots, T.$

A adjacency matrix. Set $\mathcal{M} = \{I_n, D, A, \min(1, A^2), \ldots, \min(1, A^{2^J})\}$,

$$v_l^{t+1} = \rho\left(\sum_{M \in \mathcal{M}} M v^t \theta_{M,l}{}^t\right)\ , \ l = 1, \ldots, d_{t+1}$$

with $v^t \in \mathbb{R}^{n \times d_t}$,
$\Theta = \{\theta_1^t, \ldots, \theta_{|\mathcal{M}|}^t\}_t$, $\theta_M^t \in \mathbb{R}^{d_t \times d_{t+1}}$ trainable parameters.

Scarselli, Tsoi, Hagenbuchner, Monfardini, 2009

Chen, Li, Bruna, 2017

Graph neural networks

sGNN($\mathcal{M}$)

Power method: $v^{t+1} = Mv^t \qquad t = 1, \dots, T.$

A adjacency matrix. Set $\mathcal{M} = \{I_n, D, A, \min(1, A^2), \dots, \min(1, A^{2^J})\}$,

$$v_l^{t+1} = \rho\left(\sum_{M \in \mathcal{M}} M v^t \theta_{M,l}{}^t\right), \; l = 1, \dots, d_{t+1}$$

with $v^t \in \mathbb{R}^{n \times d_t}$,
$\Theta = \{\theta_1^t, \dots, \theta_{|\mathcal{M}|}^t\}_t$, $\theta_M^t \in \mathbb{R}^{d_t \times d_{t+1}}$ trainable parameters.

- Independent of the size of the graph.

Scarselli, Tsoi, Hagenbuchner, Monfardini, 2009
Chen, Li, Bruna, 2017

Graph neural networks

sGNN($\mathcal{M}$)

Power method: $v^{t+1} = Mv^t \qquad t = 1, \ldots, T.$

A adjacency matrix. Set $\mathcal{M} = \{I_n, D, A, \min(1, A^2), \ldots, \min(1, A^{2^J})\}$,

$$v_l^{t+1} = \rho\left(\sum_{M \in \mathcal{M}} M v^t \theta_{M,l}{}^t\right), \ l = 1, \ldots, d_{t+1}$$

with $v^t \in \mathbb{R}^{n \times d_t}$,
$\Theta = \{\theta_1^t, \ldots, \theta_{|\mathcal{M}|}^t\}_t$, $\theta_M^t \in \mathbb{R}^{d_t \times d_{t+1}}$ trainable parameters.

- Independent of the size of the graph.
- Extension to line graph (GNN with non-backtracking).

Scarselli, Tsoi, Hagenbuchner, Monfardini, 2009
Chen, Li, Bruna, 2017

Graph neural networks

sGNN($\mathcal{M}$)

Power method: $v^{t+1} = Mv^t \qquad t = 1, \ldots, T.$

A adjacency matrix. Set $\mathcal{M} = \{I_n, D, A, \min(1, A^2), \ldots, \min(1, A^{2^J})\}$,

$$v_l^{t+1} = \rho\left(\sum_{M \in \mathcal{M}} Mv^t \theta_{M,l}{}^t\right), \; l = 1, \ldots, d_{t+1}$$

with $v^t \in \mathbb{R}^{n \times d_t}$,
$\Theta = \{\theta_1^t, \ldots, \theta_{|\mathcal{M}|}^t\}_t$, $\theta_M^t \in \mathbb{R}^{d_t \times d_{t+1}}$ trainable parameters.

- Independent of the size of the graph.
- Extension to line graph (GNN with non-backtracking).
- Power graph $\min(1, A^t)$ encodes t-hop connectivity in binary matrix.

Scarselli, Tsoi, Hagenbuchner, Monfardini, 2009

Chen, Li, Bruna, 2017

Graph neural networks

sGNN($\mathcal{M}$)

Power method: $v^{t+1} = Mv^t \qquad t = 1, \ldots, T.$

A adjacency matrix. Set $\mathcal{M} = \{I_n, D, A, \min(1, A^2), \ldots, \min(1, A^{2^J})\}$,

$$v_l^{t+1} = \rho\left(\sum_{M \in \mathcal{M}} Mv^t {\theta_{M,l}}^t\right), \; l = 1, \ldots, d_{t+1}$$

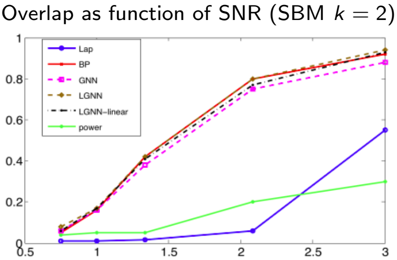
with $v^t \in \mathbb{R}^{n \times d_t}$,
$\Theta = \{\theta_1^t, \ldots, \theta_{|\mathcal{M}|}^t\}_t$, $\theta_M^t \in \mathbb{R}^{d_t \times d_{t+1}}$ trainable parameters.

- Independent of the size of the graph.
- Extension to line graph (GNN with non-backtracking).
- Power graph $\min(1, A^t)$ encodes t-hop connectivity in binary matrix.
- Equivariant wrt permutations $G \mapsto \phi(G)$ then $G_\Pi \mapsto \Pi\phi(G)$.

Scarselli, Tsoi, Hagenbuchner, Monfardini, 2009

Chen, Li, Bruna, 2017

Impressive performance

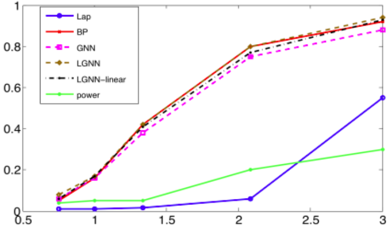


Theoretical result: Under simplifications and assumptions, for linear sGNN, the loss-value gap between local and global minima of the loss function is controlled by the concentration of relevant random matrices around their mean.

Chen, Li, Bruna, 2017

Impressive performance

Overlap as function of SNR (SBM $k = 2$)



Theoretical result: Under simplifications and assumptions, for linear sGNN, the loss-value gap between local and global minima of the loss function is controlled by the concentration of relevant random matrices around their mean.

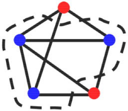
Chen, Li, Bruna, 2017

Extension to unsupervised setting

Max-cut on random regular graphs.

G graph with adjacency A. $\quad Cut(G) \in \{\pm 1\}^n$

$$\mathrm{MaxCut}(G) = \max_{x_i \in \{\pm 1\}} \frac{1}{2} \sum_{i<j} A_{ij}(1 - x_i x_j)$$



Yao, V., Bandeira, 2019

Montanari 2018

Boettcher, Percus, 1999, 2000, 2001

Zdeborová, Boettcher, 2010

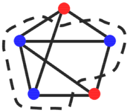
Dembo, Montanari, Sen, 2017

Extension to unsupervised setting

Max-cut on random regular graphs.

G graph with adjacency A. $\quad Cut(G) \in \{\pm 1\}^n$

$$\mathrm{MaxCut}(G) = \max_{x_i \in \{\pm 1\}} \frac{1}{2} \sum_{i<j} A_{ij}(1 - x_i x_j)$$



Methods:

- Goemans Williamson SDP.
- Extremal optimization.
- Graph neural network.
- Adaptation of (asymptotically optimal) message passing for Sherrington-Kirkpatrick Hamiltonian?

Yao, V., Bandeira, 2019
Montanari 2018
Boettcher, Percus, 1999, 2000, 2001

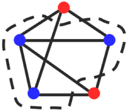
Zdeborová, Boettcher, 2010
Dembo, Montanari, Sen, 2017

Extension to unsupervised setting

Max-cut on random regular graphs.

G graph with adjacency A. $\quad Cut(G) \in \{\pm 1\}^n$

$$\mathrm{MaxCut}(G) = \max_{x_i \in \{\pm 1\}} \frac{1}{2} \sum_{i<j} A_{ij}(1 - x_i x_j)$$



Methods:

- Goemans Williamson SDP.
- Extremal optimization.
- Graph neural network.
- Adaptation of (asymptotically optimal) message passing for Sherrington-Kirkpatrick Hamiltonian?

GNN. Ground truth is not known (unsupervised learning).
Loss: weighted cut value or expected value over batch (policy gradient).

We can compare to asymptotically optimal value:

$$\mathrm{MaxCut}(G^{\mathrm{Reg}}(n, d)) = n\left(\frac{d}{4} + P_* \sqrt{\frac{d}{4}} + o_d\left(\sqrt{d}\right)\right) + o(n)$$

Yao, V., Bandeira, 2019

Montanari 2018

Boettcher, Percus, 1999, 2000, 2001

Zdeborová, Boettcher, 2010

Dembo, Montanari, Sen, 2017

Another example

Quadratic assignment problem

A, B $n \times n$ matrices. Π : set of $n \times n$ permutation matrices.

$$\text{Quadratic assignment : } \max_{X \in \Pi} \operatorname{Trace}(AXBX^\top)$$

Another example

Quadratic assignment problem

A, B $n \times n$ matrices. Π : set of $n \times n$ permutation matrices.

$$\text{Quadratic assignment} : \max_{X \in \Pi} \text{Trace}(AXBX^\top)$$

It includes many relevant problem as particular cases:

- Graph matching: $\min_{X \in \Pi} \|AX - XB\|_F^2$

Another example

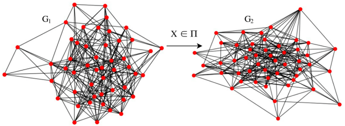
Quadratic assignment problem

A, B $n \times n$ matrices. Π : set of $n \times n$ permutation matrices.

$$\text{Quadratic assignment} : \max_{X \in \Pi} \text{Trace}(AXBX^\top)$$

It includes many relevant problem as particular cases:

- Graph matching: $\min_{X \in \Pi} \|AX - XB\|_F^2 = \|AX\|^2 + \|XB\|^2 - 2\langle AX, XB \rangle$



Another example

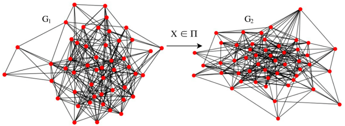
Quadratic assignment problem

A, B $n \times n$ matrices. Π : set of $n \times n$ permutation matrices.

$$\text{Quadratic assignment} : \max_{X \in \Pi} \operatorname{Trace}(AXBX^\top)$$

It includes many relevant problem as particular cases:

- Graph matching: $\min_{X \in \Pi} \|AX - XB\|_F^2 = \|AX\|^2 + \|XB\|^2 - 2\langle AX, XB\rangle$



- Graph isomorphism

Another example

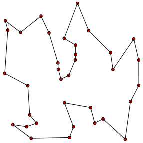
Quadratic assignment problem

A, B $n \times n$ matrices. Π : set of $n \times n$ permutation matrices.

$$\text{Quadratic assignment} : \max_{X \in \Pi} \operatorname{Trace}(AXBX^\top)$$

It includes many relevant problem as particular cases:

- Graph matching: $\min_{X\in\Pi} \|AX - XB\|_F^2$.
- Graph isomorphism.
- Traveling salesman problem.



Another example

Quadratic assignment problem

A, B $n \times n$ matrices. Π : set of $n \times n$ permutation matrices.

$$\text{Quadratic assignment} : \max_{X \in \Pi} \text{Trace}(AXBX^\top)$$

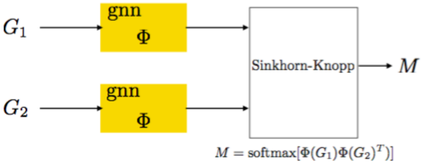
It includes many relevant problem as particular cases:

- Graph matching: $\min_{X \in \Pi} \|AX - XB\|_F^2$.
- Graph isomorphism.
- Traveling salesman problem.
- Gromov-Hausdorff distance of finite metric spaces.

It is NP-hard, even to approximate it.

GNN approach to quadratic assignment

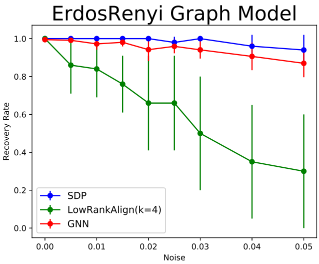
Siamese neural network:

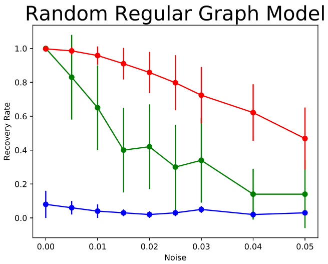


$G_2 = \pi \cdot G_1 \oplus N \qquad N \sim$ i.i.d. bit flip
$G_1 \sim$ Erdos-Renyi
$G_1 \sim$ Random regular

Nowak, V., Bandeira, Bruna, 2017

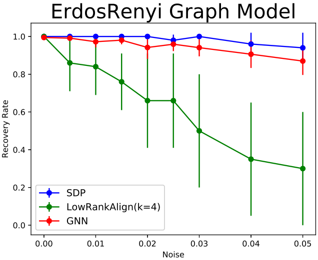
Performance at quadratic assignment

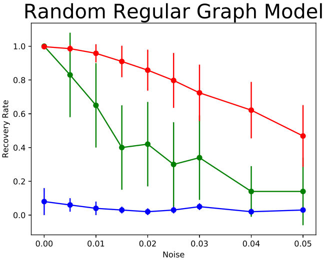


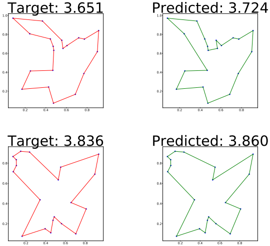


Nowak, V., Bandeira, Bruna, 2017

Performance at quadratic assignment







Nowak, V., Bandeira, Bruna, 2017

Recap

So far we have discussed

- Graph convolutional networks.
  - Application in neutrino detection (IceCube observatory data).
- Spectral GNNs as generalized spectral operators.
  - Clustering the stochastic block model.
  - Max-cut
  - Quadratic assignment (graph matching/traveling salesman).

Recap

So far we have discussed

- Graph convolutional networks.
  - Application in neutrino detection (IceCube observatory data).
- Spectral GNNs as generalized spectral operators.
  - Clustering the stochastic block model.
  - Max-cut
  - Quadratic assignment (graph matching/traveling salesman).

Next

- Message passing neural Networks.
- Permutation invariant linear layers.
- Comparison of classes of graph neural networks.
- Open problems.

Message passing neural network (MPNN)

Another GNN formulation

$$a_v^{(k)} = \text{AGGREGATE}^{(k)}\left(\{h_u^{(k-1)} : u \in \mathcal{N}(u)\}\right)$$

$$h_v^{(k)} = \text{COMBINE}^{(k)}\left(h_v^{(k-1)}, a_v^{(k)}\right)$$

Hamilton, Ying, Leskovec, 2017

Message passing neural network (MPNN)

Another GNN formulation

$$a_v^{(k)} = \text{AGGREGATE}^{(k)}\left(\{h_u^{(k-1)} : u \in \mathcal{N}(u)\}\right)$$

$$h_v^{(k)} = \text{COMBINE}^{(k)}\left(h_v^{(k-1)}, a_v^{(k)}\right)$$

- GNN essential property:
  - invariance or equivariance with respect to permutations
  - node labels are not intrinsic

Hamilton, Ying, Leskovec, 2017

How powerful are graph neural networks?

Q: How good are they at distinguishing non-isomorphic graphs?

Xu, Hu, Leskovec, Jekelga 2019

How powerful are graph neural networks?

Q: How good are they at distinguishing non-isomorphic graphs?

A: MPNN can be as powerful as the Weisfeler-Leman test (1968).
W-L test is as powerful as the LP relaxation (Ullman et al 1994).

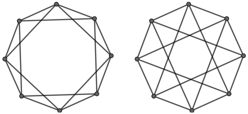
Xu, Hu, Leskovec, Jekelga 2019

How powerful are graph neural networks?

Q: How good are they at distinguishing non-isomorphic graphs?

A: MPNN can be as powerful as the Weisfeler-Leman test (1968).
W-L test is as powerful as the LP relaxation (Ullman et al 1994).

In particular MPNN cannot distinguish between non-isomorphic regular graphs with the same degree.



Xu, Hu, Leskovec, Jekelga 2019

Weisfeler-Leman test

Given (G, X) labeled graph

- $L_0(v) = X(v)$
- At each iteration

$$L_{t+1}(v) = \text{hash}\left(L_t(v), \{\{L_t(w) : w \sim v\}\}\right)$$

Extension to labels in k-tuples (k-WL).

GNN formulation based on this test.

Invariant and equivariant functions on graphs

- Linear case:
  - If $L : \mathbb{R}^{n^k} \to \mathbb{R}$ invariant, then $vec(L) = \pi^{\otimes k} vec(L)$.
  - If $L : \mathbb{R}^{n^k} \to \mathbb{R}^{n^k}$ equivariant, then $vec(L) = \pi^{\otimes 2k} vec(L)$

  - The space of invariant [equivariant] linear functions on k-tensors has dimension $b(k)$ [$b(2k)$].

    ($b(k)$ denotes Bell Number: number of partitions of a size k set).

Maron, Ben-Hamu, Shamir, Lipman, 2019

Maron, Fetaya, Segol, Lipman, 2019

Keriven, Peyré, 2019

Invariant and equivariant functions on graphs

- Linear case:
  - If $L : \mathbb{R}^{n^k} \to \mathbb{R}$ invariant, then $vec(L) = \pi^{\otimes k} vec(L)$.
  - If $L : \mathbb{R}^{n^k} \to \mathbb{R}^{n^k}$ equivariant, then $vec(L) = \pi^{\otimes 2k} vec(L)$

  - The space of invariant [equivariant] linear functions on k-tensors has dimension $b(k)$ [$b(2k)$].
    ($b(k)$ denotes Bell Number: number of partitions of a size k set).

- Universal approximation:
  - Invariant graph networks (IGNs) constructed by composition of linear invariant layers $L_t : \mathbb{R}^{n^k \times a} \to \mathbb{R}^b$ with ReLU or sigmoid activation functions universally approximate the space of invariant functions.
  - Extension to equivariant functions.

Maron, Ben-Hamu, Shamir, Lipman, 2019

Maron, Fetaya, Segol, Lipman, 2019

Keriven, Peyré, 2019

Invariant and equivalent functions on graphs

- Linear case:
  - If $L : \mathbb{R}^{n^k} \to \mathbb{R}$ invariant, then $vec(L) = \pi^{\otimes k} vec(L)$.
  - If $L : \mathbb{R}^{n^k} \to \mathbb{R}^{n^k}$ equivariant, then $vec(L) = \pi^{\otimes 2k} vec(L)$

  - The space of invariant [equivariant] linear functions on k-tensors has dimension $b(k)$ [$b(2k)$].

    ($b(k)$ denotes Bell Number: number of partitions of a size k set).

- Universal approximation:
  - Invariant graph networks (IGNs) constructed by composition of linear invariant layers $L_t : \mathbb{R}^{n^k \times a} \to \mathbb{R}^b$ with ReLU or sigmoid activation functions universally approximate the space of invariant functions.
  - Extension to equivariant functions.

Arbitrary high order tensors are needed (k-IGNs use k-tensors).

Approximation rates are not known.

Maron, Ben-Hamu, Shamir, Lipman, 2019

Maron, Fetaya, Segol, Lipman, 2019

Keriven, Peyré, 2019

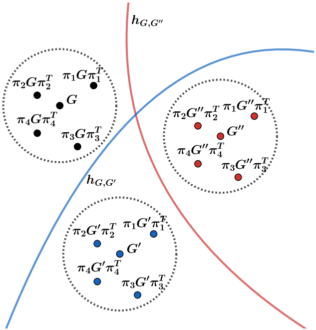
Recap

- We saw many families of (invariant) graph neural networks.
- MPNN not very expressive (cannot distinguish regular graphs).
- k-IGNs universally approximate if k arbitrary large.
- Spectral GNNs we don't know.

Graph isomorphism test

GIso-discriminating class of functions

A class $\mathcal{C}$ of permutation-invariant functions from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$ so that for all pairs $G_1 \not\simeq G_2 \in \mathcal{X}^{n\times n}$, there exists $h \in \mathcal{C}$ such that $h(G_1) \neq h(G_2)$.



Chen, V., Chen, Bruna, 2019

Graph isomorphism equivalence to universal approximation

Universally approximating

A class $\mathcal{C}$ of permutation-invariant functions from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$ so that for all permutation-invariant function f from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$, and for all $\epsilon > 0$, there exists $h_{f,\epsilon} \in \mathcal{C}$ such that $\|f - h_{f,\epsilon}\|_\infty := \sup_{G\in\mathcal{X}^{n\times n}} |f(G) - h_{f,\epsilon}(G)| < \epsilon$

Remark

Universally approximating classes of functions are also GIso-discriminating.

Chen, V., Chen, Bruna, 2019

Graph isomorphism equivalence to universal approximation

$\mathcal{C}^{+L}$

If $\mathcal{C}$ is a collection of functions from $\mathcal{X}^{n\times n}$ to $\mathbb{R}$, consider the set of functions from graphs G to $\mathcal{NN}([h_1(G), ..., h_d(G)])$ for some finite d and $h_1, ..., h_d \in \mathcal{C}$, where $\mathcal{NN}$ is a feed-forward neural network with ReLU and L layers.

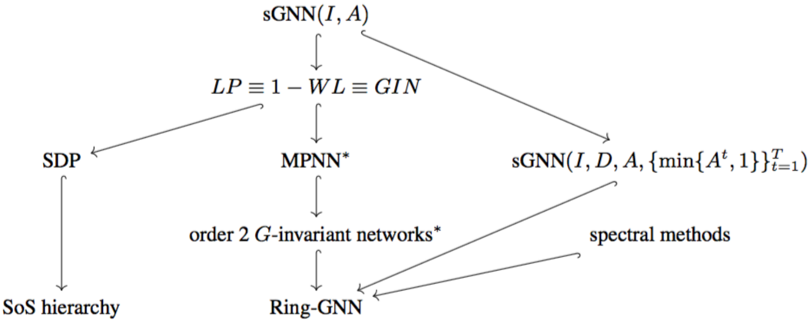
Theorem

If $\mathcal{C}$ is GIso-discriminating $\mathcal{C}^{+2}$ is universally approximating.

Chen, V., Chen, Bruna, 2019

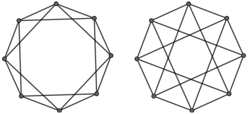
Comparison of classes of functions through GIso

$\mathcal{C} \subseteq \mathcal{C}'$ if for all pairs of non-isomorphic graphs G_1, G_2, if there exists $h \in \mathcal{C}$ so that $h(G_1) \neq h(G_2)$ then there exists $h' \in \mathcal{C}'$ so that $h'(G_1) \neq h'(G_2)$.



Comparison of classes of functions through GIso

- Order-2 graph G-invariant networks cannot distinguish between regular graphs of the same degree.



- Extended the model to RingGNNs which succeed in distinguishing these graphs

Ring GNN

Input: Graph with n nodes and d features: $A \in \mathbb{R}^{n \times n \times d}$.

Equivariant linear layer from $\mathbb{R}^{n \times n \times d}$ to $\mathbb{R}^{n \times n \times d'}$. For $\theta \in \mathbb{R}^{d \times d' \times 17}$:
$L_\theta(A)_{\cdot,\cdot,k'} = \sum_{k=1}^{d} \sum_{i=1}^{15} \theta_{k,k',i} L_i(A_{\cdot,\cdot,i}) + \sum_{i=16}^{17} \theta_{k,k',i} \overline{L}_i$.
Set $A^{(0)} = A$.

$$
\begin{aligned}
B_1^{(t)} &= \rho(L_{\alpha^{(t)}}(A^{(t)})) \\
B_2^{(t)} &= \rho(L_{\beta^{(t)}}(A^{(t)}) \cdot L_{\gamma^{(t)}}(A^{(t)})) \\
A^{(t+1)} &= k_1^{(t)} B_1^{(t)} + k_2^{(t)} B_2^{(t)}
\end{aligned}
$$

where $k_1^{(t)}, k_2^{(t)} \in \mathbb{R}$, $\alpha^{(t)}, \beta^{(t)}, \gamma^{(t)} \in \mathbb{R}^{d^{(t)} \times d'^{(t)} \times 17}$ are learnable parameters.

Scalar output: $\theta_S \sum_{i,j} A_{ij}^{(T)} + \theta_D \sum_{i,i} A_{ii}^{(T)} + \sum_i \theta_i \lambda_i(A^{(T)})$, where $\theta_S, \theta_D, \theta_1, \ldots, \theta_n \in \mathbb{R}$ are trainable parameters, and $\lambda_i(A^{(T)})$ is the i-th eigenvalue of $A^{(T)}$.

Open problems

- Find a scalable GNN model that is invariant and expressive.
  Connect GNN depth/architecture with classes of graphs they separate.

- Can k-IGNs implement k-WL test?

- Optimization landscape of GNNs:
  Current analysis of optimization landscape relies in simplified models to show that all local minima are confined in low-energy configurations.

Maron, Ben-Hamu, Lipman 2019

Extensions

- Extension to fermion-symmetry invariant architectures.
  Antisymmetric wave functions with smallest eigenvalue.

- Connection with Sum of Squares:
  For some classes of "detecting hidden structures problems" existence of degree-d SoS refutations implies success of certain (typically non-explicit) spectral methods.
  - Can we express such class of spectral methods with GNNs.
  - Can we learn them?

Hopkins, Kothari, Potechin, Raghavendra, Schramm, Steurer, 2017

References

Supervised Community Detection with Hierarchical Graph Neural Networks
Z. Chen, L. Li, J. Bruna, ICLR 2019
arXiv:1705.08415

Experimental performance of graph neural networks on random instances of max-cut
Weichi Yao, Afonso S. Bandeira, Soledad Villar, SPIE 2019
arXiv:1908.05767

A note on learning algorithms for quadratic assignment with Graph Neural Networks
A. Nowak, S. Villar, A. S. Bandeira, J. Bruna, PADL 2017
arXiv:1706.07450

On the equivalence between graph isomorphism testing and function approximation with GNNs
Z. Chen, S. Villar, L. Chen, J. Bruna, NeurIPS 2019

arXiv:1905.12560

NSF

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH
UNITED STATES AIR FORCE

EUROPEAN OFFICE AEROSPACE R&D

SIMONS FOUNDATION