Convolutional GNN on Directed Acyclic Graphs

Samuel Rey*, Hamed Ajorlou† and Gonzalo Mateos†

*Dept. of Signal Theory and Communications, Rey Juan Carlos University, Madrid, Spain

†Dept. of Electrical and Computer Eng., University of Rochester, Rochester, USA

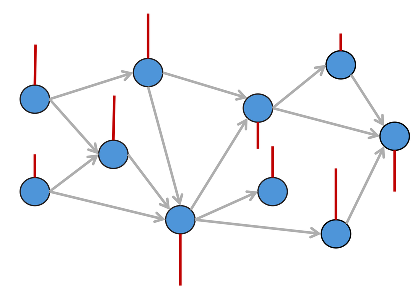
Motivation and context

- Contemporary data is becoming **heterogeneous** and **pervasive**
  - ⇒ Large amounts of data are propelling the development of data-driven methods
- **Graph neural networks (GNNs)** are the tool of choice to learn from network data
  - ⇒ Data is interpreted as signals defined on a graph
  - ⇒ Harness the information encoded in the graph topology to deal with irregular structure

- **Limitation**: most GNNs and graph-based methods focus on undirected graphs
  - ⇒ Accounting for directionality plays an important role but comes with several challenges
  - ⇒ These challenges are exacerbated when dealing with directed acyclic graphs (DAGs)
- **Prior art**: few works are starting to look into learning on DAGs [Zhang19] [Thost20]
  - ⇒ Complex architectures combining attention and sequence processing techniques
- **This work**: design a **DAG-aware convolutional GNN** to learn from data defined on DAGs
  - ⇒ Harness the partial ordering of the DAG to obtain a stronger inductive bias
  - ⇒ Simple architecture with convolution defined in a principled manner

Preliminaries and notation

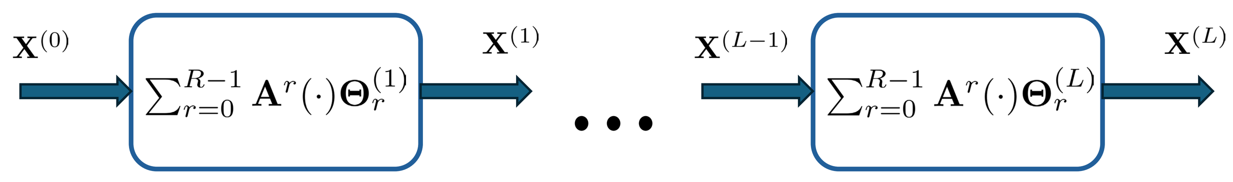
- In a DAG $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ the set of N nodes $\mathcal{V}$ is a **partially ordered set**
  - ⇒ Node j is a *predecessor* of i if $j < i$
  - ⇒ Meaning that there is a direct path from j to i
  - ⇒ Some nodes are not comparable, i.e., $i \not\leq j$ and $j \not\leq i$
- Define a signal $\mathbf{x} \in \mathbb{R}^N$ on top of the graph
  - ⇒ x_i = Signal value at node i



- The acyclicity and the order of $\mathcal{V}$ render the adjacency $\mathbf{A} \in \mathbb{R}^{N\times N}$ strictly lower-triangular
  - ⇒ $A_{ij} \neq 0$ if and only if there is an edge from j to i
- A **convolutional GNN** is a parametric function given by the recursion

$$\mathbf{X}^{(\ell+1)} = \sigma\left(\sum_{r=0}^{R-1} \mathbf{A}^r \mathbf{X}^{(\ell)} \boldsymbol{\Theta}_r^{(\ell)}\right) \tag{1}$$

  - ⇒ The aggregation function is driven by the graph topology, $\mathbf{X}^{(0)}$ are the input data
  - ⇒ $\boldsymbol{\Theta}_r^{(\ell)} \in \mathbb{R}^{F_i^{(\ell)} \times F_o^{(\ell)}}$ collects the learnable convolutional filter coefficients



Problem formulation and goal

- **Goal**: design a convolutional GNN tailored to learn from data defined over DAGs
  - ⇒ Given a training set $\mathcal{T} = \{\mathbf{X}_m, \mathbf{y}_m\}_{m=1}^M$ containing M input-output observed signals
- We learn a non-linear parametric mapping $f_{\boldsymbol{\Theta}}(\cdot|\mathcal{D})$ relating $\mathbf{X}_m$ and $\mathbf{y}_m$
  - ⇒ We estimate the weights $\boldsymbol{\Theta}$ by minimizing some loss function of interest $\mathcal{L}$ over $\mathcal{T}$

$$\min_{\boldsymbol{\Theta}} \frac{1}{M}\sum_{m=1}^{M} \mathcal{L}(\mathbf{y}_m, f_{\boldsymbol{\Theta}}(\mathbf{X}_m|\mathcal{D})) \tag{2}$$

Challenges

- The architecture must account for the partially ordered $\mathcal{V}$
- DAGs may encode causal relations, a property we wish to incorporate into our architecture
- The adjacency matrix $\mathbf{A}$ of a DAG is a nilpotent matrix
  - ⇒ This collapsed spectrum deprives us of a spectral interpretation [Seifert23]

Graph shift operators and convolution for DAGs

- We build upon the work from [Seifert23] to compute convolutions in a principled way
- Assume a signal $\mathbf{x}$ can be described by the causes at predecessor nodes $\mathbf{c} \in \mathbb{R}^N$ as $\mathbf{x} = \mathbf{Wc}$
  - ⇒ $\mathbf{W} \in \mathbb{R}^{N\times N}$ is the transitive closure of $\mathcal{D}$ with $W_{ij} \neq 0$ if $j < i$
  - ⇒ We focus on $\mathbf{W} = (\mathbf{I} - \mathbf{A})^{-1}$ closely related to structural equation models
- Every node $k \in \mathcal{V}$ induces a causal GSO given by

$$[\mathbf{T}_k\mathbf{x}]_i = \sum_{j\leq i \text{ and } j\leq k} W_{ij}c_j, \qquad \mathbf{T}_k\mathbf{x} = \mathbf{W}\mathbf{D}_k\mathbf{c} = \mathbf{W}\mathbf{D}_k\mathbf{W}^{-1}\mathbf{x} \tag{3}$$

  - ⇒ Diagonal matrix $\mathbf{D}_k \in \{0,1\}^{N\times N}$ with $[\mathbf{D}_k]_{ii} = 1$ if $i \leq k$
  - ⇒ $\mathbf{W}^{-1}$ is a DAG Fourier transform with causes $\mathbf{c}$ being the spectral coefficients
- The most general shift-invariant DAG filter $\mathbf{H}$ is given by

$$\mathbf{H} = \sum_{k\in\mathcal{V}} h_k\mathbf{T}_k = \mathbf{W}\sum_{k\in\mathcal{V}} h_k\mathbf{D}_k\mathbf{W}^{-1} \tag{4}$$

  - ⇒ Convolution given by $\mathbf{h} * \mathbf{x} = \mathbf{Hx}$ with the frequency response of $\mathbf{H}$ being $\sum_{k\in\mathcal{V}} h_k\mathbf{D}_k$

References

Thost20 V. Thost, J. Chen, "Directed Acyclic Graph Neural Networks", ICLR, 2020.

Zhang19 M. Zhang, S. Jiang, Z. Cui, R. Garnett, Y. Chen, "D-vae: A variational autoencoder for directed acyclic graphs", Neurips, 2019.

Seifert23 B. Seifert, C. Wendler, M. Puschel, "Causal fourier analysis on directed acyclic graphs and posets", IEEE Trans. Signal Process.

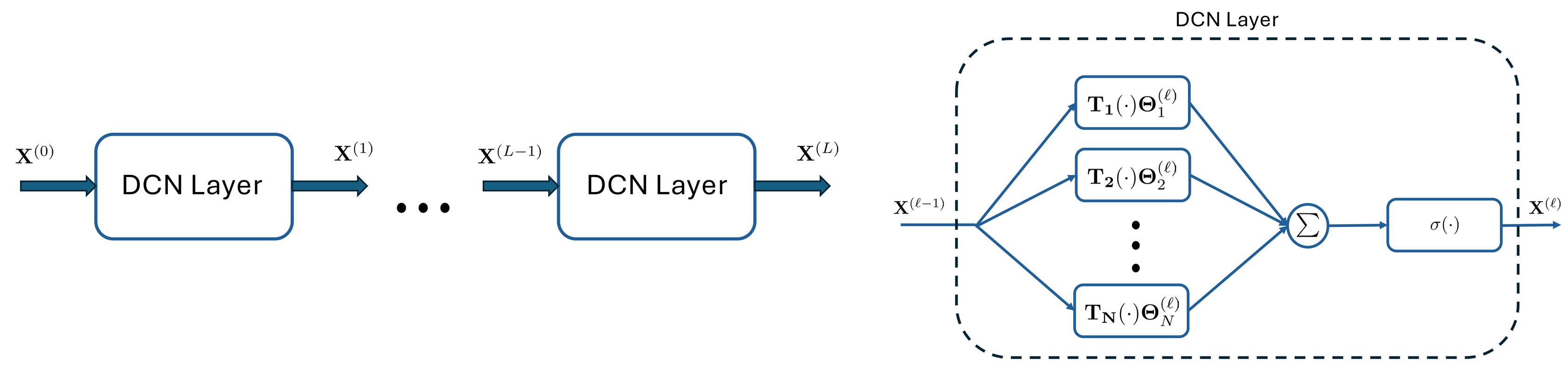
Rey24 S. Rey, H. Ajorlou, G. Mateos, "Convolutional Learning on Directed Acyclic Graphs", arXiv preprint arXiv:2405.03056, 2024.

DAG Convolutional Network (DCN)

- **DCN** concatenates several layers where the convolution is performed using a DAG filter
  - ⇒ We can gain expressive power by replacing the single-filter layer with a filter bank

$$\mathbf{x}^{(\ell+1)} = \sigma\left(\sum_{k\in\mathcal{V}} h_k^{(\ell)}\mathbf{T}_k\mathbf{x}^{(\ell)}\right), \qquad \mathbf{X}^{(\ell+1)} = \sigma\left(\sum_{k\in\mathcal{V}} \mathbf{T}_k\mathbf{X}^{(\ell)}\boldsymbol{\Theta}_k^{(\ell)}\right) \tag{5}$$

  - ⇒ Filter coefficients $h_k^{(\ell)}/\boldsymbol{\Theta}_k^{(\ell)}$ are the learnable parameters
  - ⇒ The causal convolution account for the DAG topology and partial ordering
- **Spectral interpretation**: since $\mathbf{T}_k\mathbf{x}^{(\ell)} = \mathbf{W}\mathbf{D}_k\mathbf{c}^{(\ell)}$ the convolution combines causes from predecessors and diffuses them across the DAG
- **Message passing interpretation**: at every node i each $\mathbf{T}_k$ forms a message combining features from predecessors common to nodes k and i
  - ⇒ Filter coefficients determine how to mix these messages



Desirable features and current limitations

Main advantages

- The DCN is a permutation equivariant model
- The spectrum of $\mathbf{T}_k$ is well defined endowing the DCN with a spectral representation
  - ⇒ Fundamental to analyze properties such as stability, transferability, ...
- The eigenvalues collected in $\mathbf{D}_k$ are binary so no numerical issues are expected
- The GSOs are potentially very sparse matrices since $\text{sup}(\mathbf{T}_k) \subseteq \text{sup}(\mathbf{W})$

Limitations

- The number of learnable parameters grows with the size of the graph
  - ⇒ Potential computational and memory limitations
  - ⇒ **Workaround**: approximate the convolution as $\sum_{k\in\mathcal{U}} h_k\mathbf{T}_k$, where $\mathcal{U} \subset \mathcal{V}$
  - ⇒ Shown to perform well in practice

Numerical evaluation: Synthetic experiments I

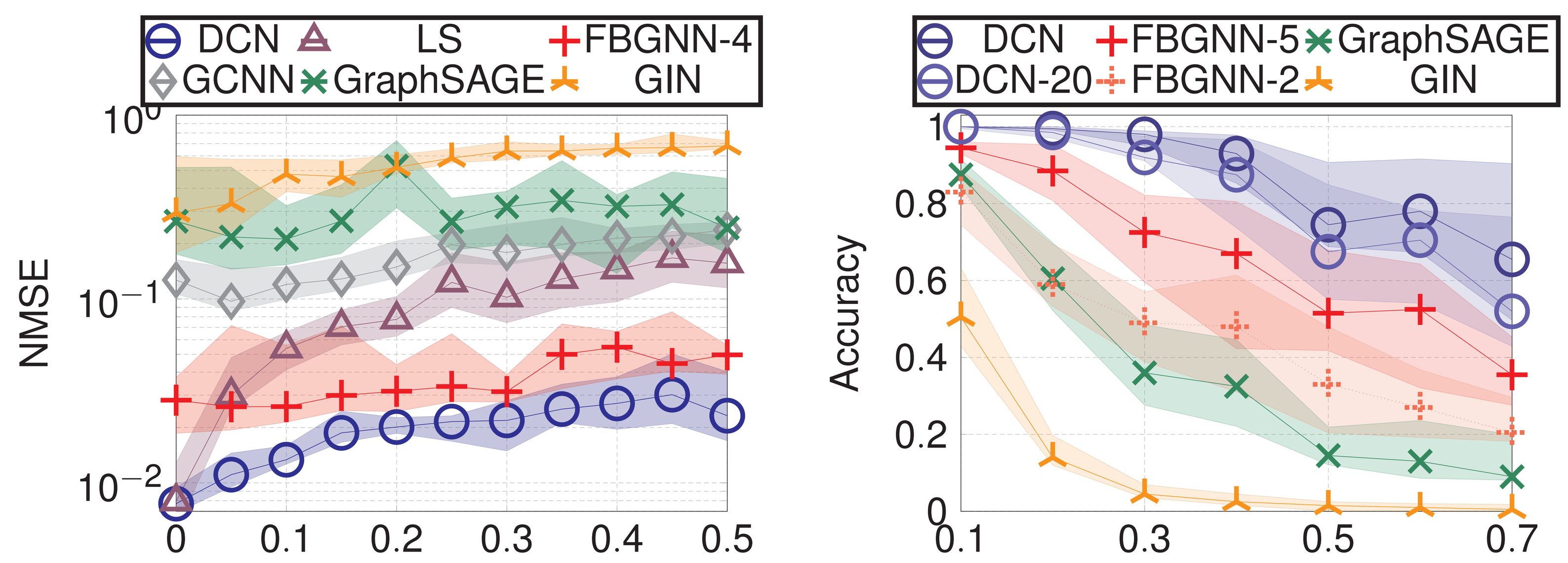
- We test DCN using synthetic data over two different tasks
  - ⇒ Network diffusion: learn to predict the output of a diffusion process given the input
  - ⇒ Source identification: learn to identify source nodes given the output
- ER graphs with $N = 200$ nodes
  - ⇒ Results are the average of 50 iid realizations
- Signals generated following the linear model $\mathbf{y}_m = \mathbf{H}\mathbf{X}_m + \mathbf{w}$, with DAG filter $\mathbf{H}$ and noise $\mathbf{w}$

	Network Diffusion		**Source Identification**	
	MNSE	Time (s)	Accuracy	Time (s)
DCN	$\mathbf{0.016} \pm \mathbf{0.014}$	3.6	0.052 ± 0.014	7.5
DCN-30	$\mathbf{0.029} \pm \mathbf{0.017}$	3.5	0.052 ± 0.016	7.4
DCN-10	0.058 ± 0.021	3.5	0.055 ± 0.015	7.2
DCN-T	0.098 ± 0.024	4.1	$\mathbf{0.991} \pm \mathbf{0.018}$	8.2
DCN-30-T	0.199 ± 0.030	3.7	$\mathbf{0.983} \pm \mathbf{0.032}$	7.64
DCN-10-T	0.229 ± 0.030	3.5	0.865 ± 0.141	7.38
LS	0.050 ± 0.022	0.4	0.05 ± 0.016	0.36
FB-GCNN	0.091 ± 0.028	3.4	0.739 ± 0.172	7.4
GCN	0.167 ± 0.037	3.3	0.155 ± 0.216	7.1
GAT	0.649 ± 0.089	13.8	0.044 ± 0.081	28.4
GraphSAGE	0.359 ± 0.039	5.9	0.676 ± 0.163	12.5
GIN	0.402 ± 0.079	6.0	0.19 ± 0.163	12.5
MLP	0.353 ± 0.039	2.2	0.050 ± 0.016	4.7

- DCN outperforms the baselines in both tasks
  - ⇒ Even when using approximate convolutions with 30/10 GSOs

Numerical evaluation: Synthetic experiments II

- DCN sensitivity to the presence of noise (left) and the sparsity of the DAG (right)



(a) Normalized noise power. (b) Edge Probability

- In the absence of noise DCN results are comparable to that of LS (optimal solution)
  - ⇒ In the presence of noise DCN outperforms the baselines
- Source identification task becomes more challenging as DAGs become denser
  - ⇒ DCN and approximate DCN with 20 GSOs outperform all other alternatives

Link to the paper with code and future research directions

- Evaluate the performance of DCN using real-world data
- Benchmarking against DAG learning models [Zhang19] [Thost20]
- Principled approach to select the subset $\mathcal{U}$ or alternative simplifications
- Establish relevant theoretical properties of the architecture