Convolutional Learning on Directed Acyclic Graphs

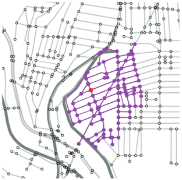
S. Rey*, H. Ajorlou†, G. Mateos†

*King Juan Carlos University - †University of Rochester

Asilomar Conference on Signals, Systems, and Computers - Pacific Grove, USA - Oct 27-Oct 30, 2024

Introduction

- Contemporary data is becoming **heterogeneous** and **pervasive**
  - ⇒ Large amounts of data are propelling data-driven methods



Traffic data

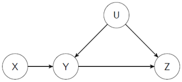
Home automation data

River flow data

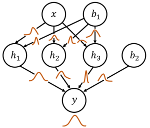
- **GNNs** are the tool of choice to learn from network data
  - ⇒ Data is interpreted as signals defined on a graph
  - ⇒ Harness the graph topology to deal with irregular structure
- GNNs and graph-based methods **focus on undirected graphs**

The impact of directionality

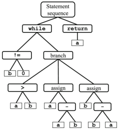
- DAGs are highly structured graphs prevalent across domains



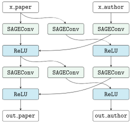
Causal inference



Bayesian nets.



Syntax tree



Neural networks

- Directionality plays an important role when processing information
  - ⇒ Directed graphs present well-known challenges

- These challenges are exacerbated when dealing with DAGs
  - ⇒ Standard architectures fail when learning from DAGs
  - ⇒ Lack of cycles results in nilpotent adjacency matrix
  - ⇒ Deprives us from a spectral interpretation

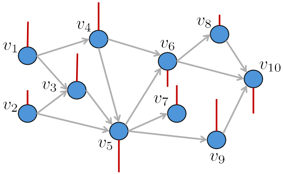
Context and goal

- Developing GNNs to learn from DAGs is drawing attention
  - ⇒ D-VAE: autoencoder to obtain embeddings for DAGs [Zhang19]
  - ⇒ DAGNN: combines sequential message passing with GRU [Thost21]
  - ⇒ DAG+Transformer: adapt transformer layer for DAGs [Luo23]

- **Limitation:** complex architectures based on sequential operations
  - ⇒ Large computational burden and difficult to interpret/analyze

- **Our goal:** design a GNN to learn from **data defined on DAGs**
  - ⇒ Simple architecture based on convolution
  - ⇒ Use the partial ordering to obtain a stronger inductive bias

- DAG $\mathcal{D} = (\mathcal{V}, \mathcal{E})$ the set of N nodes
- $\mathcal{V}$ is a **partially ordered set**
  - $\Rightarrow$ Node j is a *predecessor* of i if $j < i$
  - $\Rightarrow$ Nodes are not comparable if $i \not\leq j$ and $j \not\leq i$



- The adjacency $\mathbf{A} \in \mathbb{R}^{N \times N}$ is strictly lower-triangular
  - $\Rightarrow$ $A_{ij} \neq 0$ if and only if there is an edge from i to j
- Define a graph signal $\mathbf{x} \in \mathbb{R}^N$ on top of the graph
  - $\Rightarrow$ $x_i =$ Signal value at node i
- A graph filter is defined as a polynomial $\mathbf{H} = \sum_0^{R-1} h_r \mathbf{A}^r$
  - $\Rightarrow$ $\mathbf{H}$ allow modeling diffusion processes and graph convolution

Fundamental of GNNs

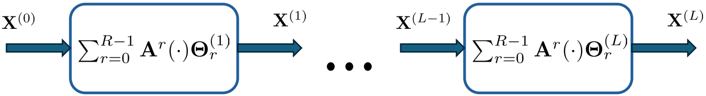
- A **convolutional GNN** is a parametric function $f_{\boldsymbol{\Theta}}(\cdot|\mathbf{A})$
- With $\mathbf{X}^{(0)}$ being the input, the output at the ℓ layer is given by

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

$\Rightarrow$ $\boldsymbol{\Theta}_r^{(\ell)} \in \mathbb{R}^{F_i^{(\ell)} \times F_o^{(\ell)}}$ collects learnable filter coefficients

$\Rightarrow$ Aggregation function driven by graph topology

- Architecture formed by staking several convolutional layers



Universidad Rey Juan Carlos
UNIVERSITY of ROCHESTER

Problem description

- Given training set $\mathcal{T} = \{\mathbf{X}_m, \mathbf{y}_m\}_{m=1}^{M}$ with input-output observations
  - $\Rightarrow$ Learn the non-linear mapping relating $\mathbf{X}_m$ and $\mathbf{y}_m$
  - $\Rightarrow$ Assuming it is well-represented by convolutional GNN $f_{\mathbf{\Theta}}(\cdot|\mathcal{D})$

- We estimate $\mathbf{\Theta}$ by minimizing some loss function of interest $\mathcal{L}$ over $\mathcal{T}$

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

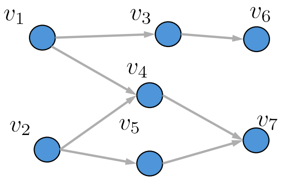
Aim and challenges

- Design a GNN with convolution tailored for DAGS
  - $\Rightarrow$ The architecture must account for the partially ordered $\mathcal{V}$
  - $\Rightarrow$ The architecture must admit a spectral representation

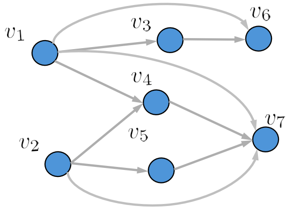
- We compute convolutions over DAGs following the work in [Seifert23]
  - $\Rightarrow$ Principled framework based on causal relations
- Signal $\mathbf{x}$ can be described by causes $\mathbf{c} \in \mathbb{R}^N$ at predecessor nodes as

$$\mathbf{x} = \mathbf{W}\mathbf{c}$$

$\Rightarrow$ $\mathbf{W} \in \mathbb{R}^{N \times N}$ is the transitive closure of $\mathcal{D}$ with $W_{ij} \neq 0$ if $i < j$



DAG from $\mathbf{A}$



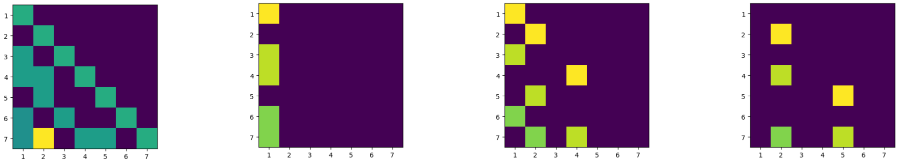
DAG from $\mathbf{W}$

- Shifting the signal $\mathbf{x}$ with respect to node k is given by

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

  - $\Rightarrow$ Every node $k \in \mathcal{V}$ induces a causal GSO
  - $\Rightarrow$ Diagonal matrix $\mathbf{D}_k \in \{0,1\}^{N \times N}$ with $[\mathbf{D}_k]_{ii} = 1$ if $i \leq k$
  - $\Rightarrow$ DAG Fourier Transform $\mathbf{W}^{-1}$ with spectral coefficients $\mathbf{c}$

- Most general shift-invariant DAG filter $\mathbf{H}$ is given by $\mathbf{H} = \sum_{k \in \mathcal{V}} h_k \mathbf{T}_k$



$\mathbf{W}$ $\quad$ $\mathbf{T}_1$ $\quad$ $\mathbf{T}_4$ $\quad$ $\mathbf{T}_5$

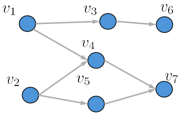
DAG Convolutional Network

- **DCN** leverages the definition of the causal filters tailored for DAGs
- The output at the ℓ-th layer is given by

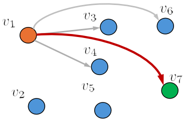
$$\mathbf{x}^{(\ell+1)} = \sigma\left(\sum_{k \in \mathcal{V}} h_k^{(\ell)} \mathbf{T}_k \mathbf{x}^{(\ell)}\right)$$

$\Rightarrow$ Filter coefficients $h_k^{(\ell)}$ are the learnable parameters

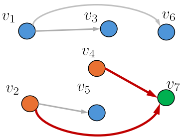
$\Rightarrow$ Causal GSO account for the DAG topology and partial ordering



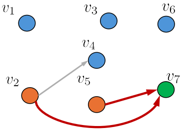
DAG from $\mathbf{A}$



DAG from $\mathbf{T}_1$



DAG from $\mathbf{T}_4$



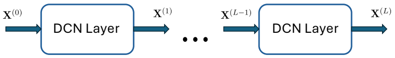
DAG from $\mathbf{T}_5$

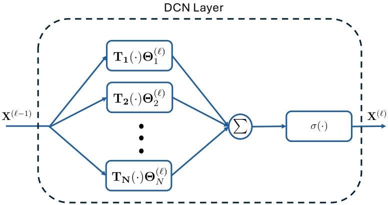
Filter Bank DCN

- Learning a single filter helps in developing intuition but lacks expressivity
  - $\Rightarrow$ Instead we can learn a filter bank at each layer

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

  - $\Rightarrow$ Learnable coefficients of the filter bank collected in $\boldsymbol{\Theta}_k^{(\ell)}$
  - $\Rightarrow$ The causal GSO still drive the convolution





Discussion

Interpretation

- **Spectral**: recall that $\mathbf{T}_k \mathbf{x}^{(\ell)} = \mathbf{W} \mathbf{D}_k \mathbf{c}^{(\ell)}$
  - $\Rightarrow$ Convolution selects and diffuses causes from predecessors across $\mathcal{D}$
- **Message passing**: filter coefficients determine how to mix messages
  - $\Rightarrow$ $\mathbf{T}_k$ forms a message from predecessors common to nodes k and i

Main advantages

- Is a permutation equivariant architecture
- Has a spectral representation thanks to GSOs $\mathbf{T}_k$
- $\mathbf{T}_k$ has binary eigenvalues avoiding numerical issues

Limitations

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

Numerical evaluation (I)

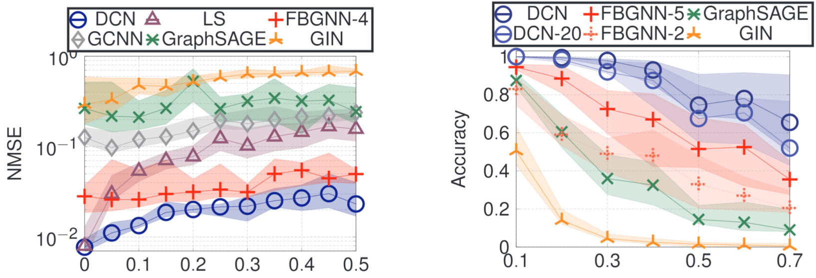
- We test the performance of DCN over synthetic data in two different tasks:
  - ⇒ Network diffusion: predict output of a diffusion process given input
  - ⇒ Source identification: identify source nodes given the output

	Network Diffusion		**Source Identification**	
	MNSE	Time (s)	Accuracy	Time (s)
DCN	$\mathbf{0.016 \pm 0.014}$	3.6	0.052 ± 0.014	7.5
DCN-30	$\mathbf{0.029 \pm 0.017}$	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	$\mathbf{0.991 \pm 0.018}$	8.2
DCN-30-T	0.199 ± 0.030	3.7	$\mathbf{0.983 \pm 0.032}$	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

- Classical architectures struggle to learn from data defined on DAGs
- DCN outperforms the alternatives even with approximate convolution
  - ⇒ Clear impact of the directionality on each task

Numerical evaluation (II)

- Analyze the sensitivity to noise (left) and DAG sparsity (right)
  - $\Rightarrow$ In network diffusion and and source identification tasks



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

- DCN consistently outperforms the alternatives
  - $\Rightarrow$ More resilient to the presence of noise and denser graphs

Conclusions and future works

Conclusions

- We introduced DCN, a DAG-aware convolutional GNN
  - $\Rightarrow$ Based on learning a bank of causal filters
- Simple architecture based on convolution for DAGs
  - $\Rightarrow$ Stronger inductive bias from DAG partial ordering
  - $\Rightarrow$ The architecture is permutation equivariant
  - $\Rightarrow$ Admits a sprectral interpretation
- Promising performance over synthetic data

Future research directions

- Strengthen the numerical evaluation of DCN
- Use the spectral representation to characterize the architecture
- Select GSOs in a intelligent way

Questions?

Questions at: **samuel.rey.escudero@urjc.es**