Untrained graph neural networks for denoising

S. Rey, S. Segarra, R. Heckel, A. G. Marques

ICASSP 2023
4 - 10 JUNE,
RHODES ISLAND, GREECE

48th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2023) - Rhodes Island, Greece - June 4-10, 2023

- Data is becoming **heterogeneous** and **pervasive** [Kolaczyk09][Leskovec20]
  - ⇒ Large amounts of data are propelling data-driven methods like NNs
- Complexity of contemporary systems and networks is increasing
  - ⇒ A popular alternative understand data as signals defined on a graph
  - ⇒ Harness graph topology to deal with irregular structure as in GNNs

Brain network

Social network

Home automation network

Why denoising graph signals?

Universidad Rey Juan Carlos

- Data is becoming **heterogeneous** and **pervasive** [Kolaczyk09][Leskovec20]
  - $\Rightarrow$ Large amounts of data are propelling data-driven methods like NNs
- Complexity of contemporary systems and networks is increasing
  - $\Rightarrow$ A popular alternative understand data as signals defined on a graph
  - $\Rightarrow$ Harness graph topology to deal with irregular structure as in GNNs
- **Problem:** data is corrupted with noise that may render data useless

Brain network

Social network

Home automation network

Why denoising graph signals?

- Data is becoming **heterogeneous** and **pervasive** [Kolaczyk09][Leskovec20]
  - ⇒ Large amounts of data are propelling data-driven methods like NNs
- Complexity of contemporary systems and networks is increasing
  - ⇒ A popular alternative understand data as signals defined on a graph
  - ⇒ Harness graph topology to deal with irregular structure as in GNNs
- **Problem:** data is corrupted with noise that may render data useless
- **This work:** design architectures to remove the noise from the data

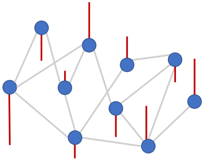
Brain network

Social network

Home automation network

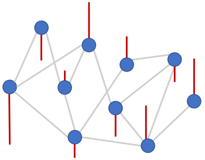
Graphs, signals, and GNNs

- Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes and adjacency $\mathbf{A}$
  - $\Rightarrow A_{ij}$ = Proximity between i and j
- Define a signal $\mathbf{x} \in \mathbb{R}^N$ on top of the graph
  - $\Rightarrow x_i$ = Signal value at node i

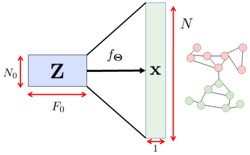


Universidad Rey Juan Carlos

- Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes and adjacency $\mathbf{A}$
  - $\Rightarrow A_{ij} =$ Proximity between i and j
- Define a signal $\mathbf{x} \in \mathbb{R}^N$ on top of the graph
  - $\Rightarrow x_i =$ Signal value at node i



- Represent GNN as parametric function $f_{\mathbf{\Theta}}(\mathbf{Z}|\mathcal{G}) : \mathbb{R}^{N^{(0)} \times F^{(0)}} \to \mathbb{R}^N$



- With $\mathbf{Y}^{(0)} = \mathbf{Z}$, a GNN with L layers given by

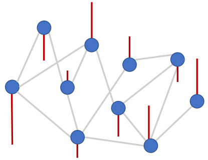
$$\hat{\mathbf{Y}}^{(\ell)} = \mathcal{T}^{(\ell)}_{\mathbf{\Theta}^{(\ell)}} \left\{ \mathbf{Y}^{(\ell-1)} | \mathcal{G} \right\}, \;\; 1 \leq \ell \leq L,$$

$$Y_{ij}^{(\ell)} = g^{(\ell)} \left(\hat{Y}_{ij}^{(\ell)} \right), \;\; 1 \leq \ell \leq L,$$

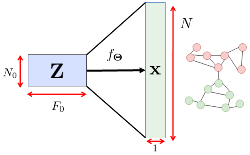
$\Rightarrow$ Graph-aware linear operator and non-linearity

Graphs, signals, and GNNs

- Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes and adjacency $\mathbf{A}$
  - $\Rightarrow A_{ij}$ = Proximity between i and j
- Define a signal $\mathbf{x} \in \mathbb{R}^N$ on top of the graph
  - $\Rightarrow x_i$ = Signal value at node i



- Represent GNN as parametric function $f_{\mathbf{\Theta}}(\mathbf{Z}|\mathcal{G}) : \mathbb{R}^{N^{(0)} \times F^{(0)}} \to \mathbb{R}^N$



- With $\mathbf{Y}^{(0)} = \mathbf{Z}$, a GNN with L layers given by

$$\hat{\mathbf{Y}}^{(\ell)} = \mathcal{T}^{(\ell)}_{\mathbf{\Theta}^{(\ell)}} \left\{ \mathbf{Y}^{(\ell-1)} | \mathcal{G} \right\}, \quad 1 \leq \ell \leq L,$$

$$Y^{(\ell)}_{ij} = g^{(\ell)} \left(\hat{Y}^{(\ell)}_{ij} \right), \quad 1 \leq \ell \leq L,$$

$\Rightarrow$ Graph-aware linear operator and non-linearity

- Focus on designing the **linear operator** $\mathcal{T}^{(\ell)}_{\mathbf{\Theta}^{(\ell)}}$ to exploit information from $\mathcal{G}$

- Graph signal denoising seeks to remove the noise from the signal
  - $\Rightarrow$ Recover **unknown** signal $\mathbf{x}_0 \in \mathbb{R}^N$ from noisy observation $\mathbf{x} = \mathbf{x}_0 + \mathbf{n}$
- Traditional approaches based on regularized LS **[Chen14][Wang15]**
  - $\Rightarrow$ Graph-related regularization to promote desired properties

Problem description

- Graph signal denoising seeks to remove the noise from the signal
  - $\Rightarrow$ Recover **unknown** signal $\mathbf{x}_0 \in \mathbb{R}^N$ from noisy observation $\mathbf{x} = \mathbf{x}_0 + \mathbf{n}$
- Traditional approaches based on regularized LS [Chen14][Wang15]
  - $\Rightarrow$ Graph-related regularization to promote desired properties
- **Our goal**: design and analyze untrained GNNs to denoise graph signals

$$\hat{\boldsymbol{\Theta}} = \mathrm{argmin}_{\boldsymbol{\Theta}} \frac{1}{2} \|\mathbf{x} - f_{\boldsymbol{\Theta}}(\mathbf{Z}|\mathcal{G})\|_2^2$$

  - $\Rightarrow$ Each $\hat{\mathbf{x}}_0 = f_{\hat{\boldsymbol{\Theta}}}(\mathbf{Z}|\mathcal{G})$ estimated individually from a single observation
  - $\Rightarrow$ Weights $\hat{\boldsymbol{\Theta}}$ fitted for each $\mathbf{x}$ without training phase

Problem description

Universidad Rey Juan Carlos

- Graph signal denoising seeks to remove the noise from the signal
  - $\Rightarrow$ Recover **unknown** signal $\mathbf{x}_0 \in \mathbb{R}^N$ from noisy observation $\mathbf{x} = \mathbf{x}_0 + \mathbf{n}$
- Traditional approaches based on regularized LS [Chen14][Wang15]
  - $\Rightarrow$ Graph-related regularization to promote desired properties
- **Our goal**: design and analyze untrained GNNs to denoise graph signals

$$\hat{\mathbf{\Theta}} = \mathrm{argmin}_{\mathbf{\Theta}} \frac{1}{2} \|\mathbf{x} - f_{\mathbf{\Theta}}(\mathbf{Z}|\mathcal{G})\|_2^2$$

  - $\Rightarrow$ Each $\hat{\mathbf{x}}_0 = f_{\hat{\mathbf{\Theta}}}(\mathbf{Z}|\mathcal{G})$ estimated individually from a single observation
  - $\Rightarrow$ Weights $\hat{\mathbf{\Theta}}$ fitted for each $\mathbf{x}$ without training phase
- **Key assumption:** GNN designed to **learn the signal faster** than noise
  - $\Rightarrow$ The GNN incorporates an implicit regularization
  - $\Rightarrow$ Apply SGD in combination with **early stopping**

Graph Convolutional Generator

- The GCG includes the graph topology via vertex-based convolution
  - $\Rightarrow$ Graph convolution via **fixed GF** $\mathbf{H} = \sum_{r=0}^{R-1} h_r \mathbf{A}^r \in \mathbb{R}^{N \times N}$
- The output of the GCG with L layers is

$$\mathbf{Y}^{(\ell)} = \text{ReLU}(\mathbf{H}\mathbf{Y}^{(\ell-1)}\boldsymbol{\Theta}^{(\ell)}), \text{ for } \ell = 1, ..., L-1$$
$$\mathbf{y}^{(L)} = \mathbf{H}\mathbf{Y}^{(L-1)}\boldsymbol{\Theta}^{(L)}$$

  - $\Rightarrow$ Fixed $\mathbf{H}$ captures prior knowledge of $\mathbf{x}_0$
  - $\Rightarrow$ Learnable parameters $\boldsymbol{\Theta}^{(\ell)} \in \mathbb{R}^{F^{(\ell-1)} \times F^{(\ell)}}$ mix columns

Graph Convolutional Generator

- The GCG includes the graph topology via vertex-based convolution
  - $\Rightarrow$ Graph convolution via **fixed GF** $\mathbf{H} = \sum_{r=0}^{R-1} h_r \mathbf{A}^r \in \mathbb{R}^{N \times N}$
- The output of the GCG with L layers is

$$\mathbf{Y}^{(\ell)} = \text{ReLU}(\mathbf{H}\mathbf{Y}^{(\ell-1)}\mathbf{\Theta}^{(\ell)}), \quad \text{for} \quad \ell = 1, ..., L-1$$
$$\mathbf{y}^{(L)} = \mathbf{H}\mathbf{Y}^{(L-1)}\mathbf{\Theta}^{(L)}$$

  - $\Rightarrow$ Fixed $\mathbf{H}$ captures prior knowledge of $\mathbf{x}_0$
  - $\Rightarrow$ Learnable parameters $\mathbf{\Theta}^{(\ell)} \in \mathbb{R}^{F^{(\ell-1)} \times F^{(\ell)}}$ mix columns
- **Features of the architecture**
  - $\Rightarrow$ The depth of GCG and the radius of $\mathbf{H}$ are independent
  - $\Rightarrow$ Avoids over-smoothing problem
  - $\Rightarrow$ Generalization of the GCNN layer [Kipf16]

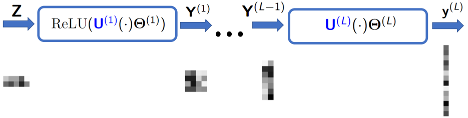
- GDec includes the graph topology via graph upsampling

  $\Rightarrow$ Design of graph upsampling operator $\mathbf{U}^{(\ell)} \in \mathbb{R}^{N^{(\ell)} \times N^{(\ell-1)}}$

- The output of the GDec with L layers is

$$\mathbf{Y}^{(\ell)} = \mathrm{ReLU}(\mathbf{U}^{(\ell)} \mathbf{Y}^{(\ell-1)} \mathbf{\Theta}^{(\ell)}), \quad \text{for } \ell = 1, ..., L-1$$
$$\mathbf{y}^{(L)} = \mathbf{U}^{(L)} \mathbf{Y}^{(L-1)} \mathbf{\Theta}^{(L)}$$

  $\Rightarrow$ $\mathbf{U}^{(\ell)}$ increases size of intermediate signals $\mathbf{Y}^{(\ell-1)}$ since $N^{(0)} < N$



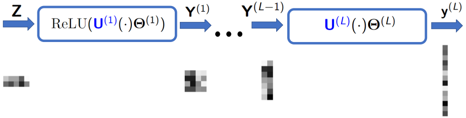
Graph Decoder

Universidad Rey Juan Carlos

- GDec includes the graph topology via graph upsampling
  - $\Rightarrow$ Design of graph upsampling operator $\mathbf{U}^{(\ell)} \in \mathbb{R}^{N^{(\ell)} \times N^{(\ell-1)}}$
- The output of the GDec with L layers is

$$\mathbf{Y}^{(\ell)} = \mathrm{ReLU}(\mathbf{U}^{(\ell)}\mathbf{Y}^{(\ell-1)}\boldsymbol{\Theta}^{(\ell)}), \quad \text{for } \ell = 1, ..., L-1$$
$$\mathbf{y}^{(L)} = \mathbf{U}^{(L)}\mathbf{Y}^{(L-1)}\boldsymbol{\Theta}^{(L)}$$

  - $\Rightarrow$ $\mathbf{U}^{(\ell)}$ increases size of intermediate signals $\mathbf{Y}^{(\ell-1)}$ since $N^{(0)} < N$



- **Features of the architecture**
  - $\Rightarrow$ Graph topology considered via **clustering-based** design of $\mathbf{U}^{(\ell)}$
  - $\Rightarrow$ Reduced dimensionality of $\mathbf{Z}$ implicitly limits the **degrees of freedom**
  - $\Rightarrow$ More robust to noise but more sensitive to model mismatch

Theoretical analysis

- Considering 2-layer implementations and bandlimited graph signals $\mathbf{x}_0$
- We prove that
  - $\Rightarrow$ $\mathbf{x}_0$ learned faster than noise so error decreases for the first iters
  - $\Rightarrow$ With too many iters, noise is also learned and error increases
- We can use early stopping to denoise signals with GCG or GDec!

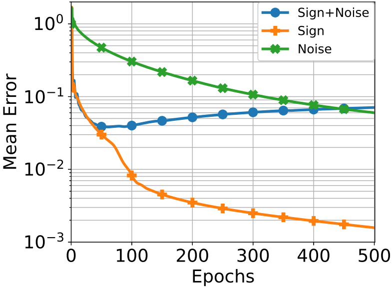
Universidad Rey Juan Carlos

Theoretical analysis

- Considering 2-layer implementations and bandlimited graph signals $\mathbf{x}_0$
- We prove that
  - ⇒ $\mathbf{x}_0$ learned faster than noise so error decreases for the first iters
  - ⇒ With too many iters, noise is also learned and error increases
- We can use early stopping to denoise signals with GCG or GDec!

Numerical validation

- Theoretical analysis validated through simulations
- Tested in real-world datasets
  - ⇒ Weather stations data
  - ⇒ S&P 500
  - ⇒ Cora



Conclusions

- We approached the problem of graph signal denoising
  - $\Rightarrow$ Designed untrained GNNs that learn signal faster than noise
- Introduced 2 GNNs that exploit the graph with different methods
  - $\Rightarrow$ GCG employs vertex-based convolutions
  - $\Rightarrow$ GDec employs graph upsampling operators
- Performance of both architectures analyzed theoretically and numerically
  - $\Rightarrow$ Introduced a bound for the error of the denoised signal
  - $\Rightarrow$ Assessed the performance in synthetic and real-world data

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