Universidad Rey Juan Carlos

Designing graph filters and graph neural networks in the presence of graph perturbations

Antonio G. Marques
King Juan Carlos University - Madrid (Spain)
`http://tsc.urjc.es/~amarques`

In collaboration with Samuel Rey, Victor Tenorio

Grants: PID2019-105032GB-I00, TED2021-130347B-I00, PID2022-136887NB-I00

McGill | Bellairs Research Institute

Coates Workshop - Barbados - Jan. 24, 2024

- Data is becoming heterogeneous and pervasive [Kolaczyk09][Leskovec20]
  - ⇒ Huge amounts of data are generated and stored
  - ⇒ Complexity of contemporary systems and networks is increasing
- Modeling the structure of the data as a graphs is an effective approach
  - ⇒ **GSP**: harness graph topology to process the data [Shuman13][Ortega18]

Social network

Brain network

Home automation network

- Data is becoming heterogeneous and pervasive [Kolaczyk09][Leskovec20]
  - ⇒ Huge amounts of data are generated and stored
  - ⇒ Complexity of contemporary systems and networks is increasing
- Modeling the structure of the data as a graphs is an effective approach
  - ⇒ **GSP**: harness graph topology to process the data [Shuman13][Ortega18]
- **Problem:** data is prone to errors and imperfections
  - ⇒ Noise, missing values, or outliers are ubiquitous in data science

Social network

Brain network

Home automation network

True signal

Noisy signal

Perturbations in the observed signals

- At the heart of SP, fairly studied in GSP
- GSP main focus: influence of the graph topology
  - $\Rightarrow$ Graph-dependent noise in signals
  - $\Rightarrow$ Node-dependent missing values

Data imperfections in GSP

True signal

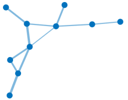
Noisy signal

Perturbations in the observed signals

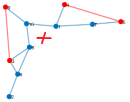
- At the heart of SP, fairly studied in GSP
- GSP main focus: influence of the graph topology
  - ⇒ Graph-dependent noise in signals
  - ⇒ Node-dependent missing values

Perturbations in the graph topology

- Critical for most GSP tools and methods
- Inherent to graph learning approach
- Even small perturbations lead to challenging problems
- Barely studied in the GSP literature!
  - ⇒ Uncertainty in the edges [Miettinen19],[Ceci20]
  - ⇒ Presence of hidden nodes



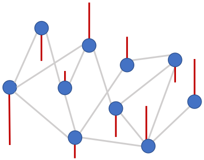
True graph



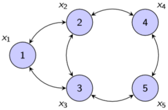
Noisy graph

Fundamentals of GSP

- 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



- Associated with $\mathcal{G}$ is the graph-shift operator $\mathbf{S} \in \mathbb{R}^{N \times N}$ (e.g. $\mathbf{A}$, $\mathbf{L}$)
  - $\Rightarrow S_{ij} \neq 0$ if $i = j$ or $(i,j) \in \mathcal{E}$ (local structure in $\mathcal{G}$) [Shuman12][Sandryhaila13]



$$\begin{pmatrix} \cdot & \cdot & \cdot & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & 0 & \cdot \\ 0 & \cdot & 0 & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot \end{pmatrix}$$

- GSP: Exploit structure encoded in $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ to process $\mathbf{x}$
  - $\Rightarrow$ Key to that end: a) eigenvecs. of $\mathbf{S}$ and b) polynomials on $\mathbf{S}$

Graph filters and GNNs

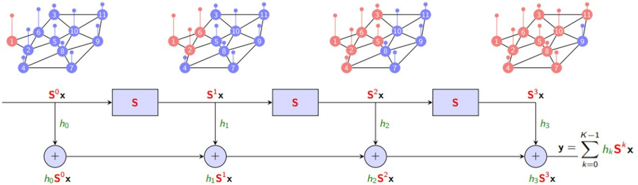
- Focus today: learn filter coefficients of GFs and GNNs when errors in $\mathbf{S}$
  - $\Rightarrow$ Let us spend more time with these two convolutional architectures
- Graph filter: mapping between graph signals written as polynomial on $\mathbf{S}$

$$\mathbf{y} \;=\; \mathbf{H}\mathbf{x} \;=\; \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x} \;=\; h_0 \mathbf{S}^0 \mathbf{x} + h_1 \mathbf{S}^1 \mathbf{x} + h_2 \mathbf{S}^2 \mathbf{x} + \ldots + h_{K-1} \mathbf{S}^{K-1} \mathbf{x}$$

$\Rightarrow$ $\mathbf{S}\mathbf{x}$ local operation (# hops) $\Rightarrow$ local and efficient computation

$\Rightarrow$ Well understood in the spectral domain $\Rightarrow$ $\mathbf{H}$ and $\mathbf{S}$ same eigenvecs.

$\Rightarrow$ Reduces to time invariant filter if $[\mathbf{S}\mathbf{x}]_n = [\mathbf{x}]_{n+1}$



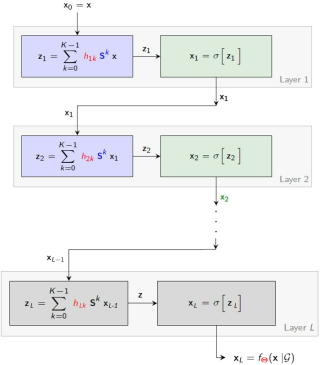
Universidad Rey Juan Carlos

- NNs stack layers composing pointwise nonlinearities with linear transforms

$$\mathbf{x}_1 = \sigma_1\Big(\mathbf{W}_1\mathbf{x}_0\Big), \; \ldots, \; \mathbf{x}_\ell = \sigma_\ell\Big(\mathbf{W}_\ell\mathbf{x}_{\ell-1}\Big), \; \ldots, \; \mathbf{x}_L = \sigma_L\Big(\mathbf{W}_L\mathbf{x}_{L-1}\Big)$$

$\Rightarrow$ NN is $\mathbf{y} = f_{\boldsymbol{\Theta}}(\mathbf{x})$ with $\mathbf{y} = \mathbf{x}_L$, $\mathbf{x}_0 = \mathbf{x}$, $\boldsymbol{\Theta} = \{\mathbf{W}_\ell\}$ overparam

- GNNs incorporate $\mathcal{G}$ ($\mathbf{S}$) into the NN $\Rightarrow \mathbf{y} = f_{\boldsymbol{\Theta}}(\mathbf{x}\,|\mathcal{G})$



- Graph-aware linear operators
- Parsimonious parametrization via GF
- Reduce to CNN if time convolution adopted
- Can be modified to deal with multi-feature

Fitting GFs and GNN to data

- Given training set $\mathcal{T} = \{(\mathbf{x}_m, \mathbf{y}_m)\}_{m=1}^{M}$ with input-output pairs over $\mathcal{G}$
  $\Rightarrow \mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]$, $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$
- GOAL: Use $\mathcal{T}$ to learn graph-aware mapping from $\mathcal{X}$ to $\mathcal{Y}$
  - Key: postulate a mapping meaningful and easy to learn $\Rightarrow$ GFs and GNNs
  - Useful for: (1) Estimating output $\hat{\mathbf{y}}$ associated with input $\mathbf{x} \notin \mathcal{T}$ and (2) Identifying some network dynamics represented by filter coefficients

Fitting GFs and GNN to data

- Given training set $\mathcal{T} = \{(\mathbf{x}_m, \mathbf{y}_m)\}_{m=1}^{M}$ with input-output pairs over $\mathcal{G}$

  $\Rightarrow \mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]$, $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$
- GOAL: Use $\mathcal{T}$ to learn graph-aware mapping from $\mathcal{X}$ to $\mathcal{Y}$
  - Key: postulate a mapping meaningful and easy to learn $\Rightarrow$ GFs and GNNs
  - Useful for: (1) Estimating output $\hat{\mathbf{y}}$ associated with input $\mathbf{x} \notin \mathcal{T}$ and (2) Identifying some network dynamics represented by filter coefficients
- If $\mathbf{S}$ is perfectly known, optimal GF fitting

$$\min_{\mathbf{H}} \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2 \qquad \min_{\mathbf{h}} \|\mathbf{Y} - \sum_{k=0}^{N-1} h_k \mathbf{S}^k \mathbf{X}\|_F^2 \qquad \min_{\tilde{\mathbf{h}}} \|\mathbf{Y} - \mathbf{V}\mathrm{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top \mathbf{X}\|_F^2$$

Fitting GFs and GNN to data

- Given training set $\mathcal{T} = \{(\mathbf{x}_m, \mathbf{y}_m)\}_{m=1}^{M}$ with input-output pairs over $\mathcal{G}$
  $\Rightarrow \mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]$, $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$
- GOAL: Use $\mathcal{T}$ to learn graph-aware mapping from $\mathcal{X}$ to $\mathcal{Y}$
  - Key: postulate a mapping meaningful and easy to learn $\Rightarrow$ GFs and GNNs
  - Useful for: (1) Estimating output $\hat{\mathbf{y}}$ associated with input $\mathbf{x} \notin \mathcal{T}$ and (2) Identifying some network dynamics represented by filter coefficients
- If $\mathbf{S}$ is perfectly known, optimal GF fitting

$$\min_{\mathbf{H}} \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2 \qquad \min_{\mathbf{h}} \|\mathbf{Y} - \sum_{k=0}^{N-1} h_k \mathbf{S}^k \mathbf{X}\|_F^2 \qquad \min_{\tilde{\mathbf{h}}} \|\mathbf{Y} - \mathbf{V}\mathrm{diag}(\tilde{\mathbf{h}})\mathbf{V}^\top \mathbf{X}\|_F^2$$

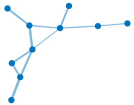
- If $\mathbf{S}$ is perfectly known, optimal GNN fitting

$$\min_{\boldsymbol{\Theta}} \sum_{m=1}^{M} \|\mathbf{y}_m - f_{\boldsymbol{\Theta}}(\mathbf{x}_m \,|\mathbf{S})\|_2^2 \quad \text{with} \quad \boldsymbol{\Theta} = \{\mathbf{h}_\ell\}_{\ell=1}^{L}$$

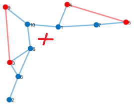
$\Rightarrow$ SGD (via backpropagation) over $\{\mathbf{h}_\ell\}_{\ell=1}^{L}$ $\Rightarrow \mathbf{h}_\ell^{(t+1)} = \mathbf{h}_\ell^{(t)} + \mu ...$

Perturbed topology in graph filter ID

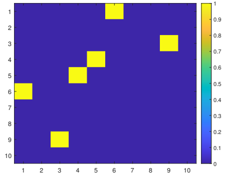
- When fitting GFs and GNN to data $\Rightarrow$ Key that linear operators are polynomials of $\mathbf{S}$
- Assume access only to perturbed $\bar{\mathbf{S}} \in \mathbb{R}^{N \times N} \Rightarrow \bar{\mathbf{S}} \neq \mathbf{S}$
  - $\Rightarrow$ The true $\mathbf{S}$ is unknown
- What if we estimate the filter as $\mathbf{H} = \sum_{r=0}^{R-1} h_r \bar{\mathbf{S}}^r$?
  - $\Rightarrow$ Error between $\mathbf{S}^r$ and $\bar{\mathbf{S}}^r$ grows with r



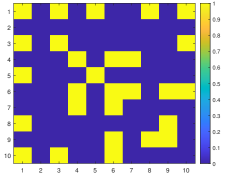
True $\mathcal{G}$



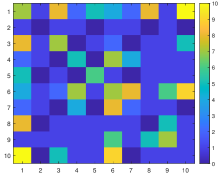
Observed $\mathcal{G}$



$|\mathbf{A} - \bar{\mathbf{A}}|$



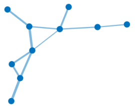
$|\mathbf{A}^2 - \bar{\mathbf{A}}^2|$

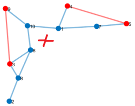


$|\mathbf{A}^4 - \bar{\mathbf{A}}^4|$

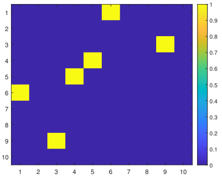
Universidad Rey Juan Carlos

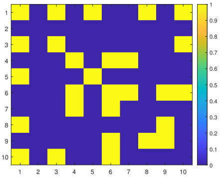
- When fitting GFs and GNN to data $\Rightarrow$ Key that linear operators are polynomials of $\mathbf{S}$
- Assume access only to perturbed $\bar{\mathbf{S}} \in \mathbb{R}^{N \times N} \Rightarrow \bar{\mathbf{S}} \neq \mathbf{S}$
  - $\Rightarrow$ The true $\mathbf{S}$ is unknown
- What if we estimate the filter as $\mathbf{H} = \sum_{r=0}^{R-1} h_r \bar{\mathbf{S}}^r$?
  - $\Rightarrow$ Error between $\mathbf{S}^r$ and $\bar{\mathbf{S}}^r$ grows with r

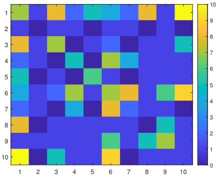

True $\mathcal{G}$


Observed $\mathcal{G}$

- **Challenge:** learning $\mathbf{H}$ as polynomial of $\bar{\mathbf{S}}$ entails **high estimation error**


$|\mathbf{A} - \bar{\mathbf{A}}|$


$|\mathbf{A}^2 - \bar{\mathbf{A}}^2|$


$|\mathbf{A}^4 - \bar{\mathbf{A}}^4|$

Universidad Rey Juan Carlos

Modeling graph perturbations

- Additive perturbation models are pervasive in SP $\Rightarrow$ In graphs $\bar{\mathbf{S}} = \mathbf{S} + \boldsymbol{\Delta}$
  - $\Rightarrow$ Structure of $\boldsymbol{\Delta} \in \mathbb{R}^{N \times N}$ depends on the type of perturbation
  - $\Rightarrow$ $\mathbf{S}$ and $\bar{\mathbf{S}}$ are close according to some metric $d(\mathbf{S}, \bar{\mathbf{S}})$

Modeling graph perturbations

- Additive perturbation models are pervasive in SP $\Rightarrow$ In graphs $\bar{\mathbf{S}} = \mathbf{S} + \boldsymbol{\Delta}$
  - $\Rightarrow$ Structure of $\boldsymbol{\Delta} \in \mathbb{R}^{N \times N}$ depends on the type of perturbation
  - $\Rightarrow$ $\mathbf{S}$ and $\bar{\mathbf{S}}$ are close according to some metric $d(\mathbf{S}, \bar{\mathbf{S}})$

Examples of topology perturbations

- When perturbations **create/destroy edges** $\implies$ $d(\mathbf{S}, \bar{\mathbf{S}}) = \|\mathbf{S} - \bar{\mathbf{S}}\|_0$
  - $\Rightarrow$ $\Delta_{ij} = 1$ if $S_{ij} = 0$ and $\Delta_{ij} = -1$ if $S_{ij} = 1$
- When perturbations represent **noisy edges** $\implies$ $d(\mathbf{S}, \bar{\mathbf{S}}) = \|\mathbf{S}_{\mathcal{E}} - \bar{\mathbf{S}}_{\mathcal{E}}\|_2^2$
  - $\Rightarrow$ $\Delta_{ij} = 0$ if $S_{ij} = 0$ and $\Delta_{ij} \sim \mathcal{N}(0, \sigma^2)$ if $S_{ij} \neq 0$

Challenges of additive graph perturbation models

- Analyzing / translating the effect on either $\mathbf{S}^r$ or $\mathbf{V}$ very difficult [Ceci20]
- Worst case bounds, AR/FIR filters of degree one, ER perturbations... [Miettinen19]

Fitting GFs and GNNs from perturbed GSO

- Given $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]$, $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$ and perturbed $\bar{\mathbf{S}}$ $\Rightarrow$ Find GF/GNN to:
  - $\Rightarrow$ (1) Estimate output $\hat{\mathbf{y}}$ associated $\mathbf{x} \notin \mathcal{T}$
  - $\Rightarrow$ (2) Identify true network dynamics represented by filter coefficients

Fitting GFs and GNNs from perturbed GSO

- Given $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]$, $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$ and perturbed $\bar{\mathbf{S}}$ $\Rightarrow$ Find GF/GNN to:
  - $\Rightarrow$ (1) Estimate output $\hat{\mathbf{y}}$ associated $\mathbf{x} \notin \mathcal{T}$
  - $\Rightarrow$ (2) Identify true network dynamics represented by filter coefficients
- Key in our approach: postulate true $\mathbf{S}$ as an optimization variable
  - $\Rightarrow$ OK: Enhanced (denoised) estimate of GSO is obtained
  - $\Rightarrow$ OK: Additive model can be leveraged / We work on vertex domain
  - $\Rightarrow$ KO: Optimization non-convex

Fitting GFs and GNNs from perturbed GSO

- Given $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]$, $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$ and perturbed $\bar{\mathbf{S}}$ $\Rightarrow$ Find GF/GNN to:
  - $\Rightarrow$ (1) Estimate output $\hat{\mathbf{y}}$ associated $\mathbf{x} \notin \mathcal{T}$
  - $\Rightarrow$ (2) Identify true network dynamics represented by filter coefficients
- Key in our approach: postulate true $\mathbf{S}$ as an optimization variable
  - $\Rightarrow$ OK: Enhanced (denoised) estimate of GSO is obtained
  - $\Rightarrow$ OK: Additive model can be leveraged / We work on vertex domain
  - $\Rightarrow$ KO: Optimization non-convex

Outline of the talk

- Formulation for a single GF
  - $\Rightarrow$ Relaxations and algorithmic alternatives
- Formulation for multiple GFs
- Formulation for GNNs
- Generalizations to adversarial setups and future work

- Since dealing with $\mathbf{V}$ is challenging, a straightforward vertex-based approach is

$$\min_{\mathbf{h},\mathbf{S}\in\mathcal{S}} \|\mathbf{Y}-\sum_{k=0}^{N-1} h_k\mathbf{S}^k\mathbf{X}\|_F^2+\lambda d(\mathbf{S},\bar{\mathbf{S}})+\beta\|\mathbf{S}\|_0$$

$\Rightarrow$ OK: Second term promotes closeness between $\bar{\mathbf{S}}$ and $\mathbf{S}$

$\Rightarrow$ KO: High order polynomials: highly non-convex and numerically unstable

Proposed RFI formulation

- Since dealing with $\mathbf{V}$ is challenging, a straightforward vertex-based approach is

$$\min_{\mathbf{h},\mathbf{S}\in\mathcal{S}} \|\mathbf{Y} - \sum_{k=0}^{N-1} h_k \mathbf{S}^k \mathbf{X}\|_F^2 + \lambda d(\mathbf{S}, \bar{\mathbf{S}}) + \beta\|\mathbf{S}\|_0$$

  - $\Rightarrow$ OK: Second term promotes closeness between $\bar{\mathbf{S}}$ and $\mathbf{S}$
  - $\Rightarrow$ KO: High order polynomials: highly non-convex and numerically unstable

Proposed RFI formulation

- Define full $\mathbf{H}$ as optimization variable
- Leverage that if GF is a polynomial of GSO, then $\mathbf{H}$ and $\mathbf{S}$ commute

$$\min_{\mathbf{S}\in\mathcal{S},\mathbf{H}} \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2 + \lambda d(\mathbf{S}, \bar{\mathbf{S}}) + \beta\|\mathbf{S}\|_0 \quad \text{s. t.} \;\; \mathbf{S}\mathbf{H} = \mathbf{H}\mathbf{S}$$

  - $\Rightarrow$ Constraint: $\mathbf{H}$ is a polynomial of $\mathbf{S}$.
  - $\Rightarrow$ Regularizers: sparsity and closeness between $\bar{\mathbf{S}}$ and $\mathbf{S}$

- Operates fully in vertex domain, avoids computation of high-order polynomials
- Bilinear terms and ℓ_0 render the problem non-convex

Universidad Rey Juan Carlos

Dealing with ℓ_0 norm

- We employ the ℓ_1 reweighted norm based on logarithmic penalty [Candes08]

$$\|\mathbf{Z}\|_0 \approx r_\delta(\mathbf{Z}) := \sum_{i=1}^{I}\sum_{j=1}^{J}\log(|Z_{ij}| + \delta)$$

$\Rightarrow$ Produces sparser solutions than ℓ_1 norm

$\Rightarrow$ Majorization-Minimization approach based on linear approximation

Universidad Rey Juan Carlos

Dealing with ℓ_0 norm

- We employ the ℓ_1 reweighted norm based on logarithmic penalty [Candes08]

$$\|\mathbf{Z}\|_0 \approx r_\delta(\mathbf{Z}) := \sum_{i=1}^{I}\sum_{j=1}^{J}\log(|Z_{ij}| + \delta)$$

  $\Rightarrow$ Produces sparser solutions than ℓ_1 norm

  $\Rightarrow$ Majorization-Minimization approach based on linear approximation

Dealing with bilinear term

- Adopt an alternating-minimization approach to break the non-linearity

  $\Rightarrow$ $\mathbf{H}$ and $\mathbf{S}$ are estimated in two separate iterative steps

  $\Rightarrow$ Each step requires solving a convex optimization problem

Universidad Rey Juan Carlos

Dealing with ℓ_0 norm

- We employ the ℓ_1 reweighted norm based on logarithmic penalty [Candes08]

$$\|\mathbf{Z}\|_0 \approx r_\delta(\mathbf{Z}) := \sum_{i=1}^{I} \sum_{j=1}^{J} \log(|Z_{ij}| + \delta)$$

$\Rightarrow$ Produces sparser solutions than ℓ_1 norm

$\Rightarrow$ Majorization-Minimization approach based on linear approximation

Dealing with bilinear term

- Adopt an alternating-minimization approach to break the non-linearity

$\Rightarrow$ **H** and **S** are estimated in two separate iterative steps

$\Rightarrow$ Each step requires solving a convex optimization problem

- Rewrite optimization problem as

$$\min_{\mathbf{S}\in\mathcal{S},\mathbf{H}} \|\mathbf{Y}-\mathbf{H}\mathbf{X}\|_F^2 + \lambda r_{\delta_1}(\mathbf{S}-\bar{\mathbf{S}}) + \beta r_{\delta_2}(\mathbf{S}) + \gamma\|\mathbf{S}\mathbf{H}-\mathbf{H}\mathbf{S}\|_F^2$$

$\Rightarrow$ Constraint $\mathbf{SH} = \mathbf{HS}$ relaxed as a regularizer

Alternating optimization algorithm

- **Step 1 - GF Identification**: estimate $\mathbf{H}^{(t+1)}$ with $\mathbf{S}^{(t)}$ fixed

$$\mathbf{H}^{(t+1)} = \arg\min_{\mathbf{H}} \|\mathbf{Y}-\mathbf{H}\mathbf{X}\|_F^2 + \gamma\|\mathbf{S}^{(t)}\mathbf{H}-\mathbf{H}\mathbf{S}^{(t)}\|_F^2$$

$\Rightarrow$ LS problem with closed-form solution inverting an $N^2 \times N^2$ matrix

- **Step 2 - Graph Denoising**: estimate $\mathbf{S}^{(t+1)}$ with $\mathbf{H}^{(t+1)}$ fixed

$$\mathbf{S}^{(t+1)} = \arg\min_{\mathbf{S}\in\mathcal{S}} \sum_{i,j=1}^{N} \left(\lambda \bar{\Omega}_{ij}^{(t)}|S_{ij}-\bar{S}_{ij}| + \beta\Omega_{ij}^{(t)}|S_{ij}|\right) + \gamma\|\mathbf{S}\mathbf{H}^{(t+1)}-\mathbf{H}^{(t+1)}\mathbf{S}\|_F^2$$

$\Rightarrow$ With ℓ_1 weights $\Omega_{ij}^{(t)}, \bar{\Omega}_{ij}^{(t)}$ computed from previous GSO $\mathbf{S}^{(t)}$

- Steps 1 and 2 repeated for $t = 0, ..., t_{max}-1$ iterations

Alternating optimization algorithm

Universidad Rey Juan Carlos

- **Step 1 - GF Identification**: estimate $\mathbf{H}^{(t+1)}$ with $\mathbf{S}^{(t)}$ fixed

$$\mathbf{H}^{(t+1)} = \arg\min_{\mathbf{H}} \|\mathbf{Y} - \mathbf{H}\mathbf{X}\|_F^2 + \gamma \|\mathbf{S}^{(t)}\mathbf{H} - \mathbf{H}\mathbf{S}^{(t)}\|_F^2$$

$\Rightarrow$ LS problem with closed-form solution inverting an $N^2 \times N^2$ matrix

- **Step 2 - Graph Denoising**: estimate $\mathbf{S}^{(t+1)}$ with $\mathbf{H}^{(t+1)}$ fixed

$$\mathbf{S}^{(t+1)} = \arg\min_{\mathbf{S} \in \mathcal{S}} \sum_{i,j=1}^{N} \left(\lambda \bar{\Omega}_{ij}^{(t)} |S_{ij} - \bar{S}_{ij}| + \beta \Omega_{ij}^{(t)} |S_{ij}| \right) + \gamma \|\mathbf{S}\mathbf{H}^{(t+1)} - \mathbf{H}^{(t+1)}\mathbf{S}\|_F^2$$

$\Rightarrow$ With ℓ_1 weights $\Omega_{ij}^{(t)}, \bar{\Omega}_{ij}^{(t)}$ computed from previous GSO $\mathbf{S}^{(t)}$

- Steps 1 and 2 repeated for $t = 0, \ldots, t_{max} - 1$ iterations

Theorem

The RFI algorithm **converges to an stationary** point if $\mathbf{S}$ does not have repeated eigenvalues and every row of $\tilde{\mathbf{X}} = \mathbf{V}^{-1}\mathbf{X}$ is nonzero

Algorithmic enhancements

- **Additional constraints**: If data is graph-stationary
  $\Rightarrow \|\mathbf{C_X S} - \mathbf{S C_X}\| \leq \epsilon_{\mathbf{X}}$ and $\|\mathbf{C_Y S} - \mathbf{S C_Y}\| \leq \epsilon_{\mathbf{Y}}$

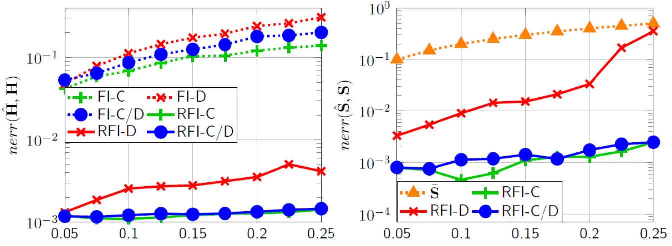
Algorithmic enhancements

- **Additional constraints**: If data is graph-stationary
  $\Rightarrow \|\mathbf{C_X S} - \mathbf{S C_X}\| \leq \epsilon_{\mathbf{X}}$ and $\|\mathbf{C_Y S} - \mathbf{S C_Y}\| \leq \epsilon_{\mathbf{Y}}$
- **Efficient implementation**: Computational complexity RFI alg. $\mathcal{O}(N^7)$
  $\Rightarrow$ Prohibitive for large graphs $\Rightarrow$ Steps 1 and 2 via an iterative process

Algorithmic enhancements

- **Additional constraints**: If data is graph-stationary
  $\Rightarrow \|\mathbf{C_X S} - \mathbf{S C_X}\| \leq \epsilon_{\mathbf{X}}$ and $\|\mathbf{C_Y S} - \mathbf{S C_Y}\| \leq \epsilon_{\mathbf{Y}}$
- **Efficient implementation**: Computational complexity RFI alg. $\mathcal{O}(N^7)$
  $\Rightarrow$ Prohibitive for large graphs $\Rightarrow$ Steps 1 and 2 via an iterative process
  - **Step 1 - Efficient GF Identification**
    $\Rightarrow$ Estimate $\mathbf{H}^{(t+1)}$ performing τ_{max_1} iterations of gradient descent
    $\Rightarrow$ Involves multiplications of $N \times N$ matrices
  - **Step 2 - Efficient Graph Denoising**
    $\Rightarrow$ Estimate $\mathbf{S}^{(t+1)}$ via alternating optimization for τ_{max_2}
    $\Rightarrow$ Solve N^2 scalar problems
    $\Rightarrow$ Closed-form solution based on projected soft-thresholding
  - Computational complexity reduced to $\mathcal{O}(N^3)$

- Test the estimates $\hat{\mathbf{H}}$ and $\hat{\mathbf{S}}$ with and without robust approach
  - $\Rightarrow$ Graphs are sampled from the small-world random graph model
  - $\Rightarrow$ We consider different types of perturbations



(a) Proportion of perturbed links (b) Proportion of perturbed links

- RFI consistently outperforms classical FI
  - $\Rightarrow$ Clear improvement in estimation of $\mathbf{S}$ with respect to $\bar{\mathbf{S}}$
- Only destroying links is the most damaging perturbation

Universidad Rey Juan Carlos

- Given $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]$, $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$ and perturbed $\bar{\mathbf{S}}$ $\Rightarrow$ Find GF/GNN
  - $\Rightarrow$ Key to our approach: postulate true $\mathbf{S}$ as an optimization variable
  - $\Rightarrow$ Perform joint optimization
  - $\Rightarrow$ Operate on the vertex domain

Outline of the talk

- Formulation for a single GF
  - $\Rightarrow$ Relaxations and algorithmic alternatives
- Formulation for multiple GFs
- Formulation for GNNs
- Generalizations to adversarial setups and future work

Robust joint graph filter ID

- Now the goal is to estimate K GFs $\{\mathbf{H}_k\}_{k=1}^K$
  - $\Rightarrow$ Are polynomials of the unknown $\mathbf{S}$ but only $\bar{\mathbf{S}}$ is observed
  - $\Rightarrow$ For each $\mathbf{H}_k$ we have M_k input/output signals $\mathbf{X}_k/\mathbf{Y}_k$
- Several GFs show up in relevant settings [Segarra17][Liu18]
  - $\Rightarrow$ Different network processes on a graph $\mathbf{Y}_k = \mathbf{H}_k \mathbf{X}_k + \mathbf{W}_k$
  - $\Rightarrow$ Graph-based multivariate time series $\mathbf{Y}_\kappa = \sum_{k=1}^K \mathbf{H}_k \mathbf{Y}_{\kappa-k} + \mathbf{X}_\kappa + \mathbf{W}_k$

Robust joint graph filter ID

Universidad Rey Juan Carlos

- Now the goal is to estimate K GFs $\{\mathbf{H}_k\}_{k=1}^K$
  - $\Rightarrow$ Are polynomials of the unknown $\mathbf{S}$ but only $\bar{\mathbf{S}}$ is observed
  - $\Rightarrow$ For each $\mathbf{H}_k$ we have M_k input/output signals $\mathbf{X}_k/\mathbf{Y}_k$
- Several GFs show up in relevant settings [Segarra17][Liu18]
  - $\Rightarrow$ Different network processes on a graph $\mathbf{Y}_k = \mathbf{H}_k\mathbf{X}_k + \mathbf{W}_k$
  - $\Rightarrow$ Graph-based multivariate time series $\mathbf{Y}_\kappa = \sum_{k=1}^K \mathbf{H}_k\mathbf{Y}_{\kappa-k} + \mathbf{X}_\kappa + \mathbf{W}_k$
- Joint identification exploits each $\mathbf{H}_k$ being a polynomial on $\mathbf{S}$

$$\min_{\mathbf{S}\in\mathcal{S},\{\mathbf{H}_k\}_{k=1}^K} \sum_{k=1}^K \alpha_k \|\mathbf{Y}_k - \mathbf{H}_k\mathbf{X}_k\|_F^2 + \lambda d(\mathbf{S},\bar{\mathbf{S}}) + \beta\|\mathbf{S}\|_0 + \sum_{k=1}^K \gamma\|\mathbf{S}\mathbf{H}_k - \mathbf{H}_k\mathbf{S}\|_F^2$$

  - $\Rightarrow$ K commutativity constraints improve estimation of $\mathbf{S}$
  - $\Rightarrow$ A better estimate of $\mathbf{S}$ leads to better estimates of $\mathbf{H}_k$
- Solved via 2-step alternating optimization

Universidad Rey Juan Carlos

- Consider an AR graph signal $\mathbf{Y}_\kappa$ of order K with exogenous input $\mathbf{X}_\kappa$

$$\mathbf{Y}_\kappa = \sum_{k=1}^{K} \mathbf{H}_k \mathbf{Y}_{\kappa-k} + \mathbf{X}_\kappa, \text{ with } \mathbf{H}_k = \sum_{r=0}^{N-1} h_{r,k} \mathbf{S}^r,$$

- Having access to $\bar{\mathbf{S}}$ and observations, we aim to solve

$$\min_{\mathbf{S}\in\mathcal{S},\{\mathbf{H}_k\}_{k=1}^K} \sum_{\kappa=K+1}^{\kappa_{max}} \left\| \mathbf{Y}_\kappa - \mathbf{X}_\kappa - \sum_{k=1}^{K} \mathbf{H}_k \mathbf{Y}_{\kappa-k} \right\|_F^2 + \lambda d(\mathbf{S}, \bar{\mathbf{S}}) + \beta \|\mathbf{S}\|_0 + \sum_{k=1}^{K} \gamma \|\mathbf{S}\mathbf{H}_k - \mathbf{H}_k\mathbf{S}\|_F^2$$

$\Rightarrow$ If exogenous input $\mathbf{X}_\kappa$ not know, use covariance norm

- Solved via block-coordinate algorithm, new GF-id step is

- Consider an AR graph signal $\mathbf{Y}_\kappa$ of order K with exogenous input $\mathbf{X}_\kappa$

$$\mathbf{Y}_\kappa = \sum_{k=1}^{K} \mathbf{H}_k \mathbf{Y}_{\kappa-k} + \mathbf{X}_\kappa, \text{ with } \mathbf{H}_k = \sum_{r=0}^{N-1} h_{r,k} \mathbf{S}^r,$$

- Having access to $\bar{\mathbf{S}}$ and observations, we aim to solve

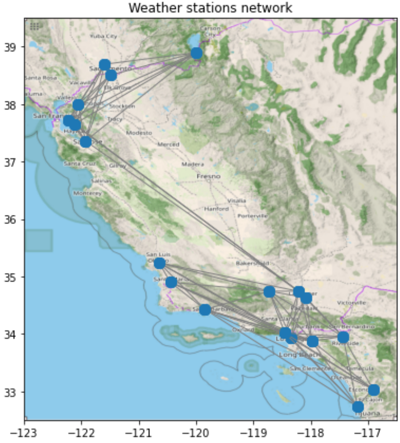
$$\min_{\mathbf{S}\in\mathcal{S},\{\mathbf{H}_k\}_{k=1}^{K}} \sum_{\kappa=K+1}^{\kappa_{max}} \left\|\mathbf{Y}_\kappa - \mathbf{X}_\kappa - \sum_{k=1}^{K} \mathbf{H}_k \mathbf{Y}_{\kappa-k}\right\|_F^2 + \lambda d(\mathbf{S}, \bar{\mathbf{S}}) + \beta\|\mathbf{S}\|_0 + \sum_{k=1}^{K} \gamma \|\mathbf{S}\mathbf{H}_k - \mathbf{H}_k\mathbf{S}\|_F^2$$

$\Rightarrow$ If exogenous input $\mathbf{X}_\kappa$ not know, use covariance norm

- Solved via block-coordinate algorithm, new GF-id step is

$$\mathbf{H}_k^{(t+1)} = \underset{\mathbf{H}_k}{\operatorname{argmin}} \sum_{\kappa=K+1}^{\kappa_{max}} \left\|\mathbf{Y}_\kappa - \mathbf{X}_\kappa - \mathbf{H}_k \mathbf{Y}_{\kappa-k} - \sum_{k'<k} \mathbf{H}_{k'}^{(t+1)} \mathbf{Y}_{\kappa-k'} - \sum_{k'>k} \mathbf{H}_{k'}^{(t)} \mathbf{Y}_{\kappa-k'}\right\|_F^2 + \sum_{k=1}^{K} \gamma \left\|\mathbf{S}^{(t)}\mathbf{H}_k - \mathbf{H}_k \mathbf{S}^{(t)}\right\|_F^2,$$

Universidad Rey Juan Carlos



NATIONAL CENTERS FOR ENVIRONMENTAL INFORMATION

Weather station network

- Nodes are weather stations in California
- Signals are temperature measurements at each station
- 5-nearest neighbor graph from geographical distance between stations

- Predict temperature 1 or 3 days in the future
  - $\Rightarrow$ Estimate $\mathbf{H}$ using 25% or 50% of the available data
- Consider LS as a naive solution and TLS-SEM as a robust baseline

Models	1-Step		3-Step	
	TTS=0.25	TTS = 0.5	TTS=0.25	TTS = 0.5
LS	$6.9 \cdot 10^{-3}$	$3.1 \cdot 10^{-3}$	$2.1 \cdot 10^{-2}$	$9.1 \cdot 10^{-3}$
LS-GF	$3.3 \cdot 10^{-3}$	$3.3 \cdot 10^{-3}$	$8.4 \cdot 10^{-3}$	$8.5 \cdot 10^{-3}$
TLS-SEM	$4.0 \cdot 10^{1}$	$3.7 \cdot 10^{-2}$	$6.8 \cdot 10^{-1}$	$5.5 \cdot 10^{-2}$
RFI	$3.4 \cdot 10^{-3}$	$3.1 \cdot 10^{-3}$	$8.5 \cdot 10^{-3}$	$7.5 \cdot 10^{-3}$
AR(3)-RFI	$\mathbf{3.2 \cdot 10^{-3}}$	$\mathbf{2.8 \cdot 10^{-3}}$	$\mathbf{7.8 \cdot 10^{-3}}$	$\mathbf{6.9 \cdot 10^{-3}}$

- Best performance achieved by joint inference assuming AR model of order 3
  - $\Rightarrow$ Follow up closely by the (separate) RFI algorithm

Outline: Robust GNN design

Universidad Rey Juan Carlos

- Given $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]$, $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$ and perturbed $\bar{\mathbf{S}}$ $\Rightarrow$ Find GF/GNN
  - $\Rightarrow$ Key to our approach: postulate true $\mathbf{S}$ as an optimization variable
  - $\Rightarrow$ Perform joint optimization
  - $\Rightarrow$ Operate on the vertex domain

Outline of the talk

- Formulation for a single GF
  - $\Rightarrow$ Relaxations and algorithmic alternatives
- Formulation for multiple GFs
- Formulation for GNNs
- Generalizations to adversarial setups and future work

Robust GNN design for perturbed GSOs

- GNNs stack layers composing pointwise nonlinearities with linear GFs

$$\mathbf{x}_1 = \sigma_1\Big(\mathbf{H}(\mathbf{h}_1\,|\mathbf{S})\mathbf{x}\Big),\ \ldots,\ \mathbf{x}_\ell = \sigma_\ell\Big(\mathbf{H}(\mathbf{h}_\ell\,|\mathbf{S})\mathbf{x}_{\ell-1}\Big),\ \ldots,\ \mathbf{y} = \sigma_L\Big(\mathbf{H}(\mathbf{h}_L\,|\mathbf{S})\mathbf{x}_{L-1}\Big)$$

  ⇒ In practice, GNNs allow for multiple hidden features

$$\mathbf{X}_\ell = \sigma_\ell\Big(\mathbf{H}(\mathbf{h}_\ell\,|\mathbf{S})\mathbf{X}_{\ell-1}\boldsymbol{\Xi}_\ell^T\Big)$$

  ⇒ Mapping GNN: $\mathbf{y} = f_{\boldsymbol{\Theta}}(\mathbf{x}\,|\mathbf{S})$, with $\boldsymbol{\Theta} = \{\mathbf{h}_\ell, \boldsymbol{\Xi}_\ell\}_{\ell=1}^L$

- Our goal is to use $\mathcal{T}$ and perturbed $\bar{\mathbf{S}}$ to estimate

  ⇒ Robust GNN parameters [Jin20],[Kenlay21]

  ⇒ Enhanced GSO

- Challenges: GNN highly nonconvex, error nonlinear in $\mathbf{h}$ / $\mathbf{H}$

  ⇒ Optimization typically addressed via SGD

Universidad Rey Juan Carlos

- Hence, we can approach the robust GNN design as

$$\min_{\mathbf{h},\boldsymbol{\Xi},\mathbf{S}\in\mathcal{S}} \sum_{m=1}^{M} \|\mathbf{y}_m - f_{[\mathbf{h},\boldsymbol{\Xi}]}(\mathbf{x}_m\,|\mathbf{S})\|^2 + \alpha d(\mathbf{S},\bar{\mathbf{S}}) + \lambda\|\mathbf{S}\|_1$$

$\Rightarrow$ As before, we could declare variables $\{\mathbf{H}_\ell\}$ and use commutativity

$\Rightarrow$ However, in GNNs the polynomials are of low degree (2, 3...)

$$\mathbf{X}_\ell = \sigma_\ell\Big(\mathbf{H}(\mathbf{h}_\ell)\mathbf{X}_{\ell-1}\boldsymbol{\Xi}_\ell^T\Big), \text{ with } \mathbf{H}(\mathbf{h}_\ell\,|\mathbf{S}) = h_{0,\ell}\mathbf{I} + h_{1,\ell}\mathbf{S} + h_{2,\ell}\mathbf{S}^2$$

- Hence, we can approach the robust GNN design as

$$\min_{\mathbf{h},\boldsymbol{\Xi},\mathbf{S}\in\mathcal{S}} \sum_{m=1}^{M} \|\mathbf{y}_m - f_{[\mathbf{h},\boldsymbol{\Xi}]}(\mathbf{x}_m|\mathbf{S})\|^2 + \alpha d(\mathbf{S},\bar{\mathbf{S}}) + \lambda\|\mathbf{S}\|_1$$

  $\Rightarrow$ As before, we could declare variables $\{\mathbf{H}_\ell\}$ and use commutativity

  $\Rightarrow$ However, in GNNs the polynomials are of low degree (2, 3...)

$$\mathbf{X}_\ell = \sigma_\ell\Big(\mathbf{H}(\mathbf{h}_\ell)\mathbf{X}_{\ell-1}\boldsymbol{\Xi}_\ell^T\Big), \text{ with } \mathbf{H}(\mathbf{h}_\ell|\mathbf{S}) = h_{0,\ell}\mathbf{I} + h_{1,\ell}\mathbf{S} + h_{2,\ell}\mathbf{S}^2$$

- Optimizing over $\{h_{k,\ell}\}$ can be a possibility, the algorithm proceeds in 3 steps

  $\Rightarrow$ **Step 1:** $\mathbf{h}^{(t+1)} = \arg\min_{\mathbf{h}} \sum_{m=1}^{M} \|\mathbf{y}_m - f_{[\mathbf{h},\boldsymbol{\Xi}^{(t)}]}(\mathbf{x}_m|\mathbf{S}^{(t)})\|^2$

  $\Rightarrow$ **Step 2:** $\boldsymbol{\Xi}^{(t+1)} = \arg\min_{\boldsymbol{\Xi}} \sum_{m=1}^{M} \|\mathbf{y}_m - f_{[\mathbf{h}^{(t+1)},\boldsymbol{\Xi}]}(\mathbf{x}_m|\mathbf{S}^{(t)})\|^2$

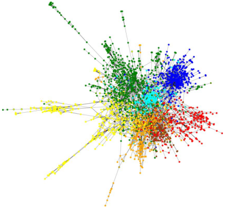
  $\Rightarrow$ **Step 3:** $\mathbf{S}^{(t+1)} = \arg\min_{\mathbf{S}\in\mathcal{S}} \sum_{m=1}^{M} \|\mathbf{y}_m - f_{[\mathbf{h}^{(t+1)},\boldsymbol{\Xi}^{(t+1)}]}(\mathbf{x}_m|\mathbf{S})\|^2$
  $+\alpha d(\mathbf{S},\bar{\mathbf{S}}) + \lambda\|\mathbf{S}\|_1$

- SGD needs to be used, Steps 1-2 standard via backpropagation

Selected datasets

Universidad Rey Juan Carlos

- We test our approach on 2 citation nets and 3 webpage nets
  - ⇒ Nodes are scientific papers and edges citations among them
  - ⇒ Nodes are webpages and edges hyperlinks
  - ⇒ Node features indicate the presence of words from a fixed dictionary
- **Cora dataset**[a]: $N = 2708$ and $E = 10556$
  - ⇒ 1433 node features, 7 classes of nodes
- **Citeseer dataset**[b]: $N = 3327$ and $E = 9228$ edges
  - ⇒ 3703 node features, 6 classes of nodes
- **WebKB1 dataset**[c]: $N = 183/251$ and $E = 295/499$
  - ⇒ 1703 node features, 5 classes of webs (course, faculty...)

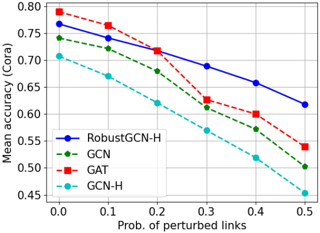


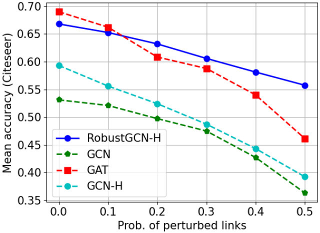
[a] https://networkrepository.com/cora.php
[b] https://networkrepository.com/citeseer.php
[c] https://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/

Numerical Evaluation: GNNs (I)

Universidad Rey Juan Carlos

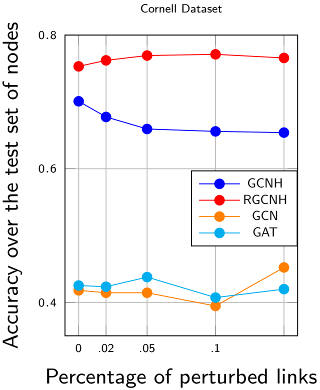
- We test our robust GNN-H relative to other competitors
  - $\Rightarrow$ Cora and Citeseer / Classical GCN and GAT
  - $\Rightarrow$ A GCN with learnable H (i.e., our model ignoring perturbations)

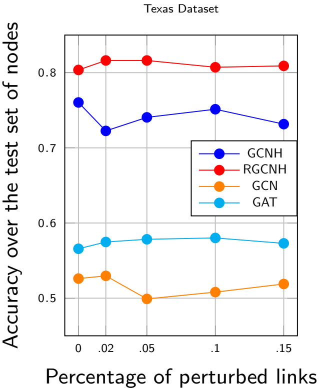




- **Results:**
  - $\Rightarrow$ As $d(\bar{\mathbf{S}}, \mathbf{S})$ increases: accuracy down $\Rightarrow$ Relevance of robust designs
  - $\Rightarrow$ When no errors, GAT outperforms, our robust GNN performs similarly
  - $\Rightarrow$ When errors, robust GNN outperforms and degrades less noticeably

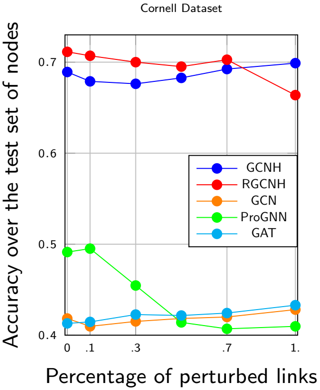
- Effect of perturbation on the graph $\Rightarrow$ Edge rewiring
- Cornell and Texas datasets (WebKB1)

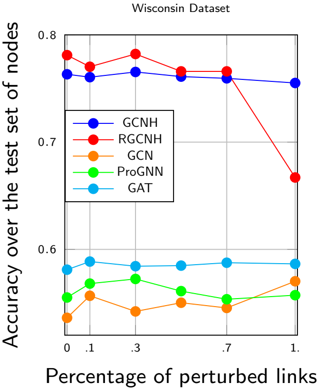




- RGCNH even improves the unperturbed case

- Using information about the perturbation
  - ⇒ Only a subset of nodes with edges perturbed





- RGCNH leverages the prior information

- Considering $\mathbf{S}$ as an optimization variable offers advantages...
- ... but it is problematic if powers are higher than 3, 4

Alternative formulation of Robust GNN

- Considering $\mathbf{S}$ as an optimization variable offers advantages...
- ... but it is problematic if powers are higher than 3, 4
- Alternative formulation

$$\min_{\boldsymbol{\Theta},\mathbf{H},\mathbf{S}\in\mathcal{S}} \mathcal{L}\left(f_{\boldsymbol{\Theta}}(\mathbf{H},\mathbf{X}),\mathcal{Y}\right)+\alpha d(\mathbf{S},\bar{\mathbf{S}})+\lambda\gamma(\mathbf{S})+\delta\|\mathbf{HS}-\mathbf{SH}\|_2^2$$

$\Rightarrow$ In this case, the recursion is defined by

$$\mathbf{X}_\ell=\sigma_\ell\left(\mathbf{H}\mathbf{X}_{\ell-1}\boldsymbol{\Theta}_\ell\right)$$

$\Rightarrow$ OK: avoids powers of $\mathbf{S}$

$\Rightarrow$ OK: commutativity term promotes $\mathbf{H}$ as a polynomial of $\mathbf{S}$ $\Rightarrow$ GF

$\Rightarrow$ OK: less parameters

$\Rightarrow$ KO: new optimization variable

Alternative formulation of Robust GNN (II)

Universidad Rey Juan Carlos

- Solve in three steps
  - $\Rightarrow$ **Step 1:** $\boldsymbol{\Theta}^{(t+1)} = \arg\min_{\boldsymbol{\Theta}} \mathcal{L}\left(f_{\boldsymbol{\Theta}}(\mathbf{H}^{(t)}, \mathbf{X}), \mathcal{Y}\right)$
  - $\Rightarrow$ **Step 2:** $\mathbf{H}^{(t+1)} = \arg\min_{\mathbf{H}} \mathcal{L}\left(f_{\boldsymbol{\Theta}^{(t+1)}}(\mathbf{H}, \mathbf{X}), \mathcal{Y}\right) + \delta\|\mathbf{H}\mathbf{S}^{(t)} - \mathbf{S}^{(t)}\mathbf{H}\|_2^2$
  - $\Rightarrow$ **Step 3:** $\mathbf{S}^{(t+1)} = \arg\min_{\mathbf{S}\in\mathcal{S}} \alpha d(\mathbf{S}, \bar{\mathbf{S}}) + \lambda\gamma(\mathbf{S}) + \delta\|\mathbf{H}^{(t+1)}\mathbf{S} - \mathbf{S}\mathbf{H}^{(t+1)}\|_2^2$
- Alternating optimization
  - $\Rightarrow$ First two steps via SGD
  - $\Rightarrow$ Gradient of commutativity term - linear
  - $\Rightarrow$ Step 3 - convex!

- Given $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]$, $\mathbf{Y} = [\mathbf{y}_1, ..., \mathbf{y}_M]$ and perturbed $\bar{\mathbf{S}}$ $\Rightarrow$ Find GF/GNN
  - $\Rightarrow$ Key to our approach: postulate true $\mathbf{S}$ as an optimization variable
  - $\Rightarrow$ Perform joint optimization
  - $\Rightarrow$ Operate on the vertex domain

Outline of the talk

- Formulation for a single GF
  - $\Rightarrow$ Relaxations and algorithmic alternatives
- Formulation for multiple GFs
- Formulation for GNNs
- Generalizations to adversarial setups and future work

Universidad Rey Juan Carlos

Considering adversarial setups

- So far we have considered that perturbations were arbitrary
  - $\Rightarrow$ As a result we focused in finding the best fit for the observations
  - $\Rightarrow$ Mathematically

$$\min_{\mathbf{S}\in\mathcal{S},\mathbf{H}} \|\mathbf{Y}-\mathbf{H}\mathbf{X}\|_F^2+\lambda r_{\delta_1}(\mathbf{S}-\bar{\mathbf{S}})+\beta r_{\delta_2}(\mathbf{S})+\gamma\|\mathbf{S}\mathbf{H}-\mathbf{H}\mathbf{S}\|_F^2$$

Considering adversarial setups

- So far we have considered that perturbations were arbitrary
  - $\Rightarrow$ As a result we focused in finding the best fit for the observations
  - $\Rightarrow$ Mathematically

$$\min_{\mathbf{S}\in\mathcal{S},\mathbf{H}} \|\mathbf{Y}-\mathbf{H}\mathbf{X}\|_F^2+\lambda r_{\delta_1}(\mathbf{S}-\bar{\mathbf{S}})+\beta r_{\delta_2}(\mathbf{S})+\gamma\|\mathbf{S}\mathbf{H}-\mathbf{H}\mathbf{S}\|_F^2$$

- However, what if perturbations are adversarial or focus on worst-case design
  - $\Rightarrow$ Min / max formulation

$$\min_{\mathbf{H}}\max_{\mathbf{S}\in\mathcal{S}} \|\mathbf{Y}-\mathbf{H}\mathbf{X}\|_F^2+\lambda r_{\delta_1}(\mathbf{S}-\bar{\mathbf{S}})+\beta r_{\delta_2}(\mathbf{S})+\gamma\|\mathbf{S}\mathbf{H}-\mathbf{H}\mathbf{S}\|_F^2$$

- Saddle point optimization $\Rightarrow$ guarantees if convex / concave
  - $\Rightarrow$ Most NN do not satisfy the above
  - $\Rightarrow$ Careful reformulations are prudent

Considering graph-perturbations for other GSP problems

- Graph-perturbations are critical in most GSP tasks but not accounted for
  - $\Rightarrow$ $\mathbf{S}$ as an optimization variable in other GSP tasks
- Incorporate prior information about the perturbations or the graph
- Instead of first learning the graph and then solving the GSP task...
  - $\Rightarrow$ ...jointly learn the graph and solving the GSP task

Considering graph-perturbations for other GSP problems

- Graph-perturbations are critical in most GSP tasks but not accounted for
  - ⇒ $\mathbf{S}$ as an optimization variable in other GSP tasks
- Incorporate prior information about the perturbations or the graph
- Instead of first learning the graph and then solving the GSP task...
  - ⇒ ...jointly learn the graph and solving the GSP task

Exploiting prior information about the graph topology

- Most applications only use the fact that $\mathbf{S}$ is sparse
- Prior information is key to accurately estimating the graph topology
- Assuming graph is a random realization and leverage statistical priors
  - ⇒ Efforts should focus on identifying models suited for the task at hand
- Assuming we have access to other related graphs
  - ⇒ Prior work based on reference graph with a similar density of motifs

Universidad Rey Juan Carlos

- Related publications:

⇒ V. M. Tenorio, S. Rey, F. Gama, S. Segarra, and A. G. Marques, "A robust alternative for graph convolutional neural networks via graph neighborhood filters," in Asilomar Conf. Signals, Syst., Computers, 2021.

⇒ S. Rey, V. M. Tenorio, and A. G. Marques, "Robust graph filter identification and graph denoising from signal observations," IEEE Trans. Signal Process. 2023 (arXiv:2210.08488)

⇒ V. M. Tenorio, S. Rey, and A. G. Marques, "Robust Graph Neural Network based on graph denoising," in Asilomar Conf. Signals, Syst., Computers, 2023.

⇒ V. M. Tenorio, S. Rey and A. G. Marques, "Robust blind deconvolution and graph denoising", in IEEE Int. Conf. Acoustics, Speech Signal Process. (ICASSP), 2024.

- Code: https://github.com/reysam93

Thanks!

References

⇒ [Segarra17] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," IEEE Trans. Signal Process., vol. 65, no. 15, pp. 4117–4131, 2017.

⇒ [Liu18] J. Liu, E. Isufi, and G. Leus, "Filter design for autoregressive moving average graph filters," IEEE Trans. Signal Process. Inf. Netw., vol. 5, no. 1, pp. 47–60, 2018.

⇒ [Miettinen19] J. Miettinen, S. A. Vorobyov, and E. Ollila, "Modelling graph errors: Towards robust graph signal processing," arXiv preprint arXiv:1903.08398, 2019

⇒ [Ceci20] E. Ceci and S. Barbarossa, "Graph signal processing in the presence of topology uncertainties," IEEE Trans. Signal Process., vol. 68, pp. 1558–1573, 2020.

⇒ [Nguyen22] H. S. Nguyen, Y. He, and H. T. Wai, "On the stability of low pass graph filter with a large number of edge rewires," in IEEE Intl. Conf. Acoustics, Speech Signal Process. (ICASSP), 2022, pp. 5568–5572.

⇒ [Jin20] W. Jin, et al, "Graph structure learning for robust graph neural networks," in Intl. Conf. Knowl. Discovery Data Mining (ACM SIGKDD), 2020, pp. 66–74.

⇒ [Kenlay21] H. Kenlay, D. Thano, and X. Dong, "On the stability of graph convolutional neural networks under edge rewiring," in IEEE Intl. Conf. Acoustics, Speech Signal Process. (ICASSP), 2021, pp. 8513–8517.