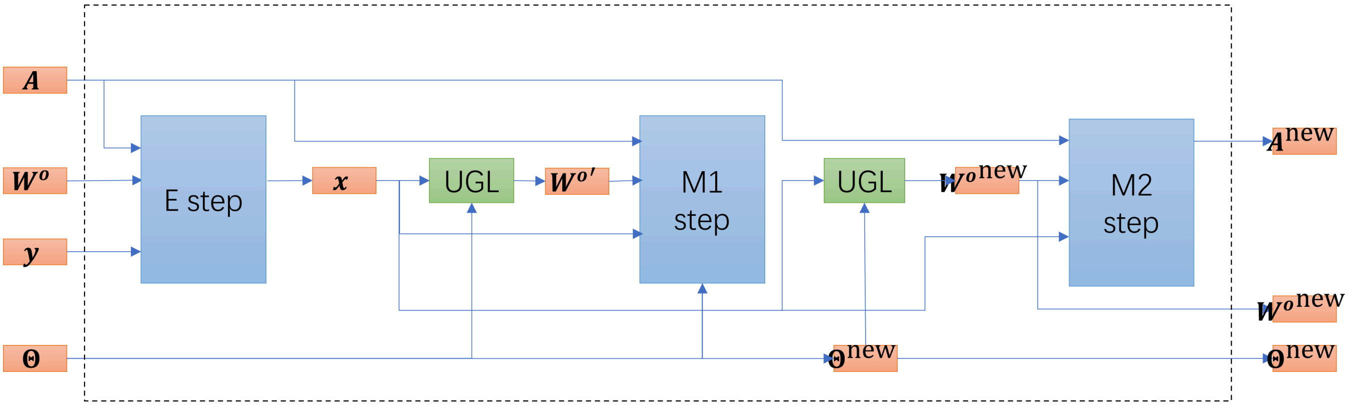
Discussion Dec 29

unrolled-EM or EM-style training of GNN?

1 GEM Framework

Energy constraint: $\|\mathbf{L}\|_F^2 = \sum_{i\in\mathcal{V}} d_i + 2\sum_{(i,j)\in\mathcal{E}} w_{ij}^2 = c^2$.

Notations: edge weights of all potential edges $\mathbf{W}^o$, actual weight matrix $\mathbf{W} = \mathbf{W}^o \circ \mathbf{A}$, corresponding Laplacian matrix $\mathbf{L} = \mathrm{diag}(\mathbf{W}\mathbf{1}) - \mathbf{W}$



1.1 E step

1. Solve the following equation by CG:

$$(\mathbf{I} + \mu\mathbf{L})\mathbf{x} = \mathbf{y}$$

2. Regenerate the graph based on new $\mathbf{x}$:

$$\mathbf{W}^{o\prime} = \mathtt{UGL}(\mathbf{x};\Theta), \quad \mathbf{W}' = \mathbf{W}^{o\prime} \circ \mathbf{A}, \quad \mathbf{L}' = \mathrm{diag}(\mathbf{W}'\mathbf{1}) - \mathbf{W}'$$

3. Rescale to $\|\mathbf{L}'\|_F^2 = c^2$ (PGD under sphere constraint)

1.2 M1 step

$$\min \mathcal{L}_1(\Theta) = \frac{1}{N}\sum_{k=1}^{N} \mathbf{x}^{(k)\top}\mathbf{L}'\mathbf{x}^{(k)} - \log|\mathbf{L}'|$$

1. Compute proxy loss: **detach** $\mathbf{R} = (\mathbf{L}' + \mathbf{J})^{-1} = \mathbf{L}'^{\dagger} + \mathbf{J}$,

$$\tilde{\mathcal{L}}_1(\Theta) = \mathrm{tr}(\mathbf{S}\mathbf{L}') - \mathrm{tr}(\mathbf{R}\mathbf{L}') = \sum_{(i,j)\in\mathcal{E}} w'_{ij}\frac{1}{N}\sum_{k=1}^{N}(x_i^{(k)} - x_j^{(k)})^2 - \sum_{(i,j)\in\mathcal{E}} w'_{i,j}(\mathbf{e}_i - \mathbf{e}_j)^\top\mathbf{R}(\mathbf{e}_i - \mathbf{e}_j)$$

2. Gradient step:

$$\Theta^{\mathrm{new}} = \Theta - \delta\nabla\tilde{\mathcal{L}}(\Theta)$$

3. Regenerate the graph based on the new Θ^{new}

$$\mathbf{W}^{o\mathrm{new}} = \mathtt{UGL}(\mathbf{x};\Theta^{\mathrm{new}}), \quad \mathbf{W}^{\mathrm{new}} = \mathbf{W}^{o\mathrm{new}} \circ \mathbf{A}, \quad \mathbf{L}^{\mathrm{new}} = \mathrm{diag}(\mathbf{W}^{\mathrm{new}}\mathbf{1}) - \mathbf{W}^{\mathrm{new}}$$

4. Rescale (PGD under sphere constraint)

1.3 M2 step

$$\min \mathcal{L}_2(\mathbf{A}) = \frac{1}{N}\sum_{k=1}^{N} \mathbf{x}^{(k)\top}\mathbf{L}\mathbf{x}^{(k)} - \log|\mathbf{L}| + \gamma\|\mathbf{A}\|_{1,\text{off}}, \quad \mathbf{W} := \mathbf{W}^{o\text{new}} \circ \mathbf{A},\ \mathbf{L} := \operatorname{diag}(\mathbf{W}\mathbf{1}) - \mathbf{W}$$

PGD solution (upper-right only, multiple iterations)

$$\mathbf{A}^{\text{new}} = \Pi_{[0,1]^{N\times N}}\left(S_{\eta\gamma}\left(\mathbf{A} - \eta\mathbf{W}^{o\text{new}}(\tilde{\mathbf{S}} - \tilde{\mathbf{R}})\right)\right), \quad \tilde{S}_{ij} = S_{ii} + S_{jj} - 2S_{ij}, \quad \tilde{R}_{ij} = R_{ii} + R_{jj} - 2R_{ij}$$

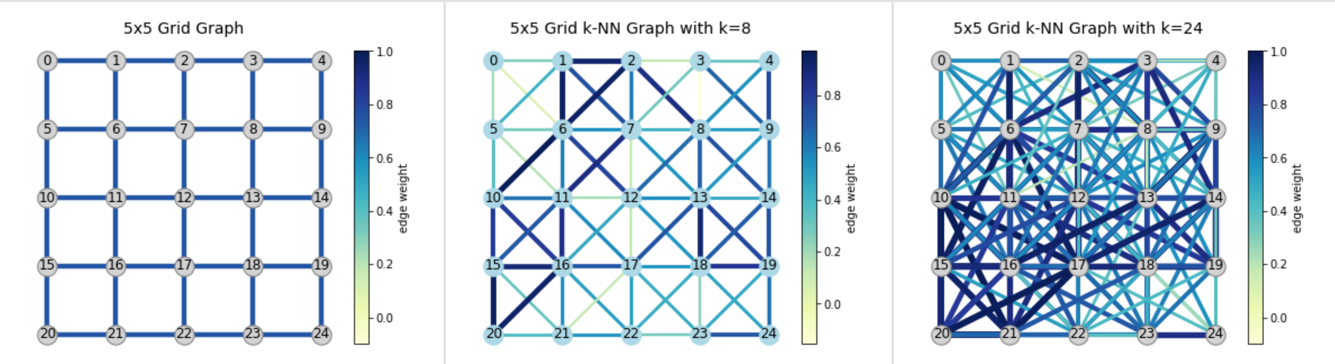
Element-wise updating formula: *non-detached* $\mathbf{R} = (\mathbf{L}^{\text{new}} + \mathbf{J})^{-1} = \mathbf{L}^{\text{new}\dagger} + \mathbf{J}$, for each $(i,j) \in \mathcal{E}$,

$$A_{ij} = \Pi_{[0,1]}\left(S_{\eta\gamma}\left(A_{ij} - \eta W_{ij}^{o\text{new}}\left(\frac{1}{N}\sum_{k=1}^{N}(x_i^{(k)} - x_j^{(k)})^2 - (\mathbf{e}_i - \mathbf{e}_j)^\top\mathbf{R}(\mathbf{e}_i - \mathbf{e}_j)\right)\right)\right)$$

2 Implementation

2.0 Sparse setup

- Real graph: 5×5 grid
- Guessed graph: 8-neighbor graph (with diagonal connection) (window size $3 \times 3, 5 \times 5$)
- kNN neighbor list `(N, k ** 2 - 1)`.



2.1 E step

compute CG

2.2 M step---Compute $(\mathbf{e}_i - \mathbf{e}_j)^\top\mathbf{R}(\mathbf{e}_i - \mathbf{e}_j)$ for each $(i,j) \in \mathcal{E}$

Methods

Method 1 (Reference point) Fix a reference point r, compute $(\mathbf{L} + \mathbf{J})^{-1}(\mathbf{e}_i - \mathbf{e}_r)$ for each i.
$\mathbf{R}(\mathbf{e}_i - \mathbf{e}_j) = \mathbf{R}(\mathbf{e}_i - \mathbf{e}_r) - \mathbf{R}(\mathbf{e}_j - \mathbf{e}_r)$

Advantage: Solving $n-1$ linear equations in total.

Memory cost: host $(n-1) \times n$ matrix for solutions of $\mathbf{R}(\mathbf{e}_i - \mathbf{e}_r)$, in-place operation to compute $\tilde{R}$

Examination: $\operatorname{tr}(\mathbf{RL}) = n - 1 = \sum_{(i,j)\in\mathcal{E}} W_{ij}^{o} A_{ij}(R_{ii} + R_{jj} - 2R_{ij})$

Problem: residual for batch CG is large: 5 iterations 10^{-1}, 10 iterations 5×10^{-3}

Alternative: ***Preconditioned CG***

Key problem: the *condition number*?

Use normalized Laplacian matrix: $\mathbf{L}_n = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$. The eigenvalues are restricted to [0,2]

Precondition matrix $\mathbf{M} = \mathbf{D} + \mathbf{J} = \mathbf{D} + \frac{1}{n}\mathbf{1}\mathbf{1}^\top$

Sherman-Morrison:

$$\mathbf{M}^{-1} = \mathbf{D}^{-1} - \frac{\frac{1}{n}\mathbf{D}^{-1}\mathbf{1}\mathbf{1}^\top\mathbf{D}^{-1}}{1+\frac{1}{n}\mathbf{1}^\top\mathbf{D}^{-1}\mathbf{1}}, \quad (\mathbf{M}^{-1})_{ij} = d_{ij}^{-1} - \frac{d_i^{-1}d_j^{-1}}{n+\sum_i d_{ii}^{-1}}, \ \mathbf{M}^{-1} = \operatorname{diag}(\mathbf{d}^{-1}) - \frac{\mathbf{d}^{-1}(\mathbf{d}^{-1})^\top}{n+\mathbf{1}^\top\mathbf{d}^{-1}}$$

$$(\mathbf{M}^{-1}\mathbf{r})_i = \frac{r_i}{d_i} - \frac{\sum_j \frac{r_j}{d_i d_j}}{n+\sum_i \frac{1}{d_{ii}}} = d_i^{-1}\Big(r_i - \frac{\sum_j r_j d_j^{-1}}{n+\sum_i d_i^{-1}}\Big), \quad \mathbf{M}^{-1}\mathbf{r} = \mathbf{u}\circ\mathbf{r} - \frac{\mathbf{u}^\top\mathbf{r}}{n+\mathbf{1}^\top\mathbf{u}}\mathbf{u}$$

Results: 5 iterations 5×10^{-2}, 10 iterations 1×10^{-3}

3 Experimental Results

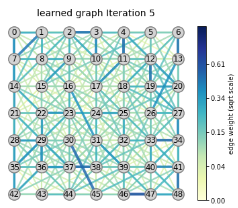
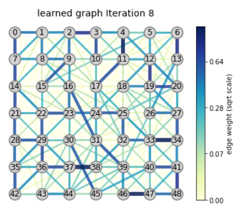
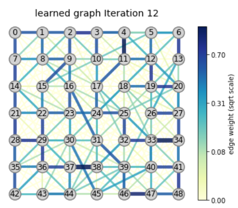
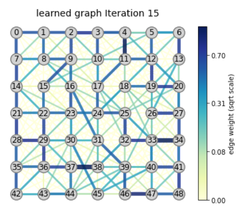
7×7 grid, window size 5×5. learn a grid graph. 512 node points, noise level $\sigma = 0.4$

Graph learning module: $\tilde{\mathbf{x}}_i = [x_i, \mathbf{e}_i], \mathbf{f} = \mathrm{LeakyReLU}(\mathbf{W}\tilde{\mathbf{x}} + \mathbf{b}), \mathbf{e}_i \in \mathbb{R}^6, \mathbf{W} \in \mathbb{R}^{6\times 6}$

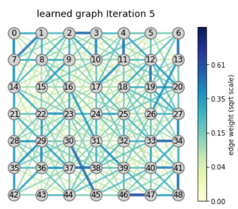
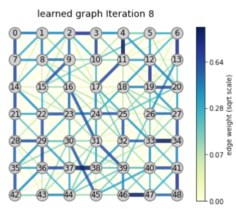
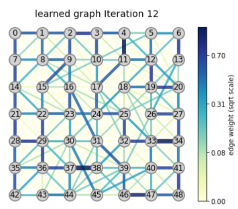
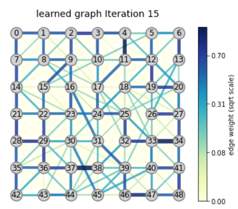
settings: $\mu = 0.15, \gamma = 0.4, \delta = 0.005, \eta = 0.04, c = 16, N_{\mathrm{PGD}} = 10, N_{\mathrm{GEM}} = 15$

same initialization of each model

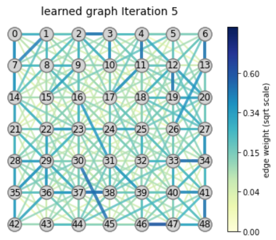
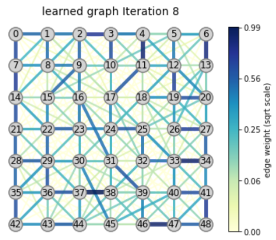
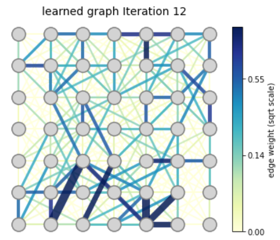
√ **Method 1:** CG, $R = (L+J)^{-1} = L^\dagger + J$

Steps	**5**	**8**	**12**	**15**
Learned graph				
edges left	395 / 396	199 / 396	157 / 196	151 / 196

Method 2: CG, $R = (L+\epsilon I)^{-1}, \epsilon = 0.005$

Steps	**5**	**8**	**12**	**15**
Learned graph				
edges left	394 / 396	198 / 396	155 / 396	151 / 396

Method 3: CHOLMOD, $\mathbf{R} = (L + \epsilon I)^{-1}$, $\epsilon = 0.005$

Steps	**5**	**8**	**12**	**15**
Learned Graph				Exploded
edge left	394 / 396	205 / 396	191 / 396	

Code link

https://github.com/SingularityUndefined/AdaptiveSparseSTForecast/blob/main/demo_sparse.ipynb

> ***Selection of*** ϵ
>
> $$\mathbf{L} = \mathbf{P}\mathrm{diag}(0, \lambda_2, \dots, \lambda_n)\mathbf{P}^\top, \mathbf{p}_1 = \frac{1}{\sqrt{n}}\mathbf{1}$$
> $$(\mathbf{L} + \mathbf{J})^{-1} = \mathbf{L}^\dagger + \mathbf{J}$$
> $$(\mathbf{L} + \epsilon\mathbf{I})^{-1} = \mathbf{P}\mathrm{diag}\left(\frac{1}{\epsilon}, \frac{1}{\lambda_2 + \epsilon}, \dots, \frac{1}{\lambda_n + \epsilon}\right)\mathbf{P}^\top \approx \mathbf{L}^\dagger + \frac{1}{\epsilon}\mathbf{J}$$
>
> Approximation error:
>
> $$\mathbf{E} = (\mathbf{L} + \epsilon\mathbf{I})^{-1} - \frac{1}{\epsilon}\mathbf{J} - \mathbf{L}^\dagger = \mathbf{P}\mathrm{diag}\left(0, -\frac{\epsilon}{\lambda_2(\lambda_2 + \epsilon)}, \dots, -\frac{\epsilon}{\lambda_n(\lambda_n + \epsilon)}\right)\mathbf{P}^\top$$
> $$\|\mathbf{E}\|_2 = \frac{\epsilon}{\lambda_2(\lambda_2 + \epsilon)} \leq \frac{\epsilon}{\lambda_2^2}$$
>
> Cheeger's inequality:
>
> $$\frac{h(G)^2}{2d_{\max}} \leq \lambda_2 \leq 2h(G), \quad h(G) = \min_{S \subset V, 0 < |S| \leq n/2} \frac{|\partial S|}{|S|}$$
>
> for a grid graph with $k \times k$ windowed neighbors, $d_{\max} = k^2 - 1$. For a square region of $a \times b$, $|S| \approx ab \leq n^2/2$, $|\partial S| \approx k^2 + 2(a+b)\frac{k}{2}$
>
> $$\frac{|\partial S|}{|S|} \approx \frac{k^2 + (a+b)k}{ab} \geq \frac{k^2 + 2\sqrt{ab}k}{ab}, \quad h(G) \approx \frac{2k^2}{n^2} + \frac{2kn/\sqrt{2}}{n^2/2} = \frac{2\sqrt{2}k}{n} + \frac{2k^2}{n^2} = (\frac{\sqrt{2}k}{n} + 1)^2 - 1$$
>
> Thus, $\lambda_2^2 \geq (\frac{h(G)^2}{2(k^2-1)})^2 \geq \epsilon$.
>
> Estimate $\lambda_2 \geq \frac{1}{2 \times 8}(\frac{6\sqrt{2}}{5} + \frac{18}{25}) \approx 0.21, \epsilon \approx 4 \times 10^{-3}$