Learning Expanding Graphs for Signal Interpolation

Bishwadeep Das and Elvin Isufi

B.Das@tudelft.nl

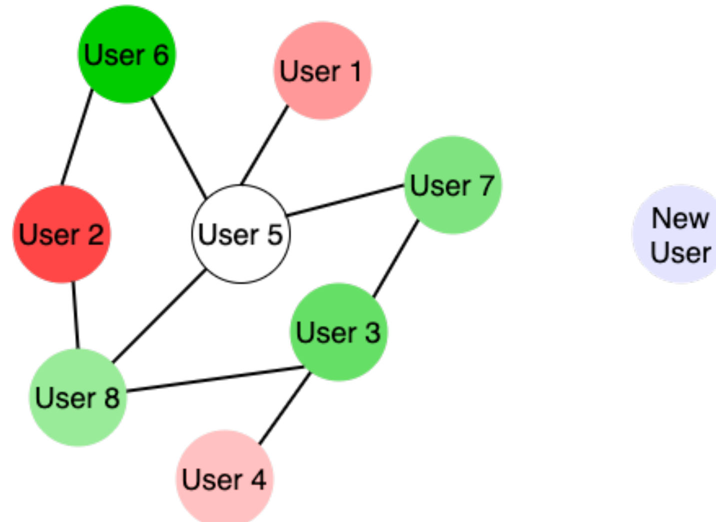
Introduction

- Typically, data Processing on graphs operations work with fixed-size graphs
- Graphs often grow in size
- This makes processing data over expanding graphs a challenge

Example: Recommendation Systems

Ratings	Item 1	Item 2	---	Item I
User 1		?		
User 2		?	?	?
-----	?		?	
User 8		?		?
New	?	?	?	?

Ratings Matrix



User graph for one item

- Graph filters process ratings over user graph to predict preferences for existing users[1] (white cells of matrix)
- New user has no data, cannot attach to the user graph

1. Huang, W., et. al, Rating prediction via graph signal processing

Related works and Gap

- All approaches rely on some information about the new node to operate, be it signal (Topology ID, Link Prediction), its connectivity (Link Prediction, related works)

- Existing works on expanding graphs require incoming node connectivity[2,3], or estimate it from features[4]

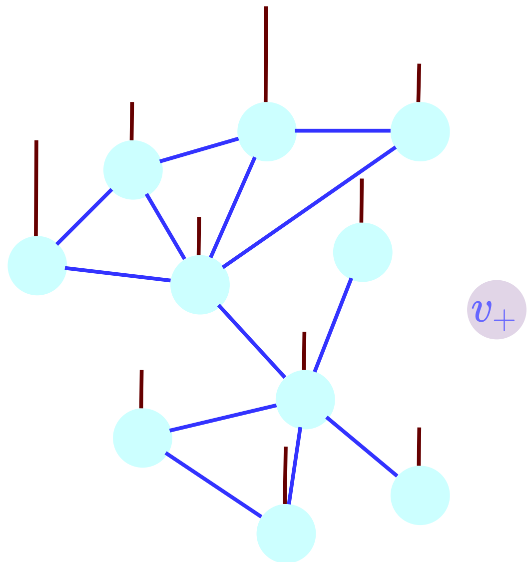
Gap: Find a way to figure connectivity and subsequent data-processing for new nodes approaching a graph when no information is available

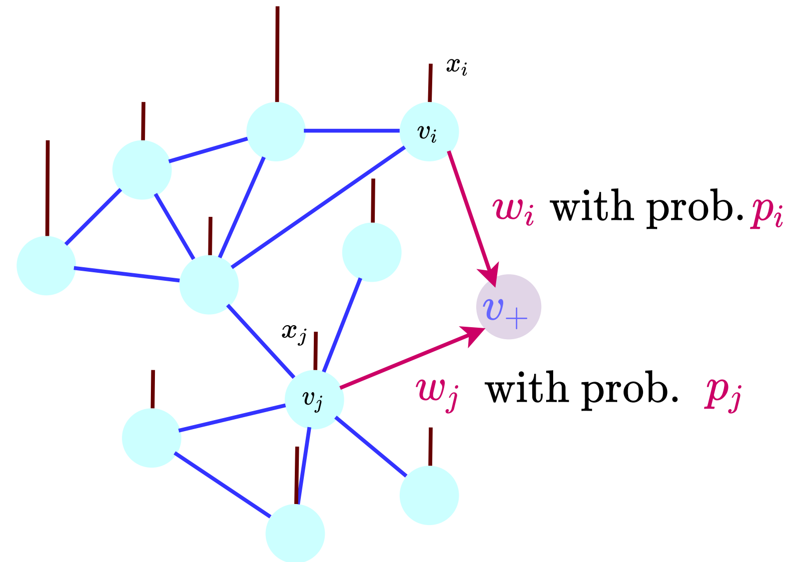
TU Delft

2. Shen et. al., Online Graph Adaptive Learning With Scalability and Privacy
3. Venkitaraman et. al., Recursive Prediction of Graph Signals With Incoming Nodes
4. Dornaika et. al., Efficient dynamic graph construction for inductive semi-supervised learning

Problem Formulation

We consider a stochastic attachment model[5,6]





$$\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A} \in \mathbb{R}^{N \times N}, \mathbf{x} \in \mathbb{R}^{N}\}$$

$$\mathcal{G}_+ = \{\mathcal{V}_+, \mathcal{E}_+, \mathbf{A}_+ \in \mathbb{R}^{N+1 \times N+1}, \mathbf{x}_+ \in \mathbb{R}^{N+1}\}$$

- Node v_+ attaches to v_i with probability p_i and edge weight w_i
- Edges directed towards v_+
- Attachment vector $\mathbf{a}_+ \in \mathbf{R}^N$, $[\mathbf{a}_+]_i = w_i$ with prob. p_i

5. Erdos, P. and Rényi, A., On the evolution of random graphs
6. Barabási, A.L. and Albert, R., Emergence of scaling in random networks

TU Delft

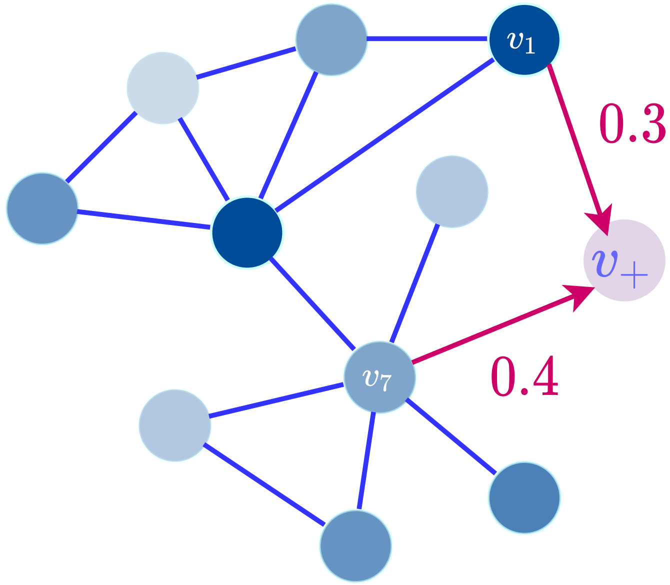
Prob. Formulation (contd.)

- Expanded adjacency matrix: $\mathbf{A}_+ = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{a}_+^\top & 0 \end{bmatrix}$
- v_+ has signal x_+, we have the expanded signal $\mathbf{x}_+ = [\mathbf{x}, x_+]^\top$
- $\mathbf{a}_+$ is an element-wise independent weighted Bernoulli random vector
- Its expectation is $\mathbb{E}[\mathbf{a}_+] = \mathbf{w} \circ \mathbf{p}$ and covariance $\mathbf{\Sigma}_+ = \mathrm{diag}(\mathbf{w}^{\circ 2} \circ \mathbf{p} \circ (\mathbf{1} - \mathbf{p}))$
- The adj. matrix after attachment obeys $\mathbb{E}[\mathbf{A}_+] = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ (\mathbf{p} \circ \mathbf{w})^\top & 0 \end{bmatrix}$

TU Delft

Adaption to a task

- Main task is to solve the parameters $\mathbf{w}, \mathbf{p}$ relative to a task
- Use training set $\mathcal{T} = \{(v_{t+}, x_{t+}, \mathbf{a}_{t+}, \mathbf{b}_{t+})\}_t$ for empirical risk minimisation
- v_{t+} : t-th node sample, x_{t+} : incoming node signal/ label
- $\mathbf{a}_{t+}$: sample attachment pattern , $\mathbf{b}_{t+}$: binary sample attachment pattern



1. v_+
2. $x_+ =$ 
3. $\mathbf{a}_+ = [0.3,0,0,0,0,0,0.4,0,0,0]^\top$
4. $\mathbf{b}_+ = [1,0,0,0,0,0,1,0,0,0]^\top$

Adaption to a task

- Task-specific cost $f_{\mathcal{T}}(\mathbf{p}, \mathbf{w}, \mathbf{x}_{t+})$

We solve

$$\min_{\mathbf{p},\mathbf{w}}. \mathbb{E}\left[f_{\mathcal{T}}(\mathbf{p}, \mathbf{w}, \mathbf{x}_{t+})\right] + g_{\mathcal{T}}(\mathbf{p}, \mathbf{b}_{t+}) + h_{\mathcal{T}}(\mathbf{w}, \mathbf{a}_{t+})$$

$$\text{subject to} \quad \mathbf{p} \in [0,1]^N, \mathbf{w} \in \mathcal{W}$$

$g_{\mathcal{T}}(\cdot), h_{\mathcal{T}}(\cdot)$ act as regularisers, $\mathscr{W}$: constraint set for edge weights

TU Delft

Task: Interpolation at incoming node

- Predict signal at an incoming node with no prior information
- Node attaches to $\mathscr{G}$, expanded signal $\mathbf{x}_+ = [\mathbf{x}, 0]^\top$ before interpolation
- For interpolation we use FIR graph filters[7] with shift operator $\mathbf{A}_+$
- Filter Output $\mathbf{y}_+ = \sum_{l=1}^{L} h_l \mathbf{A}_+^l \mathbf{x}_+$, filter $\mathbf{h} = [h_1, \ldots, h_L]^\top$
- Interested in the error $\mathbb{E}[([\mathbf{y}_+]_{N+1} - x_+)^2]$

7. Sandryhaila, A. and Moura, J.M., Discrete signal processing on graphs

Task: Interpolation at incoming node

The MSE is

$$\mathrm{MSE}(\mathbf{p}, \mathbf{w}) = ||(\mathbf{w} \circ \mathbf{p})^\top \mathbf{A}_x \mathbf{h} - x_+^\star||_2^2 + \mathbf{h}^\top \mathbf{A}_x^\top \mathbf{\Sigma}_+ \mathbf{A}_x \mathbf{h}$$

- Here, $((\mathbf{w} \circ \mathbf{p})^\top \mathbf{A}_x \mathbf{h} - x_+^\star)^2$ is the bias for that node
- The term $\mathbf{h}^\top \mathbf{A}_x^\top \mathbf{\Sigma}_+ \mathbf{A}_x \mathbf{h}$ is the output variance
- We need to avoid solutions like $\mathbf{p} = \mathbf{1}_N, \mathbf{0}_N$ by using regularisers

TU Delft

Training

$$\min_{\mathbf{p},\mathbf{w}} \quad \text{MSE}_{\mathcal{T}}(\mathbf{p},\mathbf{w}) + \sum_{t=1}^{|\mathcal{T}|}\left(\mu_p||\mathbf{p}-\mathbf{b}_{t+}||_2^2 + \mu_w||\mathbf{w}-\mathbf{a}_{t+}||_2^2\right)$$

$$\text{subject to} \quad \mathbf{p} \in [0,1]^N, \mathbf{w} \in \mathcal{W}$$

- Not always convex in **p**, convex in **w**
- We use alternating projected gradient descent

Algorithm 1 Alternating projected gradient descent for (8).

Input: Graph $\mathcal{G}$, training set $\mathcal{T}$, graph signal $\mathbf{x}$, adjacency matrix $\mathbf{A}$, number of iterations K, cost C, learning rates λ_p, λ_w.
Initialization: $\mathbf{p} = \mathbf{p}^0$, $\mathbf{w} = \mathbf{w}^0$ randomly, $k = 0$.
for $k \leq K$ **do**
  $\mathbf{p}$ gradient: $\tilde{\mathbf{p}}^{k+1} = \mathbf{p}^k - \lambda_p \nabla_{\mathbf{p}} C(\mathbf{p}^k, \mathbf{w}^k)$;
  Projection: $\mathbf{p}^{k+1} = \prod_{[0,1]^N}(\tilde{\mathbf{p}}^{k+1})$;
  $\mathbf{w}$ gradient: $\mathbf{w}^{k+1} = \mathbf{w}^k - \lambda_w \nabla_w C(\mathbf{p}^{k+1}, \mathbf{w}^k)$;
  Projection: $\mathbf{w}^{k+1} = \prod_{\mathcal{W}}(\tilde{\mathbf{w}}^{k+1})$;
end for

Convex in **p** when $\mu_p \geq w_h^2 \max_{i\in\{1,\ldots,N\}}([\mathbf{A}_x\mathbf{h}]_i)^2 - ||\mathbf{w} \circ \mathbf{A}_x\mathbf{h}||_2^2$

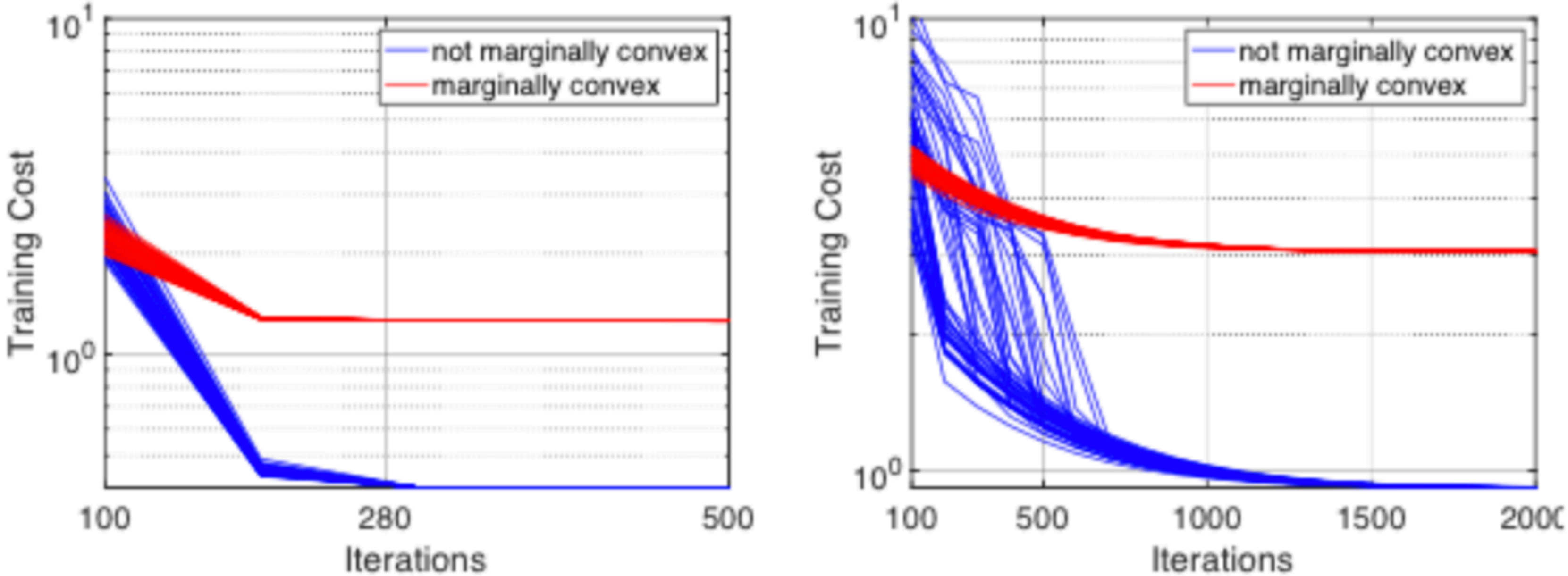
TU Delft

Numerical Results: Synthetic Graphs

- Erdos-Rényi and Barabasi-Albert, each of of 100 nodes
- Generate band-limited graph signal
- Generate $\mathscr{T}$ with corresponding $\mathbf{p}, \mathbf{w}$ pair
- Use as filter the simple shift operator to generate x_+ at each node
- Evaluate MSE over 100 such realisations for each node
- Compare with uniformly random and preferential attachment

Numerical Results: Convergence

Training with learning rates 10^{-5}



ER BA

Ensuring marginal convexity not a good idea.

Numerical : MSE at incoming node

	Erdős-Rényi			Barabasi-Albert		
	Prop.	Pref.	Rand.	Prop.	Pref.	Rand.
MSE	**0.03**	0.06	0.06	**0.05**	0.1	0.08
Std.	**0.003**	0.003	0.003	**0.006**	0.006	0.006

- Proposed outperforms rest, shows importance of task-data-topology coupling
- We also train separately for each variable , given the other
- Training only over **w** performs better because of convexity in it

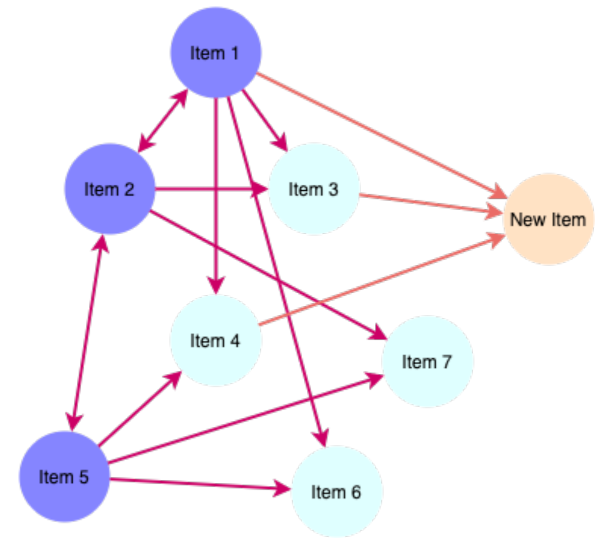
	p,**w**	only **p**	only **w**	**p**,**w**	only **p**	only **w**
MSE	**0.03**	0.07	0.039	**0.05**	0.11	0.05
Std.	**0.003**	0.003	0.003	**0.006**	0.005	0.006

Numerical Results: Item cold start collaborative filtering

Movielens 100K: 943 users, 1152 Items

	User 1	User 2	User 3
Item 1	1	2	3
Item 2	4	5	1
------	------	------	------
Item I	4	3	2

Ratings Matrix



Nearest neighbour Graph for one user

We predict ratings for new items for each user graph

Numerical Results: Violin plots

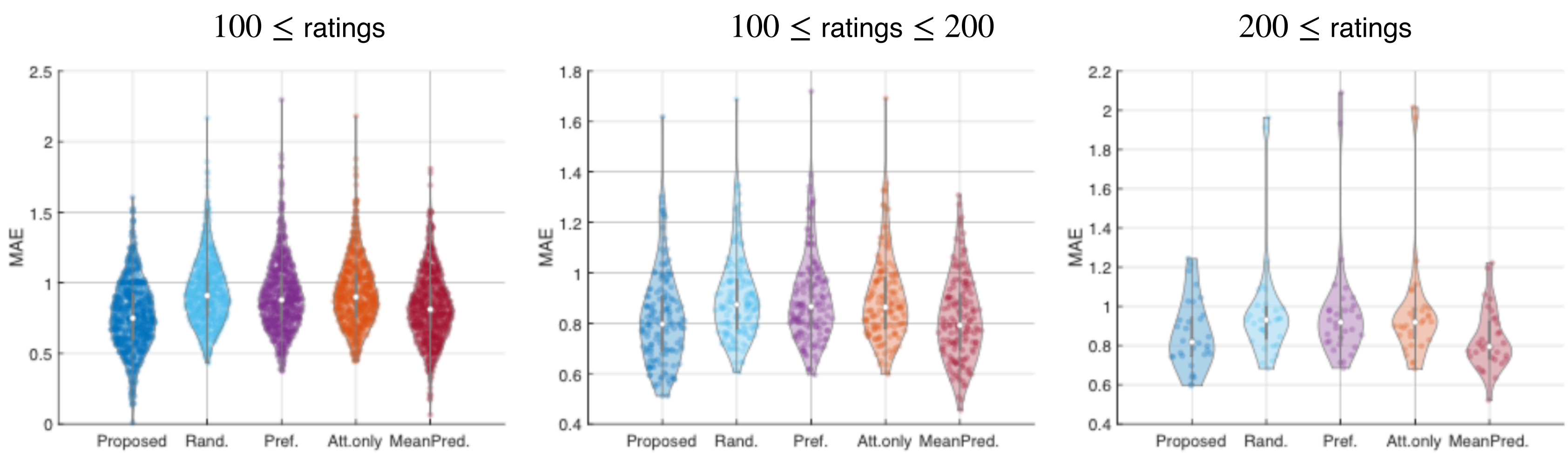


Fig. 2. Mean absolute error (MAE) violin plots for different methods and different rating densities. (Left) low ratings - proposed does best (0.75 ± 0.24), followed by mean (0.81 ± 0.24); (Centre) medium ratings - proposed and mean (0.79 ± 0.16) are tied; (Right) high ratings - mean does best (0.79 ± 0.15), followed by proposed (0.81 ± 0.17).

We do best in predicting ratings for new items in data scarcity settings

Does better than other attachments.

Shows advantage of personalised recommendations.

Conclusion

- Data, topology and task-driven attachment model for incoming nodes without prior information
- Parameterised by attachment probabilities and edge-weights, obtained by alternating projected gradient descent
- Outperforms stochastic and purely data-driven attachment

Future Work

- Process a sequence of incoming nodes without repeated re-training.
- Processing data on both the existing graph and the incoming node.

Thanks