Convolutional Graph-Tensor Net for Graph Data Completion

IJCAI 2020 workshop on Tensor Network Representations in Machine Learning

Xiao-Yang, Liu, Ming Zhu

xl2427@columbia.edu, zhumingpassional@gmail.com

Presenter: Ming Zhu

Outline

- Scenario and Related Works
- Graph-tensor Model
- Problem Formulation
- Solution
- Performance Evaluation
- Conclusion

Scenario and Related Works

- Scenario

-Data with graph structures are common.

-The data constitutes a graph-tensor.

-How to complete a graph-tensor is an important problem.

- Related works (e.g., TNN-ADMM, imputation algorithm)

-A large number of iterations.

-Time-consuming.

Graph-tensor Model

- An undirected graph, where each node has a data matrix.

- The graph Fourier transform and the inverse graph Fourier transform are defined as

$$\widetilde{\mathcal{X}}^{(k)} = \sum_{s\in[n_3]} \boldsymbol{U}(k,s)\mathcal{X}^{(s)}, \text{ for } k \in [n_3], \tag{1}$$

$$\mathcal{X}^{(k)} = \sum_{s\in[n_3]} \boldsymbol{U}^{-1}(k,s)\widetilde{\mathcal{X}}^{(s)}, \text{ for } k \in [n_3], \tag{2}$$

where $\boldsymbol{U}^{-1}$ is the inverse matrix of $\boldsymbol{U}$.

Problem Formulation

Graph-tensor nuclear-norm:

$$\|\mathcal{X}\|_{\mathrm{gTNN}} = \langle \mathcal{S}, \mathcal{I} \rangle = \sum_{i=1}^{r} |\widetilde{\mathcal{S}}(i,i,1)|, \quad r = \mathrm{rank}_{\mathcal{L}}(\mathcal{X}).$$

$$\min_{\widetilde{\mathcal{X}}} \|\widetilde{\mathcal{X}}\|_{\mathrm{gTNN}},$$

$$\text{s.t. } \mathcal{P}_{\Omega}(\mathcal{X}) = \mathcal{P}_{\Omega}(\mathcal{G}).$$

Convolutional Graph-Tensor Net (Conv GT-Net)

- 1) Imputation $\mathcal{R}^t \quad = \mathcal{P}_\Omega(\mathcal{G}) + \mathcal{P}_\Omega^\perp(\mathcal{X}^{t-1}),$

- 2) General Transform.

F(·) = Conv2D(ReLU(Conv2D(FC(·)))).

- 3) Soft-Thresholding Operator.

soft(a, λ) = max(|a| − λ, 0) · sign(a)

- 4) Inverse Transform.

$F^{-1}(\cdot)$ = FC(Conv2D(ReLU(Conv2D(·))))

Loss Function

$$
\begin{aligned}
\mathcal{L} &= \alpha \mathcal{L}_{\text{fidelity}} + \beta \mathcal{L}_{\text{inversion}}, \\
\mathcal{L}_{\text{fidelity}} &= \| P_{\Omega}^{\perp}(\mathcal{X}^{T}) - P_{\Omega}^{\perp}(\mathcal{G}) \|_F^2, \\
\mathcal{L}_{\text{inversion}} &= \frac{1}{T} \sum_{t=1}^{T} \| \mathcal{F}^{-1}(\mathcal{F}(\mathcal{X}^{t})) - \mathcal{X}^{t} \|_F^2.
\end{aligned}
$$

(fidelity of the reconstructed data)

(accuracy of the inverse function)

Structure of Conv GT-Net

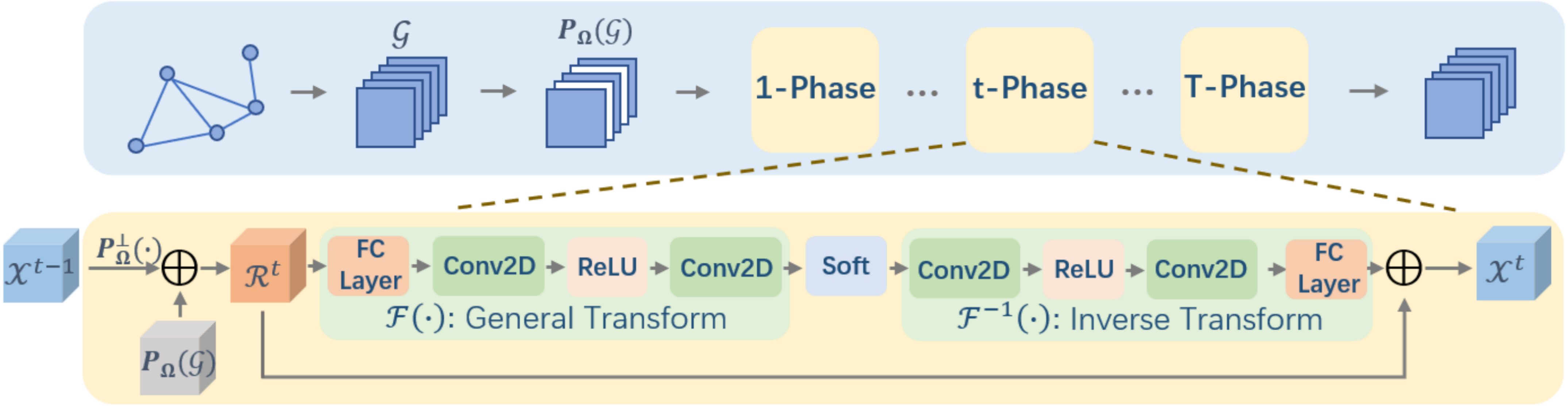

Figure 2: The structure of *Conv GT-Net*. The upper part is the data flow in *Conv GT-Net*, containing T phases. The bottom part is the detailed structure of one phase. Gray arrows denote the data flow.

Comparison Algorithms

- TNN-ADMM

Novel methods for multilinear data completion and de-noising based on tensor-svd. In IEEE CVPR, pages 3842–3849, 2014.

- Imputation algorithm

Convolutional imputation of matrix networks. In ICML, volume 80, pages 4818–4827, 2018.

Experiment Settings

- Conv GT-Net is trained in TensorFlow on a server with an NVIDIA Telsa V100 GPU (16GB device memory)

- All algorithms are tested on a laptop with an i7-8750H CPU (2.20GHz) and 16GB memory.

- Data set: ego-Facebook data sets from SNAP with 4,039 nodes and 88,234 edges. We pick 100 nodes (No. 896-995) and their corresponding edges to form a graph topology.

Completion Accuracy vs. Missing Rate.

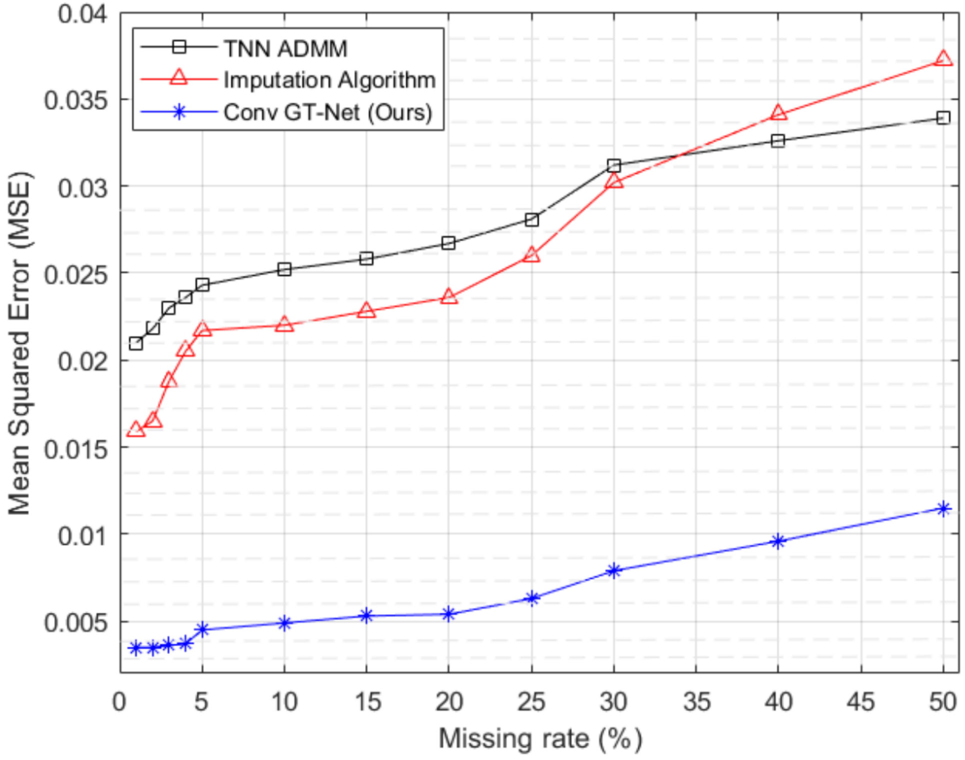

Figure 5: Experimental results of the weighted graph-tensor. Completion accuracy vs. missing rate.

Running Time vs. Missing Rate.

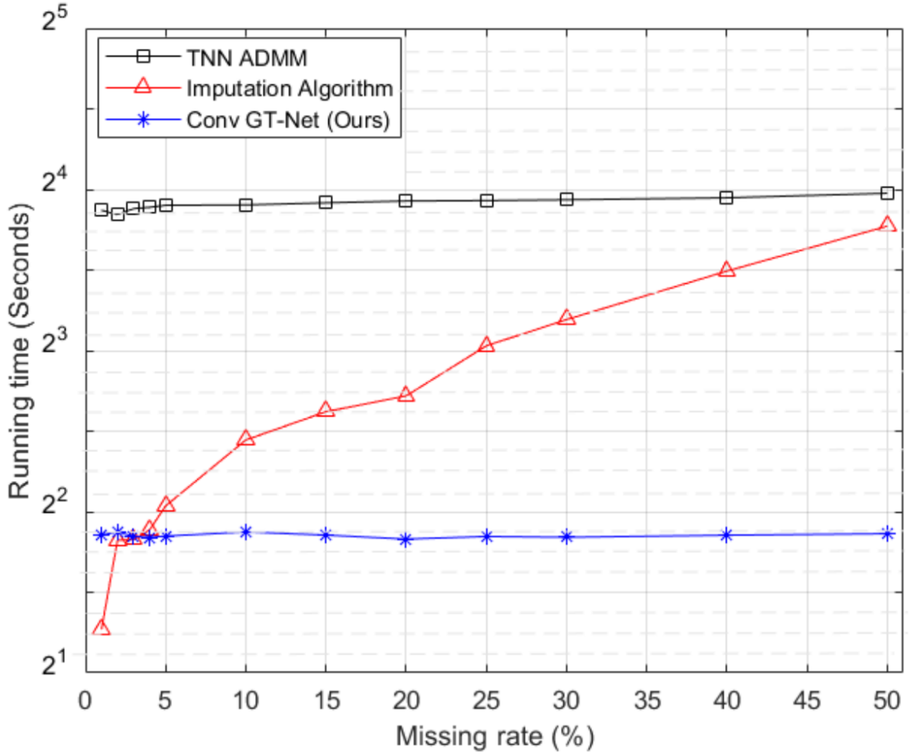

Figure 6: Experimental results of the weighted graph-tensor. Running time vs. missing rate.

Conclusion

- We propose Conv GT-Net to solve the graph-tensor completion problem using deep neural networks.

- Conv GT-Net provides higher completion accuracy and runs faster than the other algorithms.

End

Thank you