Temporal Graph Neural Networks for Irregular Data

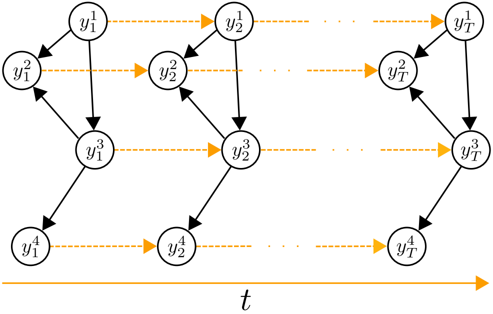
AISTATS 2023

Joel Oskarsson, Per Sidén, Fredrik Lindsten

Division of Statistics and Machine Learning
Department of Computer and Information Science
Linköping University, Sweden

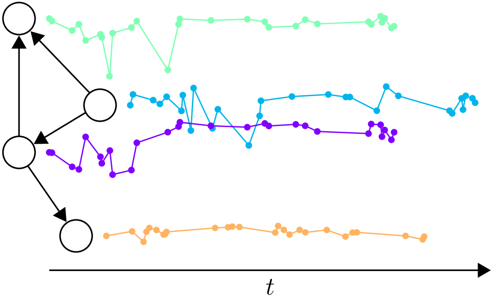
LINKÖPING
UNIVERSITY

Background: Temporal Graph Neural Networks



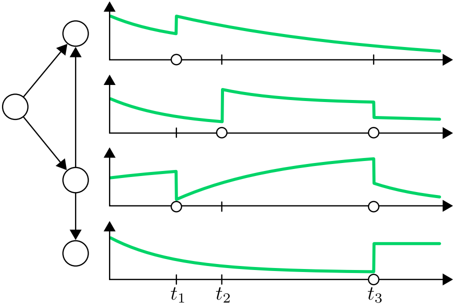
- Time series data with underlying graph structure
- Graph Neural Networks (GNNs) are deep learning models for graph-structured data
- Temporal GNNs include a time dimension

TGNN4I: Forecasting Irregular Graph-Structured Time Series



- One time series at each node
- Irregular observations
  - Irregular time steps
  - Observing subset of nodes
- Forecasting problems

Time-Continuous Latent States



- Time-continuous latent state in each node
- State defined by:
  - State dynamics in-between observations
  - State update when node is observed

Latent Dynamics: In-Between Observations

- State in node n

$$\boldsymbol{h}^n(t) = \bar{\boldsymbol{h}}_i^n + \tilde{\boldsymbol{h}}^n(t) \tag{1}$$

  - Static component $\bar{\boldsymbol{h}}_i^n$
  - Dynamic component $\tilde{\boldsymbol{h}}^n(t)$

Latent Dynamics: In-Between Observations

- State in node n

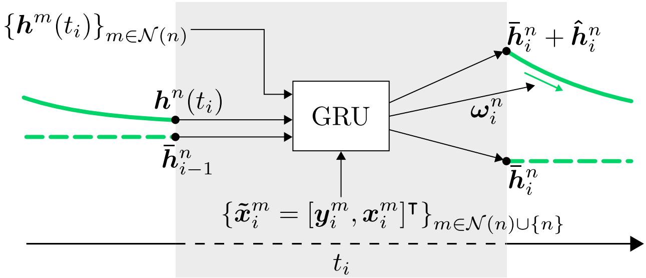
$$\boldsymbol{h}^n(t) = \bar{\boldsymbol{h}}_i^n + \tilde{\boldsymbol{h}}^n(t) \tag{1}$$

  - Static component $\bar{\boldsymbol{h}}_i^n$
  - Dynamic component $\tilde{\boldsymbol{h}}^n(t)$
- Define $\tilde{\boldsymbol{h}}^n(t)$ over $]t_i, t_j]$ as solution to the ODE

$$d\tilde{\boldsymbol{h}}^n(t) = A\tilde{\boldsymbol{h}}^n(t)\ dt \tag{2}$$

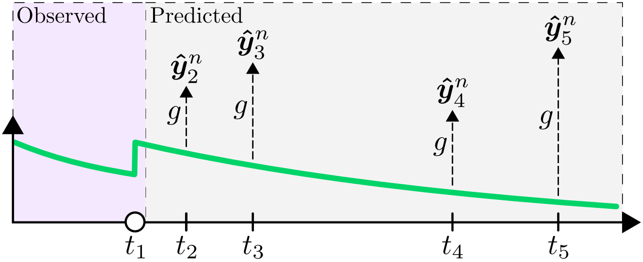
  with initial condition $\tilde{\boldsymbol{h}}^n(t_i) = \hat{\boldsymbol{h}}_i^n$.
- Closed form solution:
  - Exponential decay
  - Periodic dynamics

Latent Dynamics: Observation Update



- Gated Recurrent Unit (GRU) update
- GRU cell outputs
  - New state $\bar{\boldsymbol{h}}_i^n + \hat{\boldsymbol{h}}_i^n$
  - New static component $\bar{\boldsymbol{h}}_i^n$
  - Parameters $\boldsymbol{\omega}_i^n$ defining latent dynamics
- Extended with GNN layers

Making Predictions



- Predictive model g maps latent state $\rightarrow$ prediction

$$\hat{\boldsymbol{y}}_j^n = g(\boldsymbol{h}^n(t_j)) \tag{3}$$

  - GNN layers in g
- Predictions for arbitrary future time points!
  - Custom loss function

Experiments

Datasets

- Traffic (PEMS-BAY, METR-LA)
- Climate (USHCN)
- Synthetic periodic

Models

- TGNN4I with alternative dynamics
- Baseline models
  - All nodes (joint)ly
  - Independent (node)s

	PEMS-BAY (25% Obs.)	USHCN (T_{min})
Predict Previous	26.32	16.88
GRU-D (joint)	18.79	8.03
GRU-D (node)	8.79	13.12
Transformer (joint)	12.05	7.36
Transformer (node)	16.49	15.68
LG-ODE	27.00	-
TGNN4I (static)	7.41	6.97
TGNN4I (exponential)	**7.10**	**6.72**
TGNN4I (periodic)	**7.10**	**6.72**

Summary

- TGNN4I: A temporal GNN for forecasting irregular data
  - Time-continuous latent states to handle irregularity
  - GNN components to utilize graph structure
  - Predictions for arbitrary future time points
- Code available: `github.com/joeloskarsson/tgnn4i`

Joel Oskarsson
joel.oskarsson@liu.se[1]
joeloskarsson.github.io

Per Sidén
psiden@qti.qualcomm.com

Fredrik Lindsten
fredrik.lindsten@liu.se
lindsten.netlify.app

[1] Corresponding author