Hail risk prediction via Graph Neural Networks

A Preprint

Ivan Lukyanenko
Department of Control and Applied Mathematics
Moscow Institute of Physics and Technologies
Moscow
lukianenko.ia@phystech.edu

Abstract

Geo-spatial time series is an open area with great potential for theoretical and practical work. In particular, hail risk assessment is necessary to avoid the damage (agriculture, animal husbandry). The aim of the study is to build a model based on graph neural networks. Forecasting has been carried out in the short-term range based on the values of climate variables since 1991. The key features of the problem are: 1) rare events - over the past 30 years there have been less than 700 hail events throughout Russia, 2) the spatial structure of the series data. We are expecting to improve quality of solving such problems by combining methods from [1] and [2].

Keywords Hail risk prediction · GNN · spatial time series

1 Introduction

The main goal of this research is to introduce new GNN architecture by combining state-of-art results [1], [2] in the right way. Our object of research is geo-spatial time series. In common time series is a series of values of a quantity obtained at successive times with equal intervals between them. This research is about more extended concept: spatial time series. Spatial time series are almost the same as ordinary time series, but instead of values we observe some spatial objects. This work provides a solution to the hail forecasting problem. The hails are extreme events, because over the past 30 years there have been less than 700 hail events throughout Russia. The assessment of hail risk is necessary because of its environment damage. According to Verisk's 2021 report [5], in 2020 year losses due to hails reached $14.2 billion in the USA.

add citation

The producing architecture is based on works [1], [6] and [2]. [2] introduce **StrGNN** that solves the anomaly detection problem in dynamic graph, this architecture solves big class of problems well, but it should be adapted to narrower tasks. This work produces re-implemented **StrGNN** for our purpose in combination with GNN-network from [1]. [1] based on the model presented in [6]. We are going to use climate data from Google Earth Data, CMIP5, NOAA Storm Events Database, Severe Weather Dataset for training and evaluating our architecture.

2 Problem Statement

Using architecture is described in [2]. This is the parametric family of function to map design space to target space.

2.1 StrGNN

This section describes **StrGNN** model.
Given a temporal network $\{G(t) = \{V(t), E(t)\}_{t=1}^{n}\}$, where $G(t)$ is the graph snapshot at timestamp t consisting of verticles $V(t)$ and edges $E(t)$.
StrGNN works in three stages: ESG, GSFE, TDN.

ESG: Enclosing Subgraph Generation

The goal is to generate enclosing subgraph structure related to the target edge. Employing the whole graph for analysis is computational expensive. Recent work [7] proved that in Graph Neural Networks each node is most influenced by its neighbors. Enclosing subgraph in dynamic graphs definition according to [2] is following:

Definition 1. (Enclosing subgraph in dynamic graphs) For a temporal network $\{G(i) = \{V(i), E(i)\}\}_{i=t-w+1}^{t}$ with window size w, given a target edge e^t with source node x^t and destination node y^t, the h–hop enclosing subgraph $G^h_{x^t,y^t}$ centered on edge e^t is a collection of all subgraphs centered on e^t in the temporal network $\{G(i)^h_{x^t,y^t} | (t - w + 1) \leq i \leq t\}$. Enclosing subgraph contains only topographical information. In case of this, to distinguish the role of each node, nodes should be labeled. According to [2] good labeling contains the information of 1) target node in subgraph; 2) the contribution of each node in target one.
Suggested labeling is following:

$$f(i, x^t, y^t) = 1 + \min(d(i, x^t), d(i, y^t)) + (d_{\text{sum}}/2)[(d_{\text{sum}/2}) + (d_{\text{sum}}\%2) - 1] \tag{1}$$

where $d(i, x^t)$ is the shortest path distance between i and node x^t, $d_{\text{sum}} = d(i, x^t) + d(i, y^t)$.
If $d(i, x^t) = \infty$ or $d(i, y^t) = \infty$, node i is labeled with 0.

GSFE: Graph Structural Feature Extraction
Graph Convolution Neural Network from [4] map the subgraph space into embedding space.

TDN: Temporal Detection Network GSFE generates low-dimensional features, but it does not consider the temporal information to determine the class of the target.
Given the extracted feature matrices $\{H_i\}_{i=t-w}^{t} in \mathbb{R}^{K \times d}$, where K is the number of selected nodes in subgraph and d is dimension of feature for each node.
Employed the Gated Recurrent Units (GRUs) [3] to capture the temporal information as:

$$z_t = \sigma(W_z H_t + U_z h_{t-1} + b_z) \tag{2}$$

$$r_t = \sigma(W_r H_t + U_r h_{t-1} + b_r) \tag{3}$$

$$h_t^{'} = \text{tanh}(W_h H_t + U_h(r_t \circ h_{t-1}) + b_h) \tag{4}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ h_t^{'} \tag{5}$$

where $\circ$ represent the element-wise product, W, U and b are parameters. The GRU network is able to model the future temporal information.

2.2 Loss-function

The criterion is log-loss function. Hail Prediction is considering as classification problem. The output of the model is classified months in the prediction horizon whether hail will appear or not. The problem mathematically summarized as following:

$$w^* = \arg\min_{w} \Big(-\frac{1}{N} \sum_{t=1}^{N} (y^t \log(g(w, h^t)) + (1 - y^t) \log(1 - g(w, h^t))) \Big) \tag{6}$$

, where w^* is optimal vector of weights of our architecture; w is vector of weights of our architecture; t is timestamp from the beginning of the prediction horizon; N is number of timestamps in prediction horizon; y is target class of the timestamp; g is fully-connected network; h is output of re-implemented version of **StrGNN** [2]

References

[1] S. R. Cachay, E. Erickson, A. F. C. Bucker, E. Pokropek, W. Potosnak, S. Osei, and B. Lütjens. Graph neural networks for improved el niño forecasting. *CoRR*, abs/2012.01598, 2020.

[2] L. Cai, Z. Chen, C. Luo, J. Gui, J. Ni, D. Li, and H. Chen. Structural temporal graph neural networks for anomaly detection in dynamic graphs. *CoRR*, abs/2005.07427, 2020.

[3] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[4] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks, 2017.

[5] Verisk. The hail hazard and its impact on property insurance, 2020.

[6] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks, 2020.

[7] K. Xu, C. Li, Y. Tian, T. Sonobe, K. Kawarabayashi, and S. Jegelka. Representation learning on graphs with jumping knowledge networks. *CoRR*, abs/1806.03536, 2018.