SCode: Continuously Learning and Answering Predictive Event Queries over Graph Streams

Qu Liu **Emil Zulawnik** **Tingjian Ge**

Department of Computer Science, University of Massachusetts, Lowell
qliu@cs.uml.edu emil_zulawnik@student.uml.edu ge@cs.uml.edu

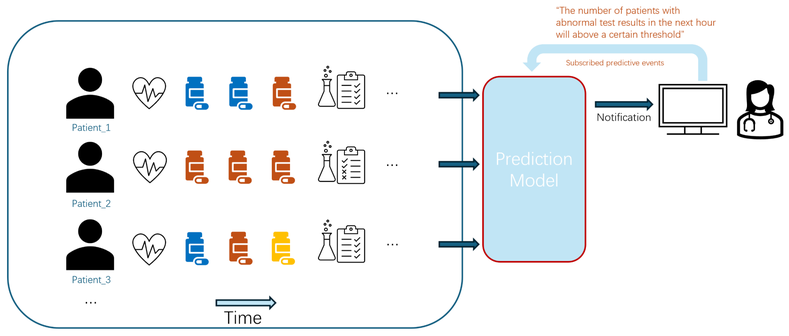
Motivation

Dynamic heterogeneous data from multiple data sources is very common in data science applications, a convenient way to model such data is graphs

- Highly dynamic graphs (graph streams) can be considered as an infinite sequence of graph snapshots
- For such dynamic and streaming data, it proves to be useful to continuously monitor a number of events that the users are interested in

Example

Consider a simple example of dynamic graphs in healthcare data (MIMIC). The graph nodes include patients, diagnosis, procedures, lab events, input events, and prescription drugs, while the edges are heterogeneous relations among the nodes.



- A hospital manager may monitor the predictive events such as "The number of patients with abnormal test results in the next hour will be above a certain threshold."

The Main Idea is continuously training data embeddings from a DGNN model and a code generator together called SCode:

- Map each of the result vectors to a codeword in a spherical code
- Get the data state embeddings in the same embedding space with margins to make sure SCode is error-correcting
- Use high-dimensional approximate nearest-neighbor (ANN) index to efficiently search for codewords
- Design a distributed-score-sharing (DSS) metric and use the Rademacher complexity theory to bound the deviation of DSS for incremental training on new data

Problem Formulation

- $\boldsymbol{G}$: A dynamic network $\boldsymbol{G=(N,E)}$ can be considered as an infinite sequence of snapshots $\boldsymbol{G_1, G_2, \dots, G_t, \dots}$, where each snapshot $\boldsymbol{G_t}$ corresponds to a time step $\boldsymbol{t}$
- $\boldsymbol{N}$: The set of nodes
- $\boldsymbol{E}$: The set of edges (either directed or undirected)
- $\boldsymbol{G_t}$: Each snapshot $\boldsymbol{G_t = (N_t, E_t)}$ satisfies $\boldsymbol{N_t \subseteq N}$ and $\boldsymbol{E_t \subseteq E}$
- v: Each node $v \in \boldsymbol{N}$ have a set of attributes $X = X_1, \dots, X_a$ that may bear different values in different snapshots
- e: Each edge $e \in \boldsymbol{E}$ may be one of r types

Preliminaries

Event prediction

It belongs to event analytics and focuses on anticipating events in the future. Compares to traditional application domains, there is a more temporal and real-time emphasis in dynamic graphs.

Complex Event Processing (CEP) and Data Stream Mining

In CEP, users subscribe to a set of events that they are interested in, and the system will need to continuously monitor the dynamic data stream and obtain the results of the events. The users are notified when the results of the events that they subscribed to change.

Graph Machine Learning

Due to the importance of data modeled as graphs, especially for heterogeneous data from multiple sources with complex connections, graph machine learning has been well developed over the years. The state-of-the-art representation learning over graphs is based on graph neural networks (GNN), and dynamic graph neural network (DGNN) models have been developed for dynamic graphs.

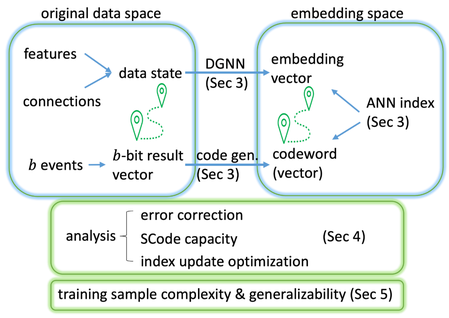
Online Learning and Change Detection

Online learning updates models from data streams sequentially and change detection has been studied for data streams

Challenges

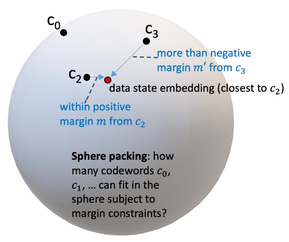
- Complex dependencies among multiple predicted events—the predicted events can correlate to and influence each other
- Real-time stream of prediction tasks—it usually requires continuous monitoring the dynamic data to trigger timely alerts of future potential events
- The trained model gradually becomes outdated when real world events continually change dynamically, concepts are fluid and distribution drifts are inevitable

Solution

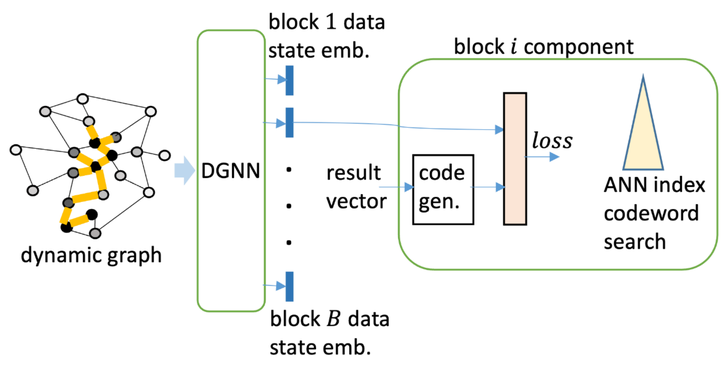


- **Reduce the problem of monitoring multiple predictive events to multi-label classification (MLC)**
- **Simultaneously performs online incremental learning and continuous prediction of a given set of events**

Multi-label classification (MLC)



- *Suppose there are b predictive events with binary labels, the b events together have 2^b possible **result vector values***
- *Map each of the result vectors to a **codeword** in a **spherical code** through a neural network model called code generator*



Data State Embedding and Partitioning Events into Blocks

- *Can employ any dynamic **graph neural network (DGNN)** model in previous work to produces node embeddings*
- **Randomly** partition all the user defined predictive events into B blocks, each of which has b events, **index and search** each block separately

Code Generator

- *For each block, a multi-layer perceptron (MLP) taking one of the 2^b result vectors as input and outputting the corresponding **codeword***

Model Training

- *The parameter update of DGNN and that of the code generator (of a block) are performed **separately** during training*
- The training is based on a loss function that intuitively characterizes the **difference** between the ***data state embedding*** and the ***codeword*** corresponding to the correct result vector of the b events

Prediction of Event Outcomes

- Obtains the **data state embedding** $\boldsymbol{d}$ by performing a forward propagation over the DGNN
- SCode uses an **ANN index** to efficiently search for the ***codeword*** c_r that is closest to the data state embedding $\boldsymbol{d}$
- The **result vector** r corresponding to the ***codeword*** c_r will be the predicted outcomes of the b events

Evaluation

Datasets:

(1) **Taxi Dataset**. This dataset contains the information of all taxi trips in the New York City in 2013. It has 14 attributes, including medallion, hack license, vendor ID, pick-up date/time, drop-off date/time, pick-up longitude/latitude, drop-off longitude/latitude, trip time, and trip distance.

(2) **Bike Dataset**. This is bike sharing data in Metro Boston. It has approximately 4.52 million bike trips with 18 attributes including trip duration, start/stop time, start/end station latitude/longitude, bike id, user type, birth year, gender, etc.

(3) **MIMIC-III Dataset**. is a large, freely-available database comprising deidentified health data associated with over 40,000 patients. It includes high temporal resolution data such as vital sign measurements, laboratory test results, procedures, medications, caregiver notes, imaging reports, demographics, etc.

(4) **UCI Messages Dataset**. This is a temporal social-network dataset that consists of private messages sent on an online social network system among students, where nodes are users and edges are messages.

Results: