# *Thesis Proposal*
# Mining Anomalies using Static and Dynamic Graphs

Dhivya Eswaran

5 November, 2019

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

**Thesis Committee:**
Christos Faloutsos (Chair)
Aarti Singh
Zico Kolter
Nina Mishra (Amazon)

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Copyright © 2019 Dhivya Eswaran

# Abstract

Data generated in a multitude of diverse contexts today have relational and temporal characteristics, with multiple entities interacting with each other and also evolving over time. Examples range from e-commerce logs to online social networks to the internet-of-things. Our thesis addresses the problem of detecting and predicting anomalies–deviations from usual patterns–in such settings. Anomalies often encode suspicious, fraudulent or malicious behavior. They do not just influence users into making sub-optimal decisions but also steadily erode their trust in businesses. As such, algorithms to detect ongoing anomalies and warn against upcoming anomalies have high impact for businesses and end-users alike.

In the first part of the thesis, we focus on the case where only static connectivity information is known, and the goal is to infer labels for vertices, e.g., whether a user account is honest or fraudulent, from limited labeled data. Our completed work broadens the scope of literature by handling heterogeneous graphs, and leveraging label uncertainty for more accurate vertex labeling.

In the second part of the thesis, we mine anomalies from data where the connectivity evolves over time. Our primary focus here is on real-time detection and early warning so as to enable timely corrective or preventive measures against anomalies. Our completed work can detect anomalous dense subgraphs and edges in near real-time, by only storing a small synopsis of the graph seen so far and requiring no supervision. We also show how to early warn against user-labeled anomalies in the presence of confounding interventions.

As part of ongoing and future work, we will continue to push on both fronts by investigating the importance of higher-order structures for vertex labeling and characterizing the anomalousness of any given graph substructure or motif.

# Chapter 1

# Introduction

Large-scale data mining has become a focal point of research in computer sciences and social sciences in recent years. Statistics[1] show that over 2.5 quintillion bytes of new data is generated worldwide every day from commercial transactions, social networks, system log data, electronic sensors and more. Much of this data has strong relational and temporal characteristics, capturing multiple entities interacting with each other and also evolving over time. Thus, it can be naturally modeled as a graph. Our thesis provides effective and scalable algorithms to analyze and garner insights from graph data, and specifically, detect anomalies or deviations from typical patterns.

## 1.1 Problem

Anomaly detection is a pressing problem for various critical tasks such as security, finance and the web. Anomalies–such as review or rating fraud–encode suspicious, fraudulent or malicious behavior and do not just influence people into making sub-optimal decisions but also steadily erode their trust in businesses. As such, algorithms to detect ongoing anomalies and warn against upcoming anomalies have high impact for businesses and end-users alike.

An immediate challenge in detecting anomalies lies in defining what anomalies or outliers are. One of the earliest definitions dates back to 1980, when Hawkins [29] observes: "*An outlier is an observation that differs so much from other observations as to arouse suspicion that it was generated by a different mechanism*". The decided vagueness of this definition makes anomaly mining a challenging and open-ended problem. A more useful and meaningful definition of anomaly is possible only under a given context or application. Anomalies in our work are motivated by online social networks, e-commerce, communication, transportation and the internet-of-things, to name a few.

## 1.2 Organization

This thesis is organized in two parts. In the first part, we focus on the case where only static connectivity information is known, and the goal is to infer a particular discrete characteristic of

[1] https://leftronic.com/big-data-statistics/

vertices, e.g., whether a user is honest or fraudulent, when given access to limited labeled data. In the second part, we mine anomalies from data where the connectivity evolves over time. Our primary focus here is on real-time detection and early warning so as to enable timely corrective or preventive measures against anomalies.

## 1.3 Completed Work

We elaborate on our completed work below.

**Static Graphs.** Our completed work in this part focuses on the core sub-problem of semi-supervised learning that arises in anomaly detection when limited labels are available. Here, broaden the scope of prior literature, specifically, the seminal belief propagation (BP) algorithm [70], in two ways. First, in §2.1, we develop a fast linearizing approximation for BP called ZooBP [17] which can handle large heterogeneous graphs with multiple vertex and edge types. Next, in §2.2 [16], we develop a variant of BP which incorporates the notion of label uncertainty or confidence to more accurately label the vertices. Both algorithms are fast and highly scalable, running $2-3\times$ faster than BP.

**Dynamic Graphs.** Our completed work in this part focuses on near real-time detection and early warning of anomalies. In §3.1 [18], we propose a randomized sketching-based approach to detect anomalous dense subgraphs in near real-time by only storing a small synopsis of the graph seen so far. Our work in §3.2 [15] considers a similar setting, but detects anomalous edges using a novel sampling technique. In §3.3 [19], we show how to learn an interpretable graph representation from time-series data and use it to early warn against user-input anomalies in the presence of interventions.

## 1.4 Ongoing and Proposed Work

We summarize our ongoing and proposed work below.

**Static Graphs.** In §2.3, we will quantify the extent to which the higher-order structures are homogeneous in labels, and explore their usefulness in vertex labeling. Higher-order structures is a signal that the present day techniques do not exploit for semi-supervised vertex labeling.

**Dynamic Graphs.** In §3.4, we will investigate how to characterize the anomalousness of specific graph substructures and detect them in near real-time.

## 1.5 Overview

Table 1.1 gives a high-level overview of both completed, as well as ongoing and proposed work described in this proposal, with sub-tasks categorized into the associated tasks. Associated section numbers, references, and PDF links are given in the table for the reader's convenience. An asterisk besides a sub-task indicates that the associated work is ongoing or proposed, whereas a lack thereof indicates completion.

| | |
|---|---|
| **Static Graphs (§2)** | **[S1]** Heterogeneous Graphs (§2.1) [17] [PDF]<br>**[S2]** Incorporating Confidence (§2.2) [16] [PDF]<br>**[S3$^*$]** Leveraging Higher-Order Structures (§2.3) |
| **Dynamic Graphs (§3)** | **[D1]** Anomalous Dense-Subgraph Detection (§3.1) [18] [PDF]<br>**[D2]** Anomalous Edge Detection (§3.2) [15] [PDF]<br>**[D3]** Early Warning of User-Input Anomalies (§3.3) [19] [PDF]<br>**[D4$^*$]** Anomalous Motif Detection (§3.4) |

**Table 1.1: Overview of completed, ongoing and proposed work.**

The following two chapters (§2, §3) describe our work corresponding to the two aforementioned tasks. For each completed sub-task, we give a high-level summary, and introduce the main ideas and results. For each proposed or ongoing sub-task, we motivate the problem's high level research questions as well as discuss preliminary results or potential directions for work.

# Chapter 2

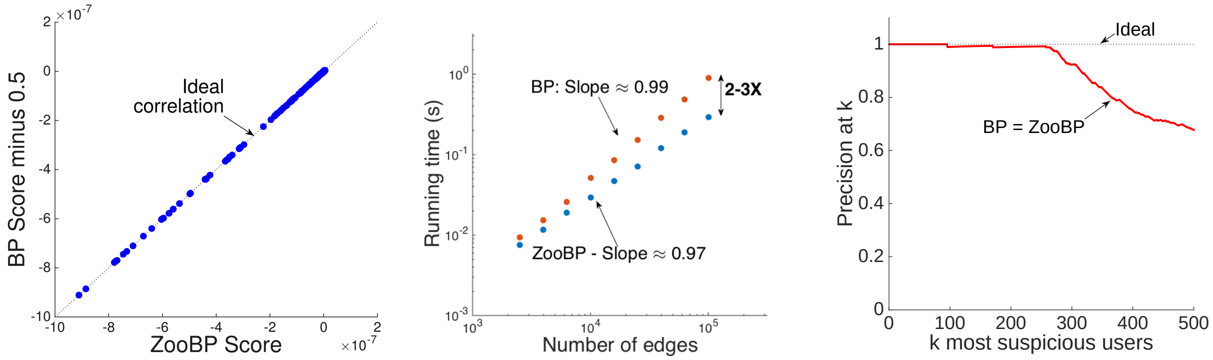
# Static Graphs

“Given a large static graph and labels for a few vertices, how can we infer the most likely labels for all remaining vertices?” This is the classic graph transductive learning or semi-supervised learning problem (SSL). Graph SSL is highly useful for fraud detection, when a few manually labeled fraudsters are given. For example, in a user×product bipartite rating or review network, using a few manually identified fraudulent user accounts, we can identify other suspicious accounts showing similar characteristics. While our work is motivated primarily by anomaly detection, it is important to note that graph SSL is general problem setting which has applications well beyond those that we consider, in recommendation and bioinformatics [63]. Our completed and ongoing work on this problem is described below.

## 2.1 Heterogeneous Graphs

Section based on work that appeared in VLDB 2017 [17] [PDF].

**Goal:** “Given a heterogeneous network, with vertices of different types – e.g., products, users and sellers from an online recommendation site like Amazon – and labels for a few vertices



**(a) Quality of approximation** **(b) Speedup** **(c) Effectiveness**

**Figure 2.1: ZooBP: Heterogeneous Graphs**

('honest', 'suspicious', etc), can we find a closed formula for Belief Propagation (BP), exact or approximate? Can we say whether it will converge?" BP, traditionally an inference algorithm for graphical models, exploits so-called "network effects" to perform graph classification tasks when labels for a subset of vertices are provided; and it has been successful in numerous settings like fraudulent entity detection in online retailers and classification in social networks. However, it does not have a closed-form nor does it provide convergence guarantees in general – leading to non-exact solutions on graphs which contain loops. Our goal in this work to derive a fast and accurate approximation of belief propagation for heterogeneous graphs, as stated below.

**Problem 2.1.1** (Fast Heterogeneous Belief Propagation)**.**

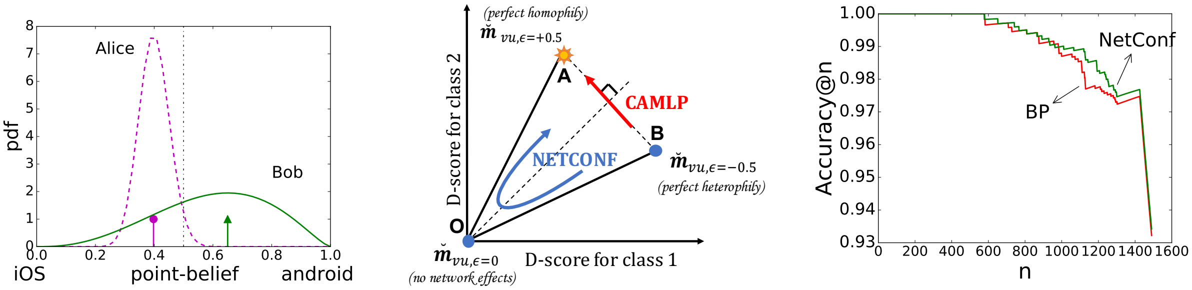
- ***Given*** *a large heterogeneous graph, a set of labels for each vertex-type, compatibility matrices for each edge-type indicating the affinity between vertex labels, and initial beliefs about the label of a vertex for a few vertices in the graph*
- ***find*** *the most probable class (label) for each vertex given by belief propagation*
- ***by identifying*** *conditions which allow accurate approximation and* ***deriving*** *a closed-form solution for the final beliefs.*

**Approach:** The core idea of ZOOBP is to derive a system of linear equations for beliefs which can be solved using matrix algebra to finally calculate all vertex beliefs in a single step of matrix operations. To do this, ZOOBP uses two insights: (i) to focus on a small set of compatibility matrices which are *constant-margin*, i.e., where all rows and columns sum up to the same value; (ii) to consider very low interaction strengths $\epsilon$ that allow only weak messages (of small magnitude) to propagate through the edges. As we show, despite these assumptions, ZOOBP still applies to a variety of real-world settings and can lead to accurate vertex labeling results. Together, these assumptions allow us to linearize the original iterative updates of BP due to [70] by approximating all beliefs and messages around their 'centered values', i.e., around $1/k$ for a $k$-class problem. Further, to handle the heterogeneity of vertices and edges in the graph, we carefully derive a persona-influence matrix $\mathbf{P}$ and an echo-cancellation matrix $\mathbf{Q}$ (see Definitions 9 and 10 in [17]). These can be put together to yield the following linearized approximation to compute final beliefs $\mathbf{b}$ from prior beliefs $\mathbf{e}$:

$$\mathbf{b} = \mathbf{e} + (\mathbf{P} - \mathbf{Q})\mathbf{b} \tag{2.1}$$

**Results:** Our main theoretical results are to show that (a) a closed-form solution for Equation (2.1) can be derived using Kronecker matrix products and Roth's column lemma [30] as key ingredients and (b) to further use results from sparse linear systems [52] to derive the exact convergence guarantee for the iterative update in Equation (2.1).

Through experiments, we first demonstrate the quality of approximation offered by ZOOBP. A plot of the final beliefs (minus 0.5, the 'centered value') returned by BP and ZOOBP on a real-world dataset is displayed in Figure 2.1(a). We see that all points lie on a line of slope 1 passing through the origin, showing that ZOOBP beliefs are highly correlated with BP beliefs. From Figure 2.1(b), we observe that ZOOBP scales linearly with the number of edges in the graph (i.e., graph size), which is same as the scalability of BP. More importantly, ZOOBP leads to $2-3\times$ speedup on C++ as it replaces the expensive logarithmic and exponentiation operations of



**(a) Motivation** **(b) Modulation of messages** **(c) Effectiveness**

**Figure 2.2: NetConf: Incorporating Confidence**

BP with light-weight additions and multiplications. We also applied ZooBP on real-world data from Flipkart e-commerce network to classify users, products and sellers as honest or dishonest by using 50 manually labeled dishonest users. We supplied the top 500 most suspicious users to domain experts from Flipkart who verified our labels by studying various aspects of user behavior such as frequency and distribution of ratings and review text. The resulting precision curve, depicted in Figure 2.1(c) shows that ZooBP achieves a high precision of nearly 100% over the top 250 and around 70% over the top 500 suspicious users. Recall studies were not possible owing to difficulty in obtaining exhaustive ground truth on all fraudulent users in the data.

## 2.2 Incorporating Confidence

Section based on work that appeared in SDM 2017 [16] [PDF].

**Goal:** "Given a friendship network, how certain are we that a vertex has a particular label? How can we propagate these certainties through the network?" Traditional semi-supervised methods which propagate labels or beliefs suffer from a major limitation that they do not take uncertainty in the labels or beliefs into account. Consequently, while propagating information, these methods treat vertices with certain and uncertain beliefs with equal weight, resulting in counter-intuitive responses. An example is shown in Figure 2.2(a) where Smith has to choose between iOS and android phones based on inputs from his friends Alice and Bob. Even though Alice is a stubborn tech-geek who favors iOS over android, uncertainty-unaware methods incorrectly recommend android to Smith. Our goal in this work is stated in Problem 2.2.1.

**Problem 2.2.1** (Vertex Classification with Certainty)**.**

- ***Given*** *a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, labels $l_v \in \{1, 2, \ldots, k\}$ for a subset of the vertices $v \in \mathcal{V}$ (with their uncertainties) and the nature of network effects (e.g., homophily),*
- ***find*** *the probability (belief/leaning) $b_u(i)$ that vertex $u$ has label $i$ along with a measure of certainty.*

**Approach:** The main idea behind NetConf is to models beliefs as Dirichlet distributions to capture uncertainty and use multinomial counts as messages to propagate these uncertainties along the edges of the network. Specifically, the messages that a vertex sends to its neighbors is obtained via a modulation process depicted in Figure 2.2(b). Here, A represents the parameters of belief distribution for a vertex, which is also the message it broadcasts when there is perfect homophily in the network, i.e., when adjacent vertices have the same beliefs. When there is perfect heterophily, i.e., when adjacent vertices have the opposite beliefs, the message sent is point B which is the mirror image of A about the $x = y$ line. When there are no network effects, neighbors do not influence each other and hence the messages sent are identically zero. For intermediate values of network effects, a linear interpolation AOB is used for modulating messages. All in all, this process ensures that the strength of messages sent by a vertex is limited by its strength of its belief as well as that of network effects in the graph. The computation of messages from beliefs and the update of belief distributions using messages can be expressed as linear updates. Using this, we can derive a closed-form matrix solution and convergence guarantees for NETCONF.

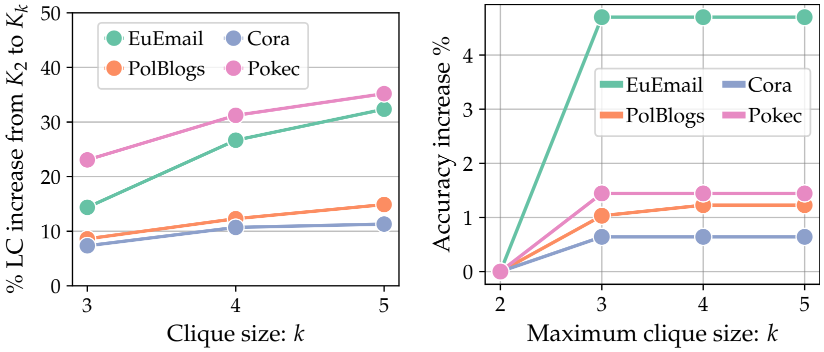
**Results:** Experiments show that NETCONF matches or outperforms uncertainty-unaware belief propagation across three real-world datasets. When vertices are sorted in increasing order of their uncertainty scores such that the more confidently classified vertices are ranked first and accuracy is measured over the top $n$ vertices (see Figure 2.2(c)), NETCONF achieves higher accuracy@$n$ than the baseline, suggesting that NETCONF is better-suited to precision-critical applications such as fraud detection. Moreover, we discovered that most confident classified vertices typically tended to have high degrees suggesting that the more incoming messages lead to more confident vertex labeling. In the case of DBLP coauthorship data, where the task is the classify authors into areas of their research (databases, data mining, artificial intelligence and information retrieval), the most confidently classified authors were the ones who had published a lot of papers and hence had many co-authors.

## 2.3 [Ongoing] Leveraging Higher-Order Structures

Section based on work under submission to WWW 2020.

**Goal:** “Do higher-order network structures aid graph semi-supervised learning?” Traditional graph SSL algorithms tend to be limited by the fact that all the neighbors of a vertex are not equal. A typical user in a friendship network has many acquaintances, but only a few close friends who belong to a small tightly-knit circle. In fact, prior research has shown that vertices with a strong connection participate in several higher-order structures, such as dense subgraphs and cliques [28, 32, 33, 56]. Thus, we hypothesize that leveraging the higher-order structure between vertices is crucial to accurately label the vertices. Our goal in this work is to answer the following research questions:

- **[RQ1]** Are higher-order network structures more homophilic in labels compared to edges?
- **[RQ2]** How can we leverage higher-order structures for graph SSL in a principled manner?
- **[RQ3]** Do higher-order structures help improve graph SSL?



**(a) Hypothesis verification** **(b) Effectiveness**

**Figure 2.3: HOLS: Leveraging Higher-Order Structures for Label Spreading**

**Approach:** To answer **[RQ1]**, we propose a novel information-theoretic metric called *label consistency* (LC) to measure the homogeneity of labels within higher-order structures in a network. LC is an entropy score normalized by the baseline distribution of expected homogeneity of labels in similar graphs. As such, LC is invariant to the size of the higher-order structure, the number of instances of the structure in the graph, and the marginal distribution of the labels in the graph, thus enabling a fair comparison of label homogeneity across structures and graphs of varying sizes and distributions. To answer **[RQ2]**, we create an algorithm, HOLS, for label spreading using higher-order structures. HOLS works for any user-inputted higher-order structure(s) and in the base case, is equivalent to the standard edge-based label spreading [73]. Furthermore, using the equivalence between HOLS and LS on a modified graph, we show that HOLS has a closed-form matrix solution and strong theoretical guarantees.

**Preliminary Results:** Applying the proposed label consistency metric to real-world data, (see Figure 2.3(a)), we find that label consistency increases monotonically with $k$, although with diminishing returns for increasing $k$. Overall, larger cliques are $7-35\%$ more label-consistent than edges, suggesting that the use of higher-order cliques can help in effective labeling of vertices.

Figure 2.3(b) shows the effect of adding higher-order structures to HOLS. We observe that label spreading via higher-order structures strictly outperforms label spreading via edges and the gain is the most when using 3-cliques (triangles). We also compared HOLS to state-of-the-art deep learning baselines based on skipgram embeddings and convolutional neural networks and observed that HOLS is over $5\times$ faster than these approaches for comparable and often better values of accuracy. Notably, HOLS is practically useful on large networks, with a total run time of under 2 minutes on networks with over 22 million edges.

**Plan:** As future work, we plan to understand the benefits of incorporating higher-order structures in the theoretical networks. For instance, what is the expected lift in using higher-order structures in Erdos-Renyi graphs? In stochastic block models, how does the behavior vary with intra-group and inter-group edge probabilities?

# Chapter 3

# Dynamic Graphs

We present our completed work on mining anomalies in dynamic graphs. For each completed task, we briefly summarize our problem, main ideas, and results. References for detailed discussions are given at the beginning of each section.

## 3.1 Anomalous Dense-Subgraph Detection

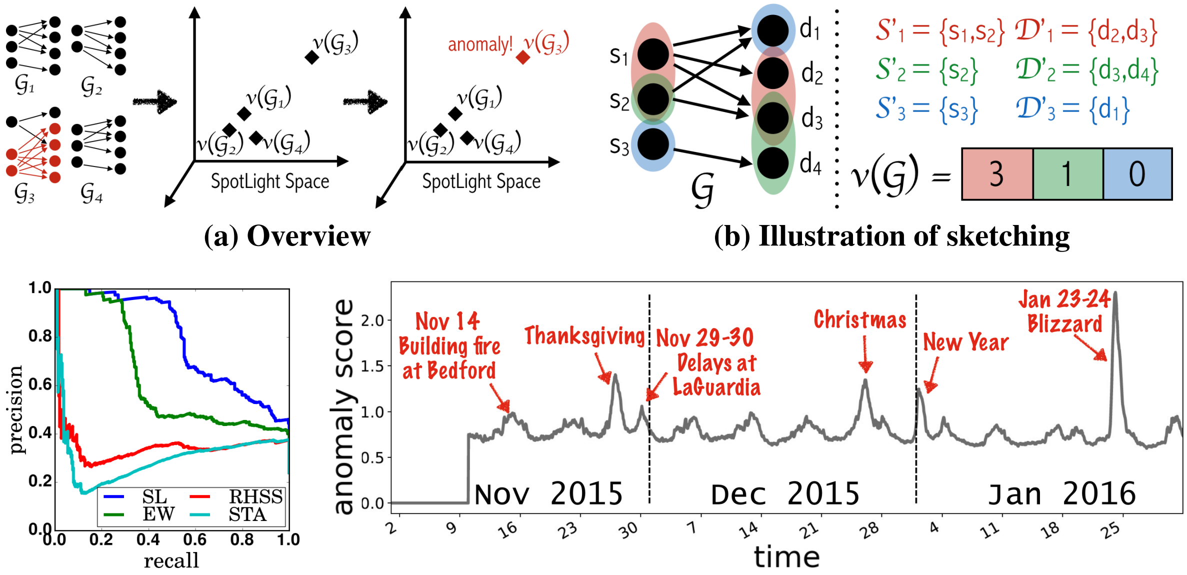
Section based on work that appeared in KDD 2018 [18] [PDF].

**Goal:** "Given a sequence of weighted, directed or bipartite graphs, each summarizing a snapshot of activity in a time window, how can we spot anomalous graphs containing the sudden appearance or disappearance of large dense subgraphs (e.g., near bicliques) in near real-time using sublinear memory?" This problem has several important applications: detecting attacks such as port scan and denial of service in network communication logs, identifying interesting or fraudulent behavior which create spikes of activity in user-user communication logs (e.g., scammers who operate fast and in bulk), discerning important events such as holidays or large delays which create abnormal traffic in/out flow to certain locations, to name a few. As formalized in Problem 3.1.1, our goal in these settings is to devise an algorithm which can detect dense subgraph anomalies in near real-time using limited memory and has provable guarantees for detection.

**Problem 3.1.1** (Anomalous Dense-Subgraph Detection in Graph Streams)**.**

- ***Given*** *a stream of weighted, directed or bipartite graphs,* $\{\mathcal{G}_1, \mathcal{G}_2, \ldots\}$*,*
- ***detect in near real-time*** *whether* $\mathcal{G}_t$ *contains a sudden (dis)appearance of a large dense directed subgraph*
- ***using sublinear memory*** *in the number of vertices in the graph.*

**Approach:** We propose a sketching-based algorithm called SPOTLIGHT which works in two main steps as shown in Figure 3.1(a). For each graph $\mathcal{G}$ in the stream, SPOTLIGHT first extracts a $K$-dimensional sketch $v(\mathcal{G})$, and then compares the extracted sketch to that from previous graphs in the stream to detect anomalies using distances in the sketch space. Crucially, comparing distances in the sketch space suffices because SPOTLIGHT guarantees that graphs containing the



**(a) Overview** **(b) Illustration of sketching**

**(c) Effectiveness** **(d) Discoveries on real-world New York City taxicab data**

**Figure 3.1: SPOTLIGHT: Anomalous Dense-Subgraph Detection in Graph Streams**

sudden (dis)appearance of large dense subgraphs are 'far' from 'normal' graphs in the sketch space (see Theorem 3.1.1).

As shown in Figure 3.1(b) for $K = 3$, each dimension in the sketch captures the total weight of edges connecting a set of randomly chosen source vertices $\mathcal{S}_i$ to a set of randomly chosen destination vertices $\mathcal{D}_i$. Here $\mathcal{S}_i$ and $\mathcal{D}_i$ are subsets of vertices chosen by sampling every source vertex (first column in Figure 3.1(b)) with a probability $p$ and every destination vertex (second column in Figure 3.1(b)) with a probability $q$. This choice is made only once per source or destination vertex (the first time it is seen) and is fixed throughout the graph stream. In practice, the mapping of vertices to these sets are maintained using hash functions, allowing the algorithm to process old and new vertices alike. When each sketch dimension is viewed as a spotlight which illuminates and allows for monitoring a region of the graph, the central idea of the algorithm is that the (dis)appearance of a large and dense subgraph would be brought to light by at least one of these spotlights, provided there are enough of them and each one is fine-grained, illuminating a small enough region of the graph.

**Results:** The main distance guarantee offered by SPOTLIGHT in the sketch-space is stated in Theorem 3.1.1. In the following, let $\mathcal{G}$ be the graph of interest and $\bar{d}$ be a notion of distance in the sketch-space (Definition 1 in [18]). Suppose $\mathcal{F}_{ER}(\mathcal{G}, n^2)$ is the family of 'normal' graphs obtained by corrupting $\mathcal{G}$ by adding $n^2$ edges uniformly at random, whereas the 'anomalous' graph $\mathcal{G}_{QB}$ is obtained by adding the same number of edges in a small $n \times n$ region of the graph. Then, Theorem 3.1.1 asserts that it is possible to achieve a significant distance separation between normal and anomalous graph instances in the sketch-space with high probability. Let vertex sampling probabilities $p = q < 0.5$ and $1 \ll n^2 \ll N^2$ where $N$ is the total number of vertices and let $K^* = (1 + p^2n^2)^2/4p^2n^2\delta \; + \; \epsilon/p^3n^3$.

**Theorem 3.1.1** (Anomaly Detection Criterion, restated from Theorem 5.3 in [18]). *A normal* *graph $\mathcal{G}_{ER}$ drawn from $\mathcal{F}_{ER}(\mathcal{G}, n^2)$ is situated at a distance at most $\bar{d}(\mathcal{G}, \mathcal{G}_{QB}) - \epsilon$ from $\mathcal{G}$ with high probability $1 - \delta$, provided the number of sketching dimensions is greater than $K^*$. Here, $\bar{d}(\mathcal{G}, \mathcal{G}_{QB})$ is the distance of the anomalous graph $\mathcal{G}_{QB}$ from $\mathcal{G}$ in the sketch-space and the probability is taken over the edge corruption process in $\mathcal{F}_{ER}$. That is,*

$$K > K^* \implies \mathbf{Pr}_{\mathcal{G}_{ER} \sim \mathcal{F}_{ER}(\mathcal{G}, n^2)} \left[ \bar{d}(\mathcal{G}, \mathcal{G}_{QB}) - \bar{d}(\mathcal{G}, \mathcal{G}_{ER}) \geq \epsilon \right] \geq 1 - \delta \tag{3.1}$$

When applied on real-world network traffic data with manually annotated attacks as anomalies, SPOTLIGHT outperforms prior methods by achieving a (statistically significant) higher precision for every recall threshold, as shown in Figure 3.1(c). Further, when applied to real-world New York City taxicab data from Nov 2015 to Jan 2016 where the anomalies are a priori unknown, SPOTLIGHT makes several unexpected but meaningful discoveries. The most anomalous period (Jan 23-24) coincided with the January 2016 US blizzard which rendered normal traffic operation impossible. The next three anomalies corresponded to festival periods–Thanksgiving, Christmas, New Year–presumably due to unusual traffic patterns around Manhattan (closed offices, Macy's Thanksgiving parade, New Year parties) and airports (people flying in and out of JFK and LaGuardia). The next two anomalies (Nov 14, Nov 29-30) are more interesting because they disrupt traffic operations only a small localized regions in New York City and are as such easily missed by prior methods. As we later discovered from archived news articles, these anomalies were the result of a huge fire which ripped through Bedford-Stuyvesant building (Nov 14) and thousands of people who were delayed at LaGuardia airport following the Thanksgiving weekend, resulting in over an hour-long wait times for taxis.

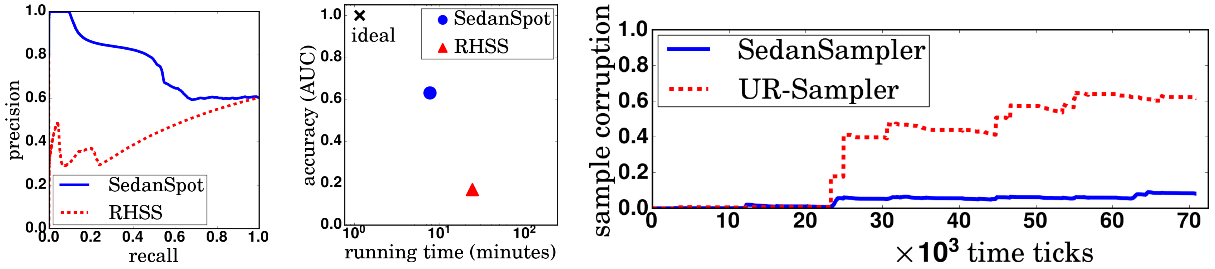
## 3.2 Anomalous Edge Detection

Section based on work that appeared in ICDM 2018 [15] [PDF].

**Goal:** "Given a stream of edges from a time-evolving (un)weighted (un)directed graph, how can we detect anomalous edges in near real-time using sublinear memory?" The goal here is to detect whether an edge is anomalous or not *immediately* after it appears, unlike SPOTLIGHT which waits for all edges in a single graph snapshot to arrive before flagging anomalies. Naturally, the requirement of per-edge decision limits the anomalies that can be detected, e.g., dense subgraphs can be spotted more easily at the graph level than on a per-edge basis. However, the advantage of such an edge streaming model is that the flagged anomalies can be used right away to curtail the impact of malicious activities and kick-start recovery processes in a timely-manner, e.g., terminating a scam phone call *when it is still ongoing*. The overall problem we aim to solve is:

**Problem 3.2.1** (Streaming Anomalous Edge Detection)**.**

- ***Given*** *an edge stream $\mathfrak{E}$=$\{e_1, e_2, \ldots\}$ from a/an (un)weighted (un)directed graph where each edge $e_i = (u_i, v_i, w_i, t_i)$ is a 4-tuple of source vertex, destination vertex, edge weight and time of its occurrence,*
- ***detect*** *whether $e_i$ is anomalous*
- *in* ***near real-time*** *using* ***sublinear memory*** *in the number of vertices in the graph.*



**(a) Effectiveness** **(b) Efficiency** **(c) Low corruption due to 'rate-adjustment'**

**Figure 3.2: SedanSpot: Streaming Anomalous Edge Detection**

**Approach:** We propose a sampling-based algorithm called SedanSpot (short for Streaming EDge ANomaly SPOTter) which exploits two tell-tale temporal and spatial signs of anomalous edges, i.e., they tend to (i) occur as bursts of activity and (ii) connect parts of the graph which are sparsely connected. To do this quickly using bounded memory, we maintain a fixed-size sample of the edges seen thus far and use it to score the anomalousness of any new edge. For the first component, we propose SedanSampler which uses a novel *rate-adjusted sampling strategy* to provably downsample bursty (and likely anomalous) edges. This ensures that the maintained sample of edges more accurately reflects normal behavior and thus sets stage for better anomaly detection. For the second component, SedanScorer scores the anomalous of an edge $(u, v)$ based on the *marginal proximity increase* between $u$ and $v$ upon adding the edge to the sample. Hence, an edge which connects sparse regions–and hence brings its incident pair of vertices significantly closer upon its addition to the sample–is given a higher anomalous score as desired. We use the relevance score provided by random walk with restarts [62] as the measure of vertex proximity as it is principled (incorporating direct and indirect paths), asymmetric, bounded in $[0, 1]$ and can be estimated fast using local random walks.

**Results:** The main sampling guarantee offered by SedanSpot is stated in Theorem 3.2.1. Suppose a time tick $\tau$ is said to be anchored if some edge occurred at time $\tau$. Then, SedanSpot ensures that the number of sampled edges belonging to a given time interval only depends on its duration and not on the number of edges occurring during it.

**Theorem 3.2.1** (Burst Resistance)**.** *Consider time ticks $\tau_0{=}0$ and $\tau_1 \leq \tau_2 \ldots {\leq}\tau_K$ which are anchored. Let $\mathcal{H}_k$ be the set of edges arriving in time interval $I_k := (\tau_{k-1}, \tau_k]$ of duration $\ell_k = \tau_k - \tau_{k-1}$. If $\mathcal{S}$ is the rate-adjusted sample till time $\tau_K$,*

$$\mathbf{Pr}\left[e \in \mathcal{H}_k \mid e \in \mathcal{S}\right] = \ell_k / \sum_{k=1}^{K} \ell_k, \; \forall \, k \tag{3.2}$$

*which is independent of $|\mathcal{H}_k|$.*

When applied to real-world network traffic data with manually annotated attacks as anomalies, SedanSpot outperforms the then state-of-the-art by achieving statistically significantly higher precision for every value of recall as shown in Figure 3.2(a). Overall, SedanSpot is 270% (= 46% absolute percentage points) more accurate on this dataset than the baseline while also yielding a 3× speedup, as portrayed in Figure 3.2(b). Digger deeper into the performance gains, we observed that a significant portion ($0.17 \rightarrow 0.45$) of the accuracy gain is due to our new sampling strategy which significantly downsamples anomalous edges, thereby decreasing 'sample corruption' compared to a classic Uniform-Reservoir sampler (see Figure 3.2(c)). The remaining gain ($0.45 \rightarrow 0.63$) in accuracy is due to holistic edge anomaly scoring using random walks based on the whole graph which is more robust than methods relying only on the local neighborhood of the edge.

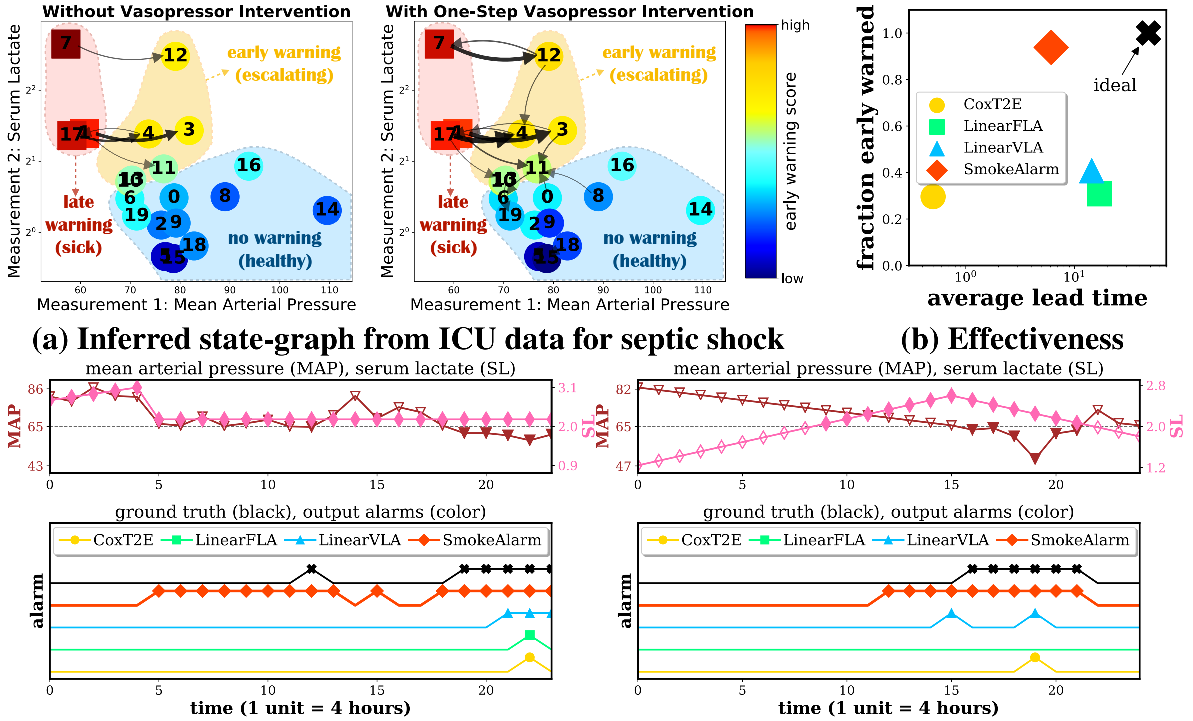
## 3.3 Early Warning of User-Input Anomalies

Section based on work that appeared in ICDM 2019 [19] [PDF].

**Goal:** "How can we early warn against user-input anomalies such an denial of service attack or an adverse health condition in near real-time? More challengingly, how do we learn to early warn from data containing confounding interventions–e.g., medicines–while remaining interpretable to the human decision maker?" Our work in §3.1 and §3.2 concerned the *detection* of anomalies that had already occurred. While this is a useful primitive to have, ideally, we want to be alerted in advance of upcoming anomalies so that preventive actions–e.g., safeguarding against expected network attacks, pulling over to the side of the road before a seizure–may be taken.

**Approach:** The key idea behind SMOKEALARM is to separately model the evolution of measurements in the presence and in the absence of interventions. During training, SMOKEALARM learns an interpretable state-transition graph from past data labeled with anomaly occurrences. The graph is learned by maximizing the posterior probability of observed data according to a novel probabilistic model which takes the stochastic and prolonged effect of interventions into account. Thus, the learned graph captures how long interventions last, and how measurements evolve in the presence and in the absence of their influence. When deployed, SMOKEALARM ingests measurements and interventions of a new trajectory, and used the learned graph to output early warning scores online. The early warning score captures the expected discounted anomaly occurrence count in the future assuming that no counteractive interventions will be given. We show that SMOKEALARM is an ideal early warning system in the sense that it produces high early warning scores when a future anomaly is more likely, or is expected to occur sooner; moreover it does not presuppose that counteractive-measures will necessarily be taken to avoid an anomaly in the future.

**Results:** We apply SMOKEALARM to real-world ICU data from Mimic-III database and train it to early warn against septic shock anomalies. Septic shock is an adverse outcome of bacterial infection in the ICU and is characterized by low mean arterial pressure (MAP < 65mmHg) and high serum lactate (SL > 2mmol/L), among other abnormalities. Figure 3.3(a) shows the graph representation learned by SMOKEALARM on this data. Vertices (states) are plotted according to their values for MAP and SL and colored according to the output early warning scores from the respective states (blue=low, red=high). We see that SMOKEALARM alarms when the patient already has a septic shock anomaly (red region), and does not warn when they are healthy (blue region). Importantly, it warns during the escalation to septic shock (yellow area), which is the



**(a) Inferred state-graph from ICU data for septic shock**

**(b) Effectiveness**

**(c) Real-world examples of early warning**

**Figure 3.3: SMOKEALARM: Early Warning of User-Input Anomalies**

opportune moment for intervention. SMOKEALARM also learns that vasopressor interventions tend to increase mean average pressure as indicated by dark, thick rightward arrows (right) and thus decreases early warning scores of low-MAP states. A practitioner can quickly inspect this graph and verify that SMOKEALARM indeed capitalizes on the correct signals to early warn.

We compare SMOKEALARM to prior methods based on their ability to early warn before the onset of septic shock, which is the hardest and the most valuable occurrence of septic shock to predict (compared to subsequent recurrences). Figure 3.3(b) plots the fraction of septic shock patients who are warned before onset vs. the average lead time of such a warning. We see that SMOKEALARM warns 93.7% patients with an average lead time of 6.1 hours before septic shock onset. The baselines warn only <41% patients; so their lead time arguably does not matter.

We further elaborate on this by presenting trajectories from two patients in Figure 3.3(c). The top panel shows the evolution of MAP and SL measurements and the dashed line (MAP=65, SL=2) separates the healthy (hollow markers) values from unhealthy (filled markers) ones. The bottom panel displays ground truth septic shock label (black crosses) and alarms output by various methods (colored markers) as spikes. In both cases, we see that SMOKEALARM is the only method which correctly early warns before the first occurrence of septic shock. Further, the timing of first early warning can also be explained: for the patient on the left, the first warning coincides with the sharp decline in MAP at $t = 5$, while for the patient on the right, the first warning is output around the time that one of the values (SL) crosses over into the unhealthy region. The baselines, on the other hand, however, either completely miss the septic shock anomaly or warn after the anomaly starts or provide a smaller lead time to take any preventive action.

## 3.4 [Proposed] Anomalous Motif Detection

**Goal:** "How can we detect anomalous occurrences of a specific graph substructure or motif–such as a triangle or a star–from a time-evolving graph in near real-time?" This problem not only naturally falls out of our completed work, but is also interesting from a theoretical standpoint because it considers the detection of the entire spectrum of anomalous graph substructures spanning from large dense subgraphs (§3.1) at one extreme, to edges (§3.2) at the other. To the best of our knowledge, prior work has not considered anomalousness of specific higher-order structures in either the static or the dynamic graph setting.

**Plan:** We will attempt to answer the following research questions in order:

- **[RQ1]** How do we characterize the anomalousness of an occurrence of a given motif?
- **[RQ2]** How can we detect anomalous occurrences of a motif offline and in near real-time?
- **[RQ3]** What anomalies do other motifs reveal that edges and large dense subgraphs miss?

A promising approach we have for addressing [RQ1] is to construct a motif-weighted graph from the original graph where an edge connecting a vertex pair appears if the two vertices participate in the same occurrence of the motif. We hypothesize that anomalous edges in this motif-weighted graph could potentially signal anomalous occurrences of the motif in the original graph.

# Chapter 4

# Related Work

In this chapter, we provide a brief survey of works closely related to the completed, ongoing and proposed works in the preceding chapters.

## 4.1 Static Graphs

Anomaly detection in static graphs is well-studied problem; we refer to [4] for a comprehensive survey. Due to our focus on semi-supervised learning (SSL) setting, we only survey prominent graph SSL works below.

**Background on graph SSL:** Given a graph and labels on a few vertices, traditional graph semi-supervised learning methods typically infer the labels for all vertices by optimizing a loss function of the form $\mathcal{L} = (1 - \eta)\mathcal{L}_s + \eta\mathcal{L}_g$. Here, the first term is the *supervised loss* which imposes a penalty when the inferred values on the labeled vertices differ from their given values and the second term is the *graph loss* which penalizes inferred values that are not *smooth* over the graph structure. A parameter $\eta \in (0, 1)$ trades off the two factors. Various graph SSL methods define their loss functions as variants of the above.

**Label propagation and spreading (homophily only):** By far, the most widely adopted graph SSL techniques are label propagation [74] and label spreading [73]. Label propagation (LP) clamps labeled vertices to their provided values and uses a graph Laplacian regularization, while label spreading (LS) uses a squared Euclidean penalty as supervised loss and *normalized* graph Laplacian regularization which is known to be better-behaved and more robust to noise [64]. Both these techniques permit closed-form solution and are extremely fast in practice, scaling well to billion-scale graphs. Consequently, a number of techniques build on top of these approaches to allow inductive generalization [6, 66], to incorporate certainty [61], and so on.

**Belief propagation (BP) for arbitrary network effects:** Belief Propagation [45] is an efficient inference algorithm in graphical models, which works by iteratively propagating network effects. It is guaranteed to converge to the true beliefs in a graph with no loops [46] or on a few other special cases [40]. However, in a general graph with cycles, the algorithm may not converge to the true beliefs, or even worse, it may not converge at all. Despite this drawback, in practice, loopy BP works well in practice and has been found to approximate the true beliefs well [41]. Consequently, it has been successfully applied to numerous settings such as errorcorrecting codes [22, 23], stereo matching in computer vision [21, 58], fraud detection [3, 43], malware detection [11], and interactive graph exploration [10]. The success of BP has increased the interest to approximate BP and to find closed-form solutions in specialized settings.

**Approximation attempts for BP:** The first successful linearizing approximation of BP [36] focused on unipartite graphs with two classes. This was later extended by [26] to the multivariate setting in undirected unipartite graphs with arbitrary number of classes and square, symmetric compatibility matrices. [25] attempted to extend this even further to $|T|$-partite networks. Independently, [68] used the degree of a node as a measure the confidence of belief to linearize BP. None of the above methods can handle a general heterogeneous graph with multiple types of nodes and edges, that our completed work in §2.1 is designed to handle.

**Efforts to incorporate uncertainty:** Efforts to incorporate uncertainty or confidence in node classification are fairly recent. [5], [61] and [42] overly penalize high degree vertices for their tendency to disagree with several neighbors. [20] and [27] address the problem of uncertainty but show poor scalability due to need to an optimization problem per iteration of propagation or cubic complexity in the number of vertices. [67] and [68] both introduce uncertainty, however the lead to counter-intuitive results when network effects are absent. Our completed work in §2.2 differs from these prior approaches in aiming to tackle the graph SSL problem by incorporating uncertainty and generalizing to arbitrary network effects in a principled manner.

**Higher-order network structures:** Recent work has shown that graphs from diverse domains have many striking higher-order network structures [7] which can be leveraged to improve graph clustering [71], link prediction [1, 8] and ranking [51]. Significant recent algorithmic advancements made in counting [34] and enumeration [13] of higher-order network structures (esp., cliques) enables and supports the aforementioned applications. From an SSL point of view, the explicit use of higher-order network structures has remained limited to belief propagation over $2 \times 2$ image cliques to improve image denoising, segmentation and rendering in computer vision [38, 47]. Till date, the importance of higher-order network structures in SSL atop explicit graph data has largely remained unexplored and our proposed work in §2.3 aims to address this gap.

## 4.2 Dynamic Graphs

Here, we review prominent works discussing anomaly detection in time-evolving graphs. For a more comprehensive treatment, refer to [48].

### 4.2.1 Detection

**Graph streams:** Many methods assume that the raw edge stream has been processed into a stream of graph snapshots (each containing edges from a given duration) before detecting anomalies. The traditional approach is to compare adjacent graphs $(\mathcal{G}_t, \mathcal{G}_{t+1})$ via a similarity function based on, e.g., belief propagation [37], random walks [57], etc., A possibly time-varying threshold is applied to the similarity scores to identify anomalies. These methods, in general, do not consider evolutionary/periodic trends. Dense subgraph detection based approaches [35, 54] model dynamic graphs as node×node×time tensors and aim to approximately identify the top-$k$ densest sub-blocks, e.g., persistent dense subgraphs. However, they do not detect anomalies in *near real-time*. Graph decomposition/partitioning based approaches [59, 60] store a summary of the graph structure based on tensor decomposition [59] or minimum description language [60] and identify change points as anomalies. Their primary focus is on the computationally hard problem of graph modeling and not anomaly detection. In contrast to these works, our completed work in §3.1 detects dense subgraph anomalies in near real-time.

**Edge streams:** In contrast, methods operating directly on the edge stream are relatively few. These include: [2] to score the likelihood of each edge in the stream based on a structural reservoir sample of edges, [72] to detect anomalous nodes using egonet-level Principal Component Analysis and [49] to score edge anomalousness in the stream based on its prior occurrence, preferential attachment and mutual neighbors (homophily). [39] is related, but applies only when *multiple* graphs with *typed* nodes and edges evolve simultaneously. Our completed work in §3.2 addresses burst resistance and holistic anomaly scoring, which prior works do not consider.

### 4.2.2 Early Warning

Prior work has considered early warning against various adverse events such as sepsis [24], septic shock [31] and heart failure [12]. In general, these approaches do not account for confounding interventions which "can mask the ground truth labels needed to train and evaluate a prediction system" [44].To cope with interventions, [9] advocates for *intelligible* models amenable to repairing by domain experts, e.g., by deleting incorrect rules such as asthma reduces the risk of pneumonia. [14] proposes a human-in-the-loop solution by seeking expert labels for pairwise comparisons of time points. More recently, [53] uses Counterfactual Gaussian Processes to forecast a single measurement in the presence of interventions. As such, [53] does not address the early warning problem and scales poorly with input size due to the use of Gaussian Processes.

State-based methods, which learn a graph from time-series data, can interpretably model the progression of trajectories. [65] infers a continuous-time Markov model for chronic obstructive pulmonary disease. [69] learns a probabilistic model to estimate the stages of chronic kidney disease. However, these methods are unsupervised, ignore interventions and do not early warn. Our completed work in §3.3 also employ a state-based model, but explicitly account for interventions, is interpretable, and capitalize on labels to early warn.

# Chapter 5

# Timeline

My proposed timeline is as follows:

- **Nov 2019:** Leveraging Higher-Order Structures for Graph Semi-Supervised Learning (§2.3)
- **Dec-Feb 2019:** Anomalous Motif Detection in Dynamic Graphs (§3.4)
- **Mar 2020:** Interviewing
- **Apr 2020:** Thesis writing
- **May 2020:** Thesis defense

# Chapter 6

# Conclusion

In this thesis, we address the problems of mining anomalies using static and dynamic graphs. Specifically,

**Static graphs (§2):** We broaden the scope of present literature on graph semi-supervised learning, a core problem in mining anomalies, by handling heterogeneous graphs, and leveraging label uncertainty for more accurate vertex labeling. As part of ongoing work, we will investigate the role of higher-order structures in vertex labeling.

**Dynamic graphs (§3):** We develop algorithms for real-time detection and early warning of anomalies so as to enable timely corrective or preventive measures. Our algorithms can detect anomalous dense subgraphs and edges in near real-time using limited memory, and can also early warn against user-input anomalies in the presence of interventions. As part of future work, we will characterize the anomalousness of specific graph substructures and develop algorithms to detect them in near real-time.

For reproducibility, we open-source most of the algorithms proposed in the thesis and mostly use publicly-available datasets in our work.

# Bibliography

[1] Ghadeer AbuOda, Gianmarco De Francisci Morales, and Ashraf Aboulnaga. Link prediction via higher-order motif features. *CoRR*, abs/1902.06679, 2019.

[2] Charu C. Aggarwal, Yuchen Zhao, and Philip S. Yu. Outlier detection in graph streams. In *ICDE*, pages 399–409. IEEE, 2011.

[3] Leman Akoglu, Rishi Chandy, and Christos Faloutsos. Opinion fraud detection in online reviews by network effects. In *ICWSM*, pages 2–11, 2013.

[4] Leman Akoglu, Hanghang Tong, and Danai Koutra. Graph based anomaly detection and description: a survey. *Data Min. Knowl. Discov.*, 29(3):626–688, 2015.

[5] Shumeet Baluja et al. Video suggestion and discovery for youtube: Taking random walks through the view graph. In *WWW*, pages 895–904, 2008.

[6] Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In *COLT*, volume 3120 of *Lecture Notes in Computer Science*, pages 624–638. Springer, 2004.

[7] Austin R Benson, David F Gleich, and Jure Leskovec. Higher-order organization of complex networks. *Science*, 353(6295):163–166, 2016.

[8] Austin R. Benson, Rediet Abebe, Michael T. Schaub, Ali Jadbabaie, and Jon M. Kleinberg. Simplicial closure and higher-order link prediction. *Proc. Natl. Acad. Sci. U.S.A.*, 115(48): E11221–E11230, 2018.

[9] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *KDD*, pages 1721–1730. ACM, 2015.

[10] Duen Horng Chau, Aniket Kittur, Jason I Hong, and Christos Faloutsos. Apolo: making sense of large network data by combining rich user interaction and machine learning. In *ACM SIGCHI*, pages 167–176, 2011.

[11] Duen Horng Chau, Carey Nachenberg, Jeffrey Wilhelm, Adam Wright, and Christos Faloutsos. Polonium: Tera-scale graph mining and inference for malware detection. pages 131–142. SIAM, 2011.

[12] Edward Choi, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. Using recurrent neural network models for early detection of heart failure onset. *JAMIA*, 24(2):361–370, 2017.

[13] Maximilien Danisch, Oana Denisa Balalau, and Mauro Sozio. Listing k-cliques in sparse real-world graphs. In *WWW*, pages 589–598. ACM, 2018.

[14] Kirill Dyagilev and Suchi Saria. Learning (predictive) risk scores in the presence of censoring due to interventions. *Machine Learning*, 102(3):323–348, 2016.

[15] Dhivya Eswaran and Christos Faloutsos. Sedanspot: Detecting anomalies in edge streams. In *ICDM*, pages 953–958. IEEE Computer Society, 2018.

[16] Dhivya Eswaran, Stephan Günnemann, and Christos Faloutsos. The power of certainty: A dirichlet-multinomial model for belief propagation. In *SDM*, pages 144–152. SIAM, 2017.

[17] Dhivya Eswaran, Stephan Günnemann, Christos Faloutsos, Disha Makhija, and Mohit Kumar. Zoobp: Belief propagation for heterogeneous networks. *PVLDB*, 10(5):625–636, 2017.

[18] Dhivya Eswaran, Christos Faloutsos, Sudipto Guha, and Nina Mishra. Spotlight: Detecting anomalies in streaming graphs. In *KDD*, pages 1378–1386. ACM, 2018.

[19] Dhivya Eswaran, Christos Faloutsos, Nina Mishra, and Yonatan Naamad. Intervention-aware early warning. In *ICDM*, pages 953–958. IEEE Computer Society, 2019.

[20] Yuan Fang, Bo-June Paul Hsu, and Kevin Chen-Chuan Chang. Confidence-aware graph regularization with heterogeneous pairwise features. In *SIGIR*, pages 951–960, 2012.

[21] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient belief propagation for early vision. *IJCV*, pages 41–54, 2006.

[22] Marc PC Fossorier, Miodrag Mihaljevic, and Hideki Imai. Reduced complexity iterative decoding of low-density parity check codes based on belief propagation. *IEEE Transactions on communications*, pages 673–680.

[23] Brendan J. Frey and Frank R. Kschischang. Probability propagation and iterative decoding. In *Allerton Conference on Communications, Control and Computing*, pages 482–493, 1996.

[24] Joseph Futoma, Sanjay Hariharan, Katherine A. Heller, Mark Sendak, Nathan Brajer, Meredith Clement, Armando Bedoya, and Cara O'Brien. An improved multi-output gaussian process RNN with real-time validation for early sepsis detection. In *MLHC*, volume 68, pages 243–254. PMLR, 2017.

[25] Wolfgang Gatterbauer. The linearization of pairwise markov networks. *arXiv preprint arXiv:1502.04956*, 2015.

[26] Wolfgang Gatterbauer, Stephan Günnemann, Danai Koutra, and Christos Faloutsos. Linearized and single-pass belief propagation. *PVLDB*, 8(5):581–592, 2015.

[27] Chen Gong, Dacheng Tao, Keren Fu, and Jie Yang. Relish: Reliable label inference via smoothness hypothesis. In *AAAI*, 2014.

[28] Robert A Hanneman and Mark Riddle. Introduction to social network methods, 2005.

[29] Douglas M Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[30] Harold V Henderson and Shayle R Searle. The vec-permutation matrix, the vec operator and kronecker products: A review. *Linear and multilinear algebra*, pages 271–288, 1981.

[31] Katharine E Henry, David N Hager, Peter J Pronovost, and Suchi Saria. A targeted real-time early warning score (trewscore) for septic shock. *Science translational medicine*, 7 (299):299ra122–299ra122, 2015.

[32] Matthew O Jackson. *Social and economic networks*. Princeton university press, 2010.

[33] Matthew O Jackson, Tomas Rodriguez-Barraquer, and Xu Tan. Social capital and social quilts: Network patterns of favor exchange. *American Economic Review*, 102(5):1857–97, 2012.

[34] Shweta Jain and C. Seshadhri. A fast and provable method for estimating clique counts using turán's theorem. In *WWW*, pages 441–449. ACM, 2017.

[35] Meng Jiang, Alex Beutel, Peng Cui, Bryan Hooi, Shiqiang Yang, and Christos Faloutsos. A general suspiciousness metric for dense blocks in multimodal data. In *ICDM*, pages 781–786. IEEE, 2015.

[36] Danai Koutra, Tai-You Ke, U Kang, Duen Horng Polo Chau, Hsing-Kuo Kenneth Pao, and Christos Faloutsos. Unifying guilt-by-association approaches: Theorems and fast algorithms. In *ECML PKDD*, pages 245–260, 2011.

[37] Danai Koutra, Neil Shah, Joshua T. Vogelstein, Brian Gallagher, and Christos Faloutsos. Deltacon: Principled massive-graph similarity function with attribution. volume 10, pages 28:1–28:43, 2016.

[38] Xiangyang Lan, Stefan Roth, Daniel P. Huttenlocher, and Michael J. Black. Efficient belief propagation with learned higher-order markov random fields. In *ECCV (2)*, volume 3952 of *Lecture Notes in Computer Science*, pages 269–282. Springer, 2006.

[39] Emaad A. Manzoor, Sadegh M. Milajerdi, and Leman Akoglu. Fast memory-efficient anomaly detection in streaming heterogeneous graphs. In *KDD*, pages 1035–1044. ACM, 2016.

[40] Joris M Mooij and Hilbert J Kappen. Sufficient conditions for convergence of the sum–product algorithm. *IEEE Transactions on Information Theory*, pages 4422–4437, 2007.

[41] Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, pages 467–475, 1999.

[42] Matan Orbach and Koby Crammer. Graph-based transduction with confidence. In *ECMLPKDD*, pages 323–338. Springer, 2012.

[43] Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *WWW*, pages 201–210, 2007.

[44] Chris Paxton, Alexandru Niculescu-Mizil, and Suchi Saria. Developing predictive models using electronic medical records: challenges and pitfalls. In *AMIA Annual Symposium Proceedings*, page 1109, 2013.

[45] Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *AAAI*, pages 133–136, 1982.

[46] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 2014.

[47] Brian Potetz and Tai Sing Lee. Efficient belief propagation for higher-order cliques using linear constraint nodes. *Computer Vision and Image Understanding*, 112(1):39–54, 2008.

[48] Stephen Ranshous, Shitian Shen, Danai Koutra, Steve Harenberg, Christos Faloutsos, and Nagiza F Samatova. Anomaly detection in dynamic networks: a survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(3):223–247, 2015.

[49] Stephen Ranshous, Steve Harenberg, Kshitij Sharma, and Nagiza F Samatova. A scalable approach for outlier detection in edge streams using sketch-based approximations. In *SDM*, pages 189–197. SIAM, 2016.

[50] Stephen Ranshous, Mandar S. Chaudhary, and Nagiza F. Samatova. Efficient outlier detection in hyperedge streams using minhash and locality-sensitive hashing. In *COMPLEX NETWORKS*, volume 689 of *Studies in Computational Intelligence*, pages 105–116. Springer, 2017.

[51] Ryan A. Rossi, Anup Rao, Sungchul Kim, Eunyee Koh, Nesreen K. Ahmed, and Gang Wu. Higher-order ranking and link prediction: From closing triangles to closing higher-order motifs. *CoRR*, abs/1906.05059, 2019.

[52] Yousef Saad. *Iterative methods for sparse linear systems*, volume 82. SIAM, 2003.

[53] Peter Schulam and Suchi Saria. Reliable decision support using counterfactual models. In *NIPS*, pages 1697–1708, 2017.

[54] Kijung Shin, Bryan Hooi, and Christos Faloutsos. M-zoom: Fast dense-block detection in tensors with quality guarantees. In *ECML/PKDD*, volume 9851, pages 264–280. Springer, 2016.

[55] Jorge G. Silva and Rebecca Willett. Hypergraph-based anomaly detection of high-dimensional co-occurrences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(3):563–569, 2009.

[56] Ann Sizemore, Chad Giusti, and Danielle S Bassett. Classification of weighted networks through mesoscale homological features. *Journal of Complex Networks*, 5(2):245–273, 2017.

[57] Kumar Sricharan and Kamalika Das. Localizing anomalous changes in time-evolving graphs. In *SIGMOD*, pages 1347–1358. ACM, 2014.

[58] Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on pattern analysis and machine intelligence*, pages 787–800, 2003.

[59] Jimeng Sun, Dacheng Tao, and Christos Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In *KDD*, pages 374–383. ACM, 2006.

[60] Jimeng Sun, Christos Faloutsos, Spiros Papadimitriou, and Philip S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *KDD*, pages 687–696. ACM, 2007.

[61] Partha Pratim Talukdar and Koby Crammer. New regularized algorithms for transductive learning. In *ECML/PKDD (2)*, volume 5782 of *Lecture Notes in Computer Science*, pages 442–457. Springer, 2009.

[62] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. In *ICDM*, pages 613–622. IEEE, 2006.

[63] Alexei Vazquez, Alessandro Flammini, Amos Maritan, and Alessandro Vespignani. Global protein function prediction from protein-protein interaction networks. *Nature biotechnology*, 21(6):697, 2003.

[64] Ulrike Von Luxburg, Mikhail Belkin, and Olivier Bousquet. Consistency of spectral clustering. *The Annals of Statistics*, pages 555–586, 2008.

[65] Xiang Wang, David Sontag, and Fei Wang. Unsupervised learning of disease progression models. In *KDD*, pages 85–94. ACM, 2014.

[66] Jason Weston, Frédéric Ratle, and Ronan Collobert. Deep learning via semi-supervised embedding. In *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 1168–1175. ACM, 2008.

[67] Yuto Yamaguchi, Christos Faloutsos, and Hiroyuki Kitagawa. Socnl: Bayesian label propagation with confidence. In *PAKDD*, pages 633–645, 2015.

[68] Yuto Yamaguchi, Christos Faloutsos, and Hiroyuki Kitagawa. Camlp: Confidence-aware modulated label propagation. In *SDM*, 2016.

[69] Jaewon Yang, Julian J. McAuley, Jure Leskovec, Paea LePendu, and Nigam Shah. Finding progression stages in time-evolving event sequences. In *WWW*, pages 783–794. ACM, 2014.

[70] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Understanding belief propagation and its generalizations. In *Exploring artificial intelligence in the new millennium*, pages 239–269. Morgan Kaufmann Publishers Inc., 2003.

[71] Hao Yin, Austin R Benson, and Jure Leskovec. Higher-order clustering in networks. *Physical Review E*, 97(5):052306, 2018.

[72] Weiren Yu, Charu C Aggarwal, Shuai Ma, and Haixun Wang. On anomalous hotspot discovery in graph streams. In *ICDM*, pages 1271–1276. IEEE, 2013.

[73] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *NIPS*, pages 321–328. MIT Press, 2003.

[74] Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 912–919. AAAI Press, 2003.