ASSIST: Automatic Summarization of SIgnificant Structural Changes in Large Temporal Graphs

Charalampos Chelmis*
University at Albany, SUNY
Albany, New York
cchelmis@albany.edu

Reshul Dani
College of Engineering Pune
Pune, Maharashtra, India
reshul.dani@gmail.com

ABSTRACT

Detecting outliers and anomalies in data is vital in numerous applications in areas such as security, finance, health care, and online social media. Such dynamic systems can be modeled as graphs that change over time. Even though considerable work has been performed on finding points in time at which a network notably differs from its past, little work has been done on characterizing or explaining such changes. However, in the era of big data where networked data are getting bigger and bigger, being able to summarize such changes is key for sensemaking and root cause analysis. To address this gap, we present a novel approach to summarize significant structural changes in large temporal graphs. Specifically, we propose an efficient approach to help the user understand sharp changes in the structure of the network by presenting to her only a summary of key subgraphs that contribute most to the change. Extensive evaluation on real-world datasets with ground truth demonstrates both quantitatively and qualitatively the ability of our approach to accurately detect changes and discover "important" structures to succinctly describe the change.

CCS CONCEPTS

•**Information systems** →**Web mining; Summarization;** *Data management systems; Data stream mining;* •**Computing methodologies** →**Anomaly detection;**

KEYWORDS

Anomaly summarization; structural change interpretation; dynamic graph; change attribution

1 INTRODUCTION

Graphs arise in numerous domains [1, 3, 5, 6] including security (e.g., network intrusion), finance (e.g., credit card fraud), health care (e.g., disease outbreaks), and online social media (e.g., fake news spread). A key problem in analyzing both static and dynamic graphs in such settings is spotting suspicious behavior or events [1, 3] before proceeding with post-even analysis. Even though event detection in temporal graphs is a well-studied problem [1, 11], there is currently no way for a practitioner to explain an event. Contrast that with Figure 1, where we highlight the most informative structure that can be used to explain the detected event.

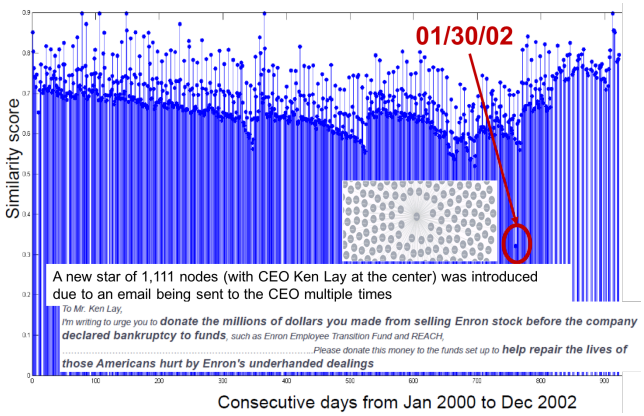

Figure 1: Informative structure identified by *ASSIST* (Section 2.3) to explain a sharp structural change in the Enron communication network (see Section 3 for details). Without *ASSIST* no clear structures stand out.

In this paper, we focus on constructing concise summaries of significant structural changes in large, real-world dynamic graphs in order to better communicate such changes to practitioners. Our hypothesis is that sharp changes in the structure of a graph are not random but instead indicate patterns, which should be discovered and ranked based on their contribution to the change.

We propose *ASSIST* (Section 2.3), a scalable approach to concisely summarize significant structural changes in large, dynamic real-world graphs in terms of a lexicon of subgraph-types. Figure 1 shows an unnaturally large star structure of 1, 111 nodes that appears on Jan. 30^{th} of 2002. The finding is the result of applying *ASSIST* to a real-world dataset of 3.6M Enron emails sent between 95.8K email addresses over 4 years. The structure corresponds to a large collection of emails sent to the Enron CEO. Identification of such structures would be labor intensive and time consuming or impossible within reasonable time constraints and with limited available human resources without our proposed approach.

We summarize our contributions as follows:

- We identify the problem of building automatic summaries of significant structural changes in large temporal networks in a fully unsupervised and parameter-free fashion.
- We propose *ASSIST*, a novel approach for summarizing significant structural changes in dynamic graphs.
- We report experiments on real datasets and show that *ASSIST* is *intuitive*, *scalable* and *practical*.

*Corresponding author.

WebSci'17, June 25-28, 2017, Troy, NY, USA.

DOI: http://dx.doi.org/10.1145/3091478.3091518

2 PROPOSED APPROACH

2.1 Significant Structural Change Detection

Many dynamic systems can be modeled as temporal graphs, which change over time with new nodes and edges arriving or existing nodes and edges disappearing [1]. Event detection in temporal graphs corresponds to finding points in time at which the graph structure notably differs from its past. Formally, given a sequence of graphs $\{G_1, G_2, \ldots, G_t\}$, find time points τ such that G_τ differs significantly from $G_{\tau-1}$.

Several methods have been proposed for the above problem, a survey of which is given in [1]. To date, however, there exists no single method that has been shown to outperform all others, mainly due to the fact that different techniques may identify different classes or types of events depending on their particular formulation.

In this paper, we identify significant changes in structrure by measuring the similarity in connectivity between two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$. We do so by computing pairwise node affinities in G_1, and comparing them with corersponding affinities in G_2. We use Fast Belief Propagation [8] to compute the affinity, s_{ij}, of any node j with respect to node i in a graph. Other similarity measures (e.g., [17]) can be used, however Fast Belief Propagation has been shown to be linear on the number of edges [8].

We assume that G_1 and G_2 have the same set of nodes V. If the node sets V_1 and V_2 are different, then we assume that $V = V_1 \cup V_2$, and some nodes are disconnected either in G_1 or G_2. Once the pairwise node affinity scores $\mathbf{S}_1$ and $\mathbf{S}_2$ have been computed for each graph G_1 and G_2, we measure the root Euclidean distance of $\mathbf{S}_1$ and $\mathbf{S}_2$, $d(\mathbf{S}_1, \mathbf{S}_2)$, which has been shown to detect even small changes [8]. We subsequently convert the distance to a similarity measure, i.e., $\sigma(G_1, G_2) = \sigma(\mathbf{S}_1, \mathbf{S}_2) = \frac{1}{1+d(\mathbf{S}_1, \mathbf{S}_2))}$, bounding the result to the interval $[0, 1]$. A similarity score of 1 means identical graphs (i.e., the temporal graph has remained unchanged), whereas a similarity score of 0 means totally different graphs.

2.2 Structural Change Summarization

Once a significant structural change between two snapshots G_τ and $G_{\tau-1}$ has been detected (Section 2.1), we need to identify the best collection M of possibly overlapping subgraphs that succinctly describe the main connectivity of the structural change. The rationale is that people can understand more easily a handful of simple structures (ranked by importance) rather than cluttered graphs.

To produce the set of subgraphs that best describes a sharp structural change we rely on a lexicon, Γ, comprising cliques (fc) and near-cliques (nc), stars (st), chains (ch) and (near) bi-partite cores (nb, bc). These structures appear often in real-world networks [1] and have real semantic meaning in practice (e.g., bi-partite cores signify fraudulent activity in transaction networks [12]).

We formulate our task as a **lossless compression problem**. Specifically, we assert that the best summary of a significant structural change is a set of subgraphs M that compresses it best, and, thus, helps a human understand the main characteristics of the change in a simple, intuitive manner. We use Minimum Description Length (MDL) [13], a parameter-free approach to identify M. Given a set of models $\mathcal{M}$, the best model $M \in \mathcal{M}$ by MDL for some observed graph snapshot G_t is that which minimizes $L(M) + L(G_t|M)$, where $L(M)$ is the length in bits used to describe M and $L(G_t|M)$ is the length in bits used to describe G_t encoded using M.

We consider models $M \in \mathcal{M}$ to be composed of ordered lists of graph structures, with possible node (but not edge) overlaps. Each structure $\gamma \in M$ describes the interconnectivity of a subset of nodes in the adjacency matrix $\mathbf{A}$. The family of models $\mathcal{M}$ in our approach consists of all possible permutations of all possible subsets of C, where $C = \cup_x C_x$ denotes the set of possible subgraph-types $x \in \Gamma$ over all possible subsets of V. For example, C_{fc} is the set of all possible full cliques. Given M, we induce the approximation $\mathbf{M}$ of $\mathbf{A}$ as described by each structure $\gamma \in M$, i.e., for each structure γ, we induce the edges of G_t described by γ in $\mathbf{M}$ accordingly. Given that M is a summary to $\mathbf{A}$ it is likely that $\mathbf{M} \neq \mathbf{A}$. Thus, we also compute the error matrix $\mathbf{E} = \mathbf{M} \oplus \mathbf{A}$ so that $\mathbf{A}$ can be reconstructed in a lossless manner from $\mathbf{M}$ and $\mathbf{E}$.

We formalize the problem we tackle as follows:

Given snapshots G_τ and $G_{\tau-1}$ involved in a significant structural change of temporal graph G, and lexicon Γ of sungraph-types;
Find the minimum description model $M' = (M_\tau - M_{\tau-1}) \cup (M_{\tau-1} - M_\tau)$, where M_τ and $M_{\tau-1}$ are the minimum description models that minimize the total encoding lengths $L_{G_\tau}(M_\tau) = L(M_\tau) + L(\mathbf{E}_\tau)$ and $L_{G_{\tau-1}}(M_{\tau-1}) = L(M_{\tau-1}) + L(\mathbf{E}_{\tau-1})$ respectively, where $L(\gamma)$ denotes the number of bits required to describe γ.

Encoding the Model. To encode the length of a model, $L(M)$, we begin by encoding the total number of structures in M. Next, we optimally encode the number of structures of each type $x \in \Gamma$, the type for each structure, and finally the structure itself. We define the encoding for each subgraph-type in our lexicon such that we can compute the encoded length of a model for all structures in the model. For example, we encode a *star* (st), in which a single hub is connected to a set of two or more peripheral nodes as $L(st) = L_N(|st| - 1) + log_2 n + log_2\binom{n-1}{|st|-1}$, where the first term encodes the number of peripheral nodes (L_N denotes the MDL optimal encoding for integers ≥ 1 [13]), followed by the id of the hub, and the ids of the peripheral nodes. Similarly, we encode a *clique* (fc), in which all nodes are directly connected to all other nodes, by encoding the number of nodes in the clique followed by their ids such that $L(fc) = L_N(|fc|) + log\binom{n}{|fc|}$. We define the encoded length of *near cliques*, *bipartite cores* and *chains* similarly.

Encoding the Error. Next, we discuss how we encode errors made by $\mathbf{M}$ into matrix $\mathbf{E}$. Specifically, we consider two types of connectivity errors, $\mathbf{E}^+$ and $\mathbf{E}^-$, as these two sources of error are likely to have different error distributions. The former corresponds to the areas of $\mathbf{A}$ that M models but for which $\mathbf{M}$ introduces extraneous edges not present in the original graph. The latter consists of the areas of $\mathbf{A}$ which M doesn't model. The encoding for $\mathbf{E}^+$ (similarly $\mathbf{E}^-$) is as follows: $L(\mathbf{E}^+) = log(|\mathbf{E}^+|) + ||\mathbf{E}^+||_{l_1} + ||\mathbf{E}^+||_{l_0}$.

2.3 Automatic Summarization of Significant Structural Changes in Temporal Graphs

We propose *ASSIST* (Alg. 1), an approach for Automatic Summarization of SIgnificant Structural changes in large Temporal graphs. Both components of *ASSIST* are linear on the number of edges. Specifically, Fast Belief Propagation, which we use to detect singificant structural changes has $O(|E|)$ complexity [8]. For summarization, the complexity of calculating the encoding scheme is $O(|E|)$

when there is no overlap between structures in M. In the case of overlapping structures, the complexity of computing the encoding scheme becomes $O(|M|^2 + |E|)$. This is because for any two overlapping structures γ_1 and γ_2, such that γ_1 comes before γ_2, finding how much of **A** γ_2 explains relative to γ_1 requires $O(|M|^2)$ operations. Typically, $|M| \ll |E|$. Thus, *ASSIST* has $O(|E|)$ runtime complexity.

Algorithm 1: ASSIST

Input: A sequence of graphs $\{G_1, G_2, \ldots, G_t\}$ and lexicon Γ
Detection of significant structural changes (Sect. 2.1):
Find time points τ s.t. G_τ differs significantly from $G_{\tau-1}$.
for each pair G_τ and $G_{\tau-1}$
Generate summary $M_{\tau-1}$ **of** $G_{\tau-1}$ **using Alg. 2**
Generate summary M_τ **of** G_τ **using Alg. 2**
Compose structural change summary: Construct model $M'_\tau = (M_\tau - M_{\tau-1}) \cup (M_{\tau-1} - M_\tau)$.
Return the collection of $M'_\tau, \forall\tau$ (and their encoding costs).

We begin by identifying time points τ such that G_τ differs significantly from $G_{\tau-1}$ as discussed in Section 2.1. Once graphs $G_{\tau-1}$ and G_τ have been identified, we use the encoding scheme discussed in Section 2.2 on each graph respectively to identify perfect and approximate structures that minimize their corresponding encoding cost. Specifically, given a subgraph in G_τ (similarly $G_{\tau-1}$), we first search for error-free matches with subgraph-types in our lexicon (e.g., perfect clique, or near-clique with missing edges encoded as error). If the subgraph does not belong to Γ, the search continues for the lexicon entry that best approximates the subgraph. Once the structure that has the lowest encoding cost has been identified, we add the structure to the candidate set C. Any combination of clustering or community detection algorithms [10] can be used to decompose G_τ (similarly $G_{\tau-1}$) into subgraphs.

Finally, given a set of candidate structures, C, we induce the model M by associating each candidate structure with the number of bits that are gained by encoding the subgraph as structure x instead of noise, and considering the structures in decreasing order (Section 2.2). Specifically, to reduce the search space of all possible permutations we consider each structure in C sequentially, and include it in M if the graph's encoding cost does not increase. Note that without this heuristic, we would have to consider all possible ordered combinations of the candidate structures; this would lead to a combinatorial problem that would render our solution impractical for any non-trivial candidate set. We compose the summary of the change at time τ by finding the smallest model $M' = M_\tau \cap M_{\tau-1}$ that best summarizes the change from $G_{\tau-1}$ to G_τ.

As new significant structural changes are detected, *ASSIST* tries to link them with detected changes from earlier timeslots. Note that depending on the frequency with which snapshots are acquired a single event may be split across multiple time points. Conversely, multiple events may be mistaken for being one event if the temporal resolution of graph snapshot collection is too coarse. Intuitively, if two graph snapshots of successive timeslots share a significant amount of structures in their summaries, the later snapshot arises from the previous one. To identify that two significant structural changes are referring to the same event we use the structural similarity between the underlying summaries, expressed as the Jaccard coefficient of their structures. If this similarity exceeds a certain threshold, we merge the two changes into a single event.

Algorithm 2: Summary (Sect. 2.2)

Input: Graph G and lexicon Γ
Generate Candidate Structures: Generate candidate subgraphs using graph decomposition methods.
Label Candidate Structures: Label each subgraph as as a perfect or approximate structure $x \in \Gamma$ which minimizes the local encoding cost. Populate the candidate set C.
Summary Composition: Compose a model M of non-redundant structures which summarizes G. Choose M that produces the smallest total encoding cost.

3 EXPERIMENTAL EVALUATION

3.1 Datasets and Experimental Setup

We ran our experiments on a Intel(R) Core(TM) i5-4200U CPU at 1.60GHz with 4GB memory. The significant structure change detection component (Section 2.1) is implemented in Matlab, while the summarization process (Section 2.2) in Python. We use two publicly available real-wolrd dynamic graph datasets [1] for our evaluation.

Large-Scale Twitter Dataset with Ground Truth [1]. We use tweets collected over a period which spans the World Cup 2014 season (Jun 12 - Jul 13), and filtered by popular/official World Cup hashtags, such as #worldcup, #fifa, and #brazil. After named entity extraction and resolution (URLs, hashtags, @ mentions), we created entity-entity co-mention temporal graphs on 5 minute sample rate for a total of $1,790,999$ time points. Each snapshot contains up to $161,895$ nodes. Compiled ground truth events comprise the goals, penalties, and injuries in all matches that involve at least one of Argentina, Brazil, France, Germany, Netherlands, and Spain teams.

Enron Emails Dataset [1]. We use three years of Enron emails (Jan 01 2000 - Dec 31 2002) which exhibit many periodic structures (e.g., near-cliques that persevere with time) as a reflection of office e-mail correspondence, and one-shot structures (e.g., stars) due to events such as company-wide emails sent out by key players John Lavorato (CEO of Enron America), Sally Beck (COO) and Kenneth Lay (CEO/Chairman). In our temporal graphs, nodes represent email addresses ($95,829$ addresses in total) and directed edges depict sent/received relations. We analyze the data with daily sample rate skipping weekends ($1,069$ time points in total).

3.2 Significant Structural Change Detection

We begin by evaluating *ASSIST* on detection accuracy as a function of temporal granularity and similarity threshold. Specifically, we experiment with entity-entity co-mention graphs for time windows of 5, 10, and 20 mins and compute the similarity scores between consecutive graphs. We identify graphs G_t that differ "too much" from G_{t-1} if their similarity score $s(G_t, G_{t-1})$ drops below a threshold. To quantify the **threshold impact** on the performance of *ASSIST*, we perform a grid search over threshold values (i.e., threshold is defined as median$\pm 4\sigma$ with step 0.5). High similarities between consecutive graph snapshots (e.g., during weekends in the Enron

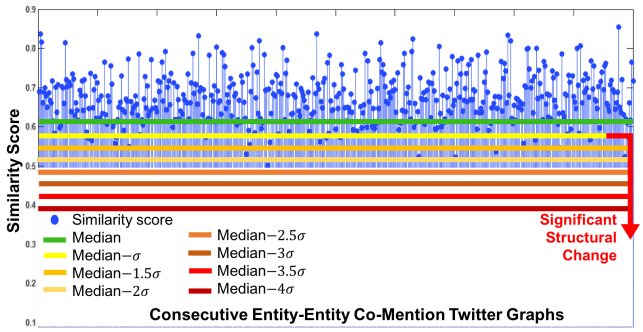

Figure 2: Significant changes identified by *ASSIST* coincide with major World Cup events. Blue points denote similarity scores between consecutive instances of the entity-entity co-mention graphs. Horizontal lines represent lower control limits for median $-k\sigma$, where $k \in [1, 4]$ increases in 0.5 steps.

Emails dataset due to less email communication) do not reflect significant structural changes and hence, we report our findings for median$-k\sigma$, where $k \in [1, 4]$ increases in 0.5 steps.

Figure 2 shows the similarity scores between consecutive entity-entity co-mention Twitter graphs with "anomalous" points laying below the corresponding lower control limits. For threshold set to median-2σ, 20 points relate to actual events that span 13 games in the World Cup. For example, *ASSIST* summarized the goal scored by player Kroos in the infamous match (Game 61) between Brazil and Germany on 8th July 2014 as 2 stars of $\sim$ 200 nodes with hashtags '#tragedy' and '#heartache' as the centers. *ASSIST* also identified stars with '#worldCup', '#worldCup2014' and entities 'brazil' and 'germany' at the time the goals happened during the match.

So far, we used entity-entity co-mention graphs of 5min intervals to evaluate the performance of *ASSIST*. We have also compared its performance for different time windows (5mins, 10mins, 20mins). The performance of *ASSIST* is similar for windows of 10mins and 20mins. This indicates that **temporal granularity** does not impact the performance of *ASSIST*.

ASSIST should not be penalized for an inaccurate detector. This is particularly true when multiple events happen almost **instantaneously** as is the case for our Twitter World Cup dataset. Specifically, the events in this data (e.g., goals and injuries) are quite instantaneous when we use a sample rate of 5 minutes. Moreover, such events are likely to be reflected on Twitter with some **delay** by social media users. As such, it is extremely hard for *ASSIST* to pinpoint the exact time of the events. However, some time delay in detecting an event is often tolerable in practice (e.g., detecting an event that occurred at τ within time window $[\tau - 5; \tau + 5]$ is counted as accurate). For example, during Game 61, 3 goals happened very close to each other (at the 24th, 26th and 29th minutes). Player Kroos scored the 24th and 26th minute goals, whereas player Khedira scored the 29th goal. *ASSIST* could not differentiate between these events using a 5mins window since summaries generated by *ASSIST* comprised mainly stars with centers such as '#worldcup', '#ned', and '#aus' for a match between Australia and Netherlands, making it difficult to identify the exact cause behind such event. To address this issue, we experiment with temporal graphs of 1min sample rate for this match. In this case, the summaries generated by *ASSIST* showed the presence of entity 'Kroos' for the goals in the 24th and 26th minute, whereas entity 'Khedira' was included in the summary for the goal in 29th minute.

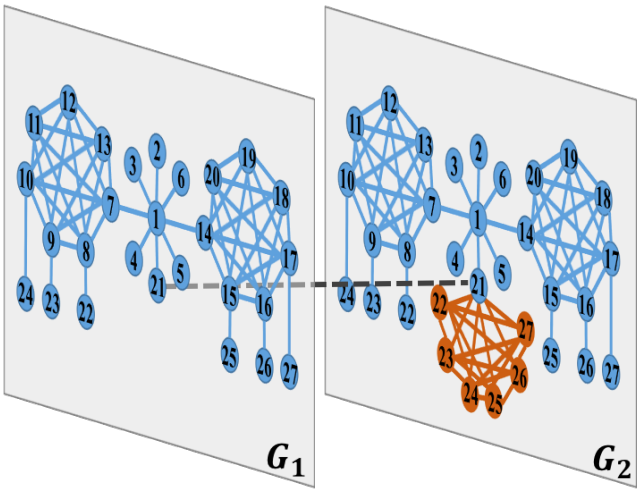

Figure 3: *ASSIST* reports the emergent full clique $\{22, 23, 24, 25, 26, 27\}$ as the cause of the sharp structural change in this toy graph sequence.

3.3 Summarization Quality

To demonstrate that *ASSIST* produces **intuitive** and **informative** summaries of sharp structural changes in temporal graphs, we begin by conducting experiments on small toy networks that include classic topologies (i.e., cliques and stars).

Figures 3 and 4 show two sample graph sequences where *ASSIST* detects the introduction of a clique and the dissolution of a clique into a star. In Figure 3, the intermediate summaries produced by *ASSIST* for G_1 comprises full cliques (fc) $\{14, 15, 16, 17, 18, 19, 20\}$ and $\{7, 8, 9, 10, 11, 12, 13\}$, whereas for G_2 fc $\{14, 15, 16, 17, 18, 19, 20\}$, $\{7, 8, 9, 10, 11, 12, 13\}$, and $\{22, 23, 24, 25, 26, 27\}$. As a result, *ASSIST* reports the emergent fc $\{22, 23, 24, 25, 26, 27\}$ as the cause of the sharp structural change. Similarly, as shown in Figure 4, *ASSIST* generates intermediate summaries consisting of fc $\{15, 16, 17, 18, 19, 20\}$ and $\{8, 9, 10, 11, 12, 13\}$ for G_1 and fc $\{8, 9, 10, 11, 12, 13\}$ for G_2 accordingly. As a consequence, *ASSIST* reports the dissolution of fc $\{15, 16, 17, 18, 19, 20\}$ as the cause of the sharp structural change.

Next, we evaluate *ASSIST* using the Enron communication dataset. The visualization of events identified by *ASSIST* in this dataset (e.g., Figure 1) confirm that sharp topological changes in the Enron communication network correspond to major events in the company's history. For instance, the unnaturally large star structure of 1, 111 nodes that appears on Jan. 30^{th} of 2002 is due to a large collection of emails being sent to the Enron CEO at that time. Such **informative** structures contain (often at their center) high ranking officials.

Finally, we evaluate the ability of *ASSIST* to produce **concise** summaries by measuring the compression ratio, $C_R = \frac{|V_S|}{|V|}$, between the total number of nodes that a practitioner would be required to examine to identify the root cause of a sharp structural change between two graph snapshots $G_\tau = (V, E_\tau)$ and $G_{\tau-1}(V, E_{\tau-1})$, and the number of nodes in the summary $S = (V_S, E_S), V_s \subset V$ produced by *ASSIST*. Thus, smaller compression

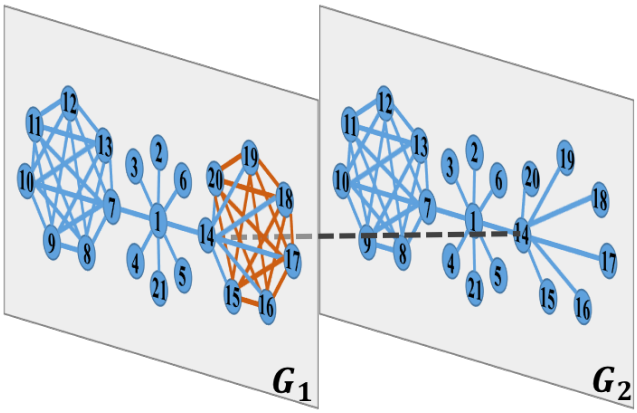

Figure 4: For this sample graph sequence *ASSIST* reports the dissolution of full clique {15, 16, 17, 18, 19, 20} **as the cause of the sharp structural change.**

ratios indicate better summaries. Note that we use compression ratio not as our end goal but as a principled way for routing the attention of practitioners to significant structures that we believe are associated with the sharp structural changes in the graph.

In our experiments, we found that *ASSIST* produces models with few informative structures (mainly stars and near-bipartite cores) due to its high effectiveness in pruning redundant, overlapping or error-prone structures from the candidate set C, and rewarding structures that explain unexplored parts of the change. For example, *ASSIST* summarizes the event on 30th Jan 2002 in the Enron dataset using just 37% of the total number of nodes; this corresponds to a compression ratio of 63%. Similarly, *ASSIST* achieves a compression ratio of 69% by summarizing the 24th minute goal in Game 61 in the Twitter World Cup dataset.

4 RELATED WORK

Related work comprises three main areas. The advantage of *ASSIST* compared to prior work is that it is scalable, intuitive, extensible, and parameter-free. No existing method has all these properties.

Graph Compression and Summarization. Prior works, such as [2, 4, 7, 9] operate on static graphs. A recently proposed method for dynamic graph summarization [16] uses MDL to discover temporally coherent subgraphs.

Event and Anomaly Detection in Graphs. A comprehensive overview of existing graph-based approaches for event and anomaly detection [11, 15], as well as the open research challenges in developing such algorithms have been presented at [1]. Such approaches focus on anomaly detection rather than the summarization and interpretation of detected abnormalities.

Event Detection and Summarization. Identifying and summarizing events is a fundamental and challenging problem of practical importance in many fields of science [3, 5, 6, 14]. None of these methods focuses on both identifying structural changes in temporal graphs and summarizing these changes at the same time.

5 CONCLUSION AND FUTURE WORK

We identified and studied the problem of building automatic summaries of significant structural changes in large-scale temporal networks in a fully unsupervised and parameter-free fashion. To address this problem, we proposed a *scalable* approach, and we showed that it is producing summaries that agree with common sense and can be easily explained in real-world datasets.

Future work includes incorporating more subgraph-types in our lexicon or conversely tackling the problem with when a lexicon of subgraph-types is unknown, as well as boosting detection performance by exploring other methods for event and anomaly detection in graphs. Another possible direction is in developing a system for the early discovery, warning, prediction, and prevention of disruptive events that cannot be found in historical data. Such "predictive analytics" is of high value and has received significant attention in the last decade mainly due to the Big Data and Data Science revolution.

REFERENCES

[1] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2015. Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery* 29, 3 (2015), 626–688.
[2] Paolo Boldi and Sebastiano Vigna. 2004. The webgraph framework I: compression techniques. In *Proceedings of the 13th international conference on World Wide Web*. ACM, 595–602.
[3] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM computing surveys (CSUR)* 41, 3 (2009), 15.
[4] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, Michael Mitzenmacher, Alessandro Panconesi, and Prabhakar Raghavan. 2009. On compressing social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 219–228.
[5] Victoria J Hodge and Jim Austin. 2004. A survey of outlier detection methodologies. *Artificial intelligence review* 22, 2 (2004), 85–126.
[6] Raymond Kosala and Hendrik Blockeel. 2000. Web mining research: A survey. *ACM Sigkdd Explorations Newsletter* 2, 1 (2000), 1–15.
[7] Danai Koutra, U Kang, Jilles Vreeken, and Christos Faloutsos. 2014. VOG: summarizing and understanding large graphs. In *Proceedings of the 2014 SIAM international conference on data mining*. SIAM, 91–99.
[8] Danai Koutra, Joshua T Vogelstein, and Christos Faloutsos. 2013. Deltacon: A principled massive-graph similarity function. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 162–170.
[9] Saket Navlakha, Rajeev Rastogi, and Nisheeth Shrivastava. 2008. Graph summarization with bounded error. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. ACM, 419–432.
[10] Mark EJ Newman. 2004. Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems* 38, 2 (2004), 321–330.
[11] Caleb C Noble and Diane J Cook. 2003. Graph-based anomaly detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 631–636.
[12] Shashank Pandit, Duen Horng Chau, Samuel Wang, and Christos Faloutsos. 2007. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 201–210.
[13] Jorma Rissanen. 1978. Modeling by shortest data description. *Automatica* 14, 5 (1978), 465–471.
[14] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*. ACM, 851–860.
[15] Hassan Sayyadi, Matthew Hurst, and Alexey Maykov. 2009. Event detection and tracking in social streams. In *In Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2009). AAAI*.
[16] Neil Shah, Danai Koutra, Tianmin Zou, Brian Gallagher, and Christos Faloutsos. 2015. Timecrunch: Interpretable dynamic graph summarization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1055–1064.
[17] Ellen Spertus, Mehran Sahami, and Orkut Buyukkokten. 2005. Evaluating similarity measures: a large-scale study in the orkut social network. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. ACM, 678–684.