Visual Analytics of Time Evolving Large-scale Graphs

Raju N. Gottumukkala, Siva R. Venna and Vijay Raghavan

***Abstract*—Several real-world observations from streaming data sources, such as sensors, click streams, and social media, can be modeled as time-evolving graphs. There is a lot of interest in domains such as cybersecurity, epidemiology networks, social community networks, and recommendation networks to both study and build systems to track the evolutionary properties of graphs. However, the size and complexity of these graphs present several challenges in terms of processing, analyzing, and visualizing this data. This paper provides a conceptual introduction to time evolving graphs and discusses state-of-the-art techniques and tools for analyzing and visualizing massive time evolving graphs. A visual analytics sandbox implementation architecture and some ongoing projects in this area are also discussed.**

***Index Terms*— visual analytics, time evolving graphs, data streams, graph visualization**

I. Introduction

Big data technologies are radically transforming the pace at which knowledge is created from data. Organizations are looking to leverage these technologies to collect and process real-world datasets to make decisions that reflect ground realities. Large scale graphs with billions of nodes and edges created from real world observations are emerging in multiple domains and disciplines – these include social community networks [1] infrastructure networks [2], epidemiology networks [3], IP traffic networks [4], etc. The dynamics and evolution of these graphs can be captured by introducing time dimension - this introduces additional complexity for organizations to track all the graph properties with respect to their evolution. In literature these graphs are frequently mentioned as Time Evolving Graphs (TEG), Time Varying Graphs, or Temporal Graphs. These graphs are also closely related to dynamic graphs. The key distinguishing feature of evolutionary graphs compared to dynamic graphs is that the graph topology also changes and these changes are significant.

The effects of time varying topologies on various dynamic processes in networks is an increasing subject of interest with big data [5][6][7]. Example applications are the spread of information, infectious diseases, and malware.

Raju Gottumukkala is Director of Research in Informatics Research Institute and Site Director of Center for Visual and Decision Informatics, Siva R. Venna is Ph.D. student, Vijay Raghavan is Professor in School of Computing and Informatics and Director of Center for Visual and Decision Informatics, from University of Louisiana at Lafayette.
e-mail:
raju@louisiana.edu,vennasivaram@gmail.com,vijay@cacs.louisiana.edu

The time dimension also introduces new properties to the graph to study how the node, edges, subgraphs, or particular graph properties evolve over time. As such, the data models of time-evolving graphs are much more complex to manage compared to key-value stores, column stores, relational SQL, or document stores. Also, the computational and visualization requirements to manage these tools far exceed the capabilities of commercially available tools. The evolutionary aspect of graph has been studied in many applications [5][13][14]. These include studying travel patterns, disease epidemics, and human interactions from *human and animal proximity networks* constructed from sensors, cell phone, RFID or GPS devices, understanding *co-authorships and citation networks* to predict future collaborations, predicting vehicle traffic from *transportation networks*, understanding evolution of events of interest from *social media graphs*, understanding *malware and network traffic* anomalies from internet traffic, understanding disease spread in cancer cells from gene networks contain information on protein and DNA information, etc. Existing tools provides for summarization or provide aggregate statistics on these time evolving graphs. Performing any analytics or visualizing evolutionary patterns beyond these basic operations is very labor intensive and time consuming

This paper presents some background on time evolving graphs, and how visual analytics processes can assist in managing these graphs. The paper also discusses the state of the art analytics, visualization and user interaction techniques and tools available for knowledge discovery in graphs. We also present our ongoing work in building a big data sandbox for visual analytics, and discuss ongoing big data projects that use time evolving graphs.

II. Time Evolving Graphs

A. Definition

A time-evolving graph is defined as a graph G= (V, E, T), where V is the set of vertices (or nodes), E is the set of edges, and T is the set of time instants. Also, $E \subseteq (V \times T \times V \times T)$ is the set of edge. An edge $e \in E$ is defined by e = (v1, ta, v2, tb), where v1, v2 $\in$ V are the origin and destination nodes and ta, tb $\in$ T are origin and destination time instants. e = (v1, ta, v2, tb) is basically a directed edge from node v1 at time ta to node v2 at time ta. An undirected edge can be represented when E has both (v1, ta, v2, tb) and (v1, tb, v2, ta). The usage of the definition was introduced in [11]. Evolution could represent the variation of availability of a node, edge, or a graph. Detailed definition of time evolving graphs and evolutionary properties of these graphs are discussed in [11][12].

B. *Data model:*

The most straightforward method to store a graph is in the form of an adjacency list, or an adjacency matrix. There are multiple ways to store a time evolving graph while preserving a temporal structure of the graph. Choosing the right data model depends on the nature of the data, the type of graph (strongly connected, vs. weakly connected, sparse, or dense graphs, etc.) and the targeted data processing and analytical tasks.

The most straightforward approach is to store a snapshot of the graph for time instance (shown in Figure 1(a)) [15]. This model consumes a lot of memory, and works only when it is not necessary to capture relationships between nodes across time-stamps. Also, running certain queries across time-stamps is inefficient. Other ways include, creating a single graph for all time stamps and storing the time information on the edge as an attribute. This can be accomplished in two different ways as shown in Figure 1(b) as a simple list of timestamps [6] or as in figure 1(c) by specifying limits when edges are persistent during sequences of timestamps [7]. For example, in Figure 1(b), the edge between node A and node B is available at time-stamps 6, 7 and 11. In Figure 1(c), the edge from node A to node B is available at time stamps 1, 2 and 3. One of the limitations of these models is that the relationship between the nodes across time-stamps cannot be stored. More complex TEGs where there is possibility of having edges across nodes from different timestamps, one way of storing such graphs is creating duplicates of nodes for each timestamp it is present in and adds edges between required nodes as shown in Figure 2 [12].

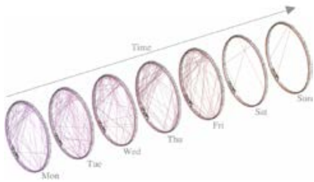

Figure 1(a)

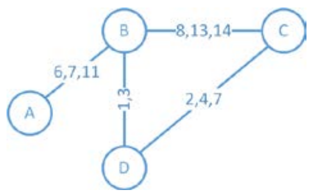

Figure 1(b)

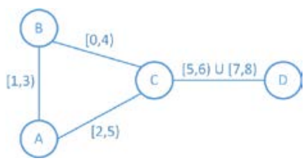

Figure 1(c)

Figure 1. Three ways of storing a time-evolving graph

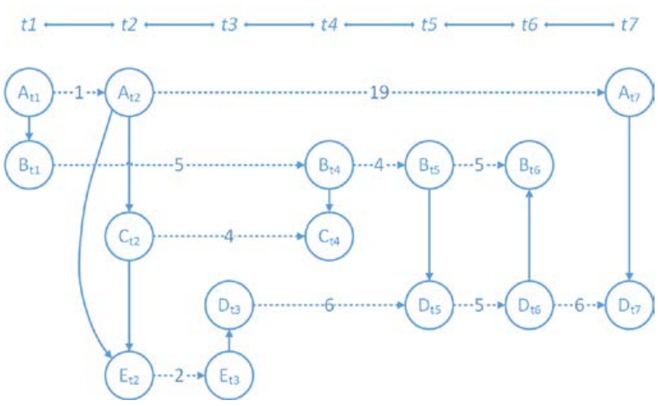

Figure 2. Graph representation of TEG where edges are present across nodes from different time stamps

III. Visual Analytics Process for Time Evolving Graphs

One of the key distinguishing features of visual analytics, as compared to emerging areas such as automated analysis, is the integration of visualization and human's visual exploration components into analytics.

A. *What is Visual Analytics?*

According to "Illuminating the Path" by Thomas and Cook [16], Visual analytics is an interdisciplinary field that integrates the following areas: *analytical reasoning* approaches that let users obtain deep insights that directly support assessment, planning and decision making, *visual representations and interaction techniques* that exploit the human eye's broad bandwidth pathway into the mind to let users see, explore, and understand large amounts of information simultaneously, *Data representations and transformations* that convert all types of conflicting and dynamic data in ways that support visualization and analysis, techniques to support production, presentation, and dissemination of analytical results to communicate information in appropriate context to a variety of audiences. Figure 3 shows the visual analytics process as an integrated framework that has data, models, knowledge, and visualization interaction process interact with each other.

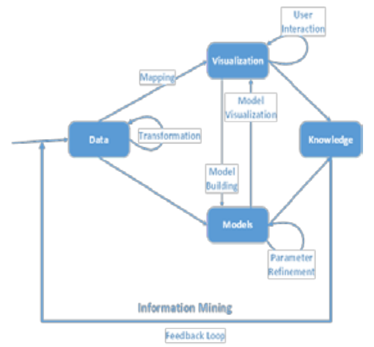

Figure 3. Visual analytics as an integrated framework [17]

Visual analytics has become a buzz word in the business intelligence domain, and many companies including SAS™, IBM SPSS™, considered leaders in statistical analysis for business are pursuing novel ways to improve their data presentation through new products such as SAS Visual Analytics™, and IBM's Many Eyes™. Moreover, several new BI tools such as Tableau™, Birst™ and Google Fusion Tables™ also provide various interactive visualization capabilities. While these tools provide some basic visualization and interaction capabilities for users to interact with the data, these tools are far from promoting analytics discourse with the visualization environment. The overarching vision of visual analytics is to provide technology that combines the strengths of human and electronic processing [16][17].

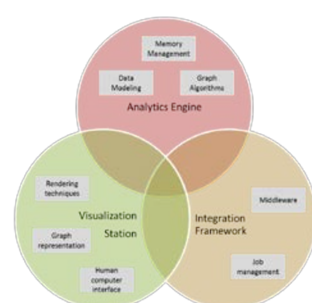

Figure 4. Visual Analytics as an integration framework with various components

Most of the existing research has focused on graph theoretical representation of time-evolving graphs, visualization of dynamic aspects of time-evolving graphs, and interaction techniques and tools to interact with these graphs. A recent comparative study on the landscape of various open-source and commercially available BI platforms [18], and state-of-current research in visual analytics capabilities of BI systems highlight the capabilities and limitations with respect to individual components, i.e. data management, automated analysis, visualization, and system architecture. Another survey paper on visual analytics [19] also highlights the state-of-the-art in visual analytics and the challenges in individual research areas. However, the visual analytics solutions actually lie in the integration of various research areas, and optimization of, data management, analytics, visualization, and human interaction modules. All these business intelligence tools have visual analytics capabilities added into the existing platform, hence offer limited flexibility to support visual analytics of complex, and real-time datasets. Bridging these disciplines into an integrated framework offers new opportunities for researchers to experiment with different visual analytics components to improve the overall end-user experience to manipulate the information.

Visual analytics framework is an integration of various components, (refer Figure 4) namely (1) An efficient data model and memory management to store, and run graph mining algorithms, (2) interaction techniques based on a touch interface to manipulate the graphs with respect to its dynamics, and (3) an integration framework that facilitates seamless interaction of graph datasets. Not all the graph operations can be performed on the visualization system; hence there should be seamless communication between the visualization system and the analytics server. The middleware serves as the key interface between the visualization system and the analytics server. The middleware takes care of management and prioritization of various jobs, and translation of users' actions into analytical queries

IV. Graph Analytics Engines

Three is a great demand for close to real-time analysis of massive graphs - given the demand in several real-time applications (online recommendations for click stream processing, fraud detection, analysis of cyber-attack graphs, etc.). The performance of a graph analytics engine is affected by three important factors, the graph data model, the memory management / caching scheme, and the graph analytics algorithms.

A. *Data management*:

One of the most important elements of the graph database is the data model (or the database model), which is basically the data structures for schema and instances modeled as graphs or generalizations of them – to support efficient way to store and query, index, or aggregate data. The data can be centralized and distributed that either store graphs in the memory, or store them on the disk and retrieve them on demand.

Graph access patterns have very poor spatial memory locality and this result in large amounts of random memory access. High throughput processing of massive graphs that may not fit in the main memory require efficient memory management that includes efficient caching strategies to write unused data to disk, indexing mechanisms for efficient retrieval of these graphs. Storing and managing graph data on disks suffers from very poor I/O latency, and it is not possible to store the entire graph in the memory. Solid State Drive (SSD)'s are also an efficient way to store or cache the data. Most of the graph databases provide some basic cache management and indexing schemes, which may not be optimal for all types of graphs, or graph operations. The strategies for storing the graph in a single or distributed nodes, the dynamic nature of the data (bursty, or highly dynamic), the topology of the graph, the type of processing that needs to be done, etc.

One of the most widely used graph database Neo4j [20] for example stores the graph on the disk, and retrieves them into the main memory for computation. FlockDB from Twitter, RDF based AllegroGraph [21], and Objectivity's InfiniteGraph [22] are all well-known distributed databases than can support storing node or edge labels as temporal attributes. The choice of graph database depends on the requirements of the application and graph type. This includes storing features (main memory, external storage, indexing), the graph structures to store temporal attributes (either on nodes, edges, or graphs) for efficient retrieval.

B. *Graph Analytics*

TEGs evolve over time as new edges or nodes are added while some old ones vanish and it is important to understand and extract patterns following these evolutionary changes. The complexity of these algorithms depend on the speed of evolution of these graphs (1) slowly evolving graphs are those where the substantial changes occur on a large time scale of days or weeks e.g. web based networks, citation graphs etc. and (2) streaming or fast evolving graphs where overall graph structure changes very rapidly in matter of seconds e.g. social media graphs, transportation networks etc. [12][13][14][23]. Based on the domain and data at hand different analysis models can be built and a brief summary of some high level tasks are explained below:

Table 1. A summary of graph analysis operations

Type of Temporal Characteristic	Graph Operations
Temporal network topology & structure	Degree, connectivity, density
Reachability analysis	Paths, walks, trails
Predicting network topological properties	Link prediction & classification
Detecting outliers	Node or edge clustering
Node neighborhoods and communities	Persistent patterns & motifs

V. VISUAL REPRESENTATION & USER INTERACTION

The overall goal of visualization is to enable users to obtain insights from data. Given the scalability and dynamic nature of time evolving graphs – visualization needs to take into account how much information can be perceived and understood, computed and displayed. These include the graph topology to be projected (graph representation) into 2D or 3D space using different layout schemes, understanding what interaction and human computer interface tools are best suitable, and how to render data efficiently on display screens. These factors are common for visualization of any large-scale multivariate graphs.

In the case of visualizing TEG's, users need to understand changes in the graph (the temporal aspect) in terms of the nodes, edges or the subgraph, their topological structure, or graph characteristics The most common way to represent dynamic graphs are animated diagrams, or static graphs with a timeline.

A. *Animation-based visualization:*

When no interaction or manipulation of graphs is desired, a simplest way to show temporal evolution is through animation. A graph is constructed by creating an animation from a series of graphs at different time stamps. An initial supergraph layout is created to have a consistent layout for graphs in multiple time-stamps.

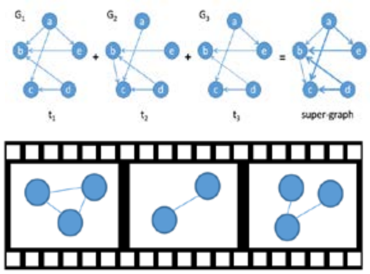

Figure 5 (a). Constructing a super graph from 3 sequential dynamic graphs, (b) A simple animation for a small TEG

Generally a super graph is constructed using the graphs from considered time stamps and a single graph layout is computed as shown in Figure 5 [19]. There are several variations of animation based approaches that represent the time transitions using color coding, shape, or layout techniques that were covered in [19].

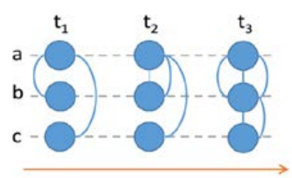

Figure 6(a). Juxtaposed node-link based timeline presentation

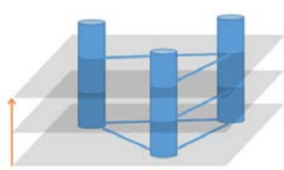

Figure 6(b). Super-imposed node-link approach with layers representing time steps

B. *Timeline-based visualization:*

Another way (and the most common way) to display temporal evolution is by projecting time into space dimension. This can be done multiple ways by juxtaposed node-link presentation over time (refer Figure 6(a)), or superimposed nodes and links where layers are used to represent time-stamps (refer Figure 6(b)). The temporal changes in graphs may also be represented by matrix-based approaches which are better suitable for more readability. The matrix notation provides the ability to encode dynamic changes in the cell and an edge using colors, and charts. For example, Figure 6(c) uses super-imposed node-link approach to show different forms of intracell timeline representation.

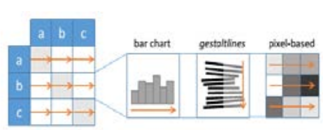

Figure 6(c). Intra-cell timelines in a matrix representation

For visualizing TEGs it is important to choose a good visual representation to show them in a presentable and understandable format and these visual representations should reduce visual clutter and minimize temporal aliases for node positions across time, maximize readability and scalability. Selecting a visual representation for TEGs is restricted by the data at hand, the size of the graph, amount of data to visualize, purpose of the visualization etc. Some of the visual representations are limited by graph layouts as it is difficult to find automatic layouts for static graphs and to do that for every time stamp is an enormous task [18][20]. Other extensions to the animation-based and timeline-based visualization techniques include 3D visualization [24][25], hybrid representations combining animation and timeline drawings. Several application specific visualization techniques are available in literature – these include time line trees [26], tree maps [27], icicle plots [28], node link diagrams with time series [29], time arc trees [30], etc.

C. *Human Computer Interaction*

The Human Computer Interaction (HCI) enables users to browse the data set with set of interactions using a human computer interface to discover hidden insights on the data. An effective HCI is equally important as visual representation for a good VA framework. These HCIs should enable the user to have control over what and how they want to see and to define the flow and parameters of decision informatics. Recent studies [31][32][33] provided taxonomies for visual interactions techniques to help better understand and improve VA designs. Interactions with the visual representations are divided into three high level categories:

Data and view specifications: HCI should allow the user to reconfigure the views based on attributes of interest, to filter portions of the graph, to derive simple analytics using statistical computations.

View manipulations: User should be able to select, highlight and bookmark portions of the graph by either manual selection or through search criterion, to navigate and explore over graphs using zooming, magic and fish eyed lenses, panning etc., should allow the user to coordinate and organize multiple views for easy comparisons of results from different interactions.

Process and Provenance: VA systems should record different interactions for fast recall or revisiting of past analyses, they should also support multi-user-collaboration, reporting, and sharing of views, interactions and results.

The other important aspect of visualization is rendering the graph to display large scale datasets. GPU based rendering is becoming increasingly common. Gephi provides a timesliding based tool to navigate a time-varying graph. There are several other rendering techniques for multi-variate graphs that can be applied for time-varying graphs. There are several graph visualization libraries and tools available for use. There are several network visualization tools available for use. The choice of tools depends on the size, scale, the nature of the graphs, the type of analysis (flow-based, relationships, clusters, cliques), and the platform for visualization (desktop or web browser). Some of the widely used visualization desktop visualization tools are Gephi,[34] Cytoscape [35], Palantir [36], and Dato (GraphLab) [37]. There are also several web-based visualization libraries that include D3.js [38], Sigma.js [39], and Vivagraph.js [40]. A more detailed list of visualization tools are available in [41].

VI. A Sandbox Implementation of Real-Time Visual Analytics Platform

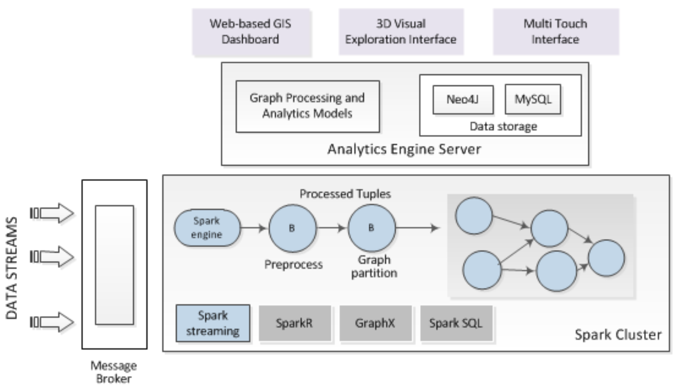

Figure 7. A Reference Visual Analytics Sandbox Implementation

Component	Purpose	Methods
Data broker	Integration and distribution of different data streams	Social media streams, Internet traffic streams, Sensor network streams etc.
Online data preparation	Collect data and prepare for pre-processing	Data collection, integration, normalization, representation (schema) etc.
Distributed pre-processing	Improve quality of the data	Data cleaning, correction, transformation, dimensionality reduction etc.
Batch processing	Generate and clean graphs	Graph generation, pruning, clustering, transformation etc.
Analytics Engine	Graph processing	Graph querying – topology, paths, walks, persistent patterns, motifs, link classification and prediction etc.
Visual processing	Prepare graphs for visualization	Layout computation, visual representation
Visual interface	User interaction	Web based, 3D exploration and Multi touch interfaces

Table 2. Description of various tasks performed by different components of the Visual Analytics Sandbox

The visual analytics sandbox environment is an experimental infrastructure for developing novel integrated data stream management, analytics and visualization algorithms for multiple application domains.

The big data system architecture (refer Figure 7) provides an end-to-end implementation of a system that consumes data streams, constructs graphs and updates the TEG stored in the graph database based on new incoming data streams. The dynamically updated TEG can be accessed by a browser, a 3D environment, or a multi-touch interface. The message broker receives data streams from multiple real-time sources, integrates these streams and sends them to a spark cluster. The spark cluster does initial pre-processing in terms of extracting relevant information, reducing the dimension of the graph. A graph is constructed for every time window. The transformed graph is loaded into an in-memory graph database. The temporal information about nodes and edges are updated in the new transformed graph. The graph can be queried from a visual interface. The queries include basic node and edge based statistics, to mining motifs, cliques, and other persistent graph patterns. The visual interface has various libraries for multiple devices.

VII. Case Studies

A. *Real-time Forecasting of Influenza*

Influenza is one of the major causes of deaths throughout the world and is the top medical reason for Emergency Department (ED) visits. In this case study, we aim to forecast flu counts using historical data from heterogeneous data sources that includes electronic records Google Flu Trends (GFT) data and other environmental variables like Temperature, Precipitation, humidity etc. These datasets are integrated to create a graph-based model to forecast influenza across different geographical locations. The results of the flu prediction model are available for viewing both on a browser and multi-touch interface.

B. *Link prediction*

Link prediction is a widely used social network analysis tool. Link prediction has wide range of applications such as identifying missing information, identifying spurious interactions, and studying the evolution of a network. In ecommerce, link prediction is used for building recommendation systems; and in bio-informatics, it is used to predict protein-protein interactions.

A supervised method to predict unknown association of medical concepts, using bio-medical publication information from Medline [43], is proposed and evaluated. Medline is a National Institute of Health (NIH)'s citation database with more than 21 million publication citations. A temporal series of concept networks are generated using relevant medical concepts extracted from these publications, by segmenting the data over multiple time snapshots. In a concept network, each node represents a bio-medical concept and an edge between two nodes represents relationship that two medical concepts that co-occurred in at least in one publication. The document frequency of a given concept is the weight of the node and the co-occurrence frequency of two concepts is the weight of the edge connecting them. Now, the link prediction problem is formulated as a process of identifying whether a pair of concepts, which are not directly connected in the current duration concept network, will be connected directly in the future. A concept pair is labeled positive if a direct connection occurs in a future time snapshot; otherwise, the pair is negative. For each concept pair in the labeled data set, a set of topological features (random-walk based and neighborhood-based) is extracted from the current snapshot of the concept network. Supervised classification algorithms, such as SVM, and C4.5 decision tree are used to generate prediction models. The experimental evaluations show that the performance of our approach is in the range of 68 – 72%, in terms of classification accuracy, recall and precision.

C. Social Media: Detecting Emerging Events

Many events happen every day across the world and people often comment on events in real time, with thousands of tweets posted in real time. Prominent examples for this include the US Airways plane clash on Hudson and bombings at Boston marathon. There are also other types of user generated content, such as microblogs, catering to users communicating among each other or sharing information. The goal of new event detection (also referred to as event detection) is to identify the first story to detect a particular event. It is beneficial to identify these stories and report on them as soon as possible. This implies that we need to process the data in real-time, since these microblogs are faster and more up to date compared to traditional news stories. Prior works done on event detection on Twitter domain either perform a post-hoc analysis of tweets and detect events that have already happened or use domain-specific knowledge to identify events. Hence, most of these methods are unable to detect a broad range of events within near real time.

We developed a domain independent event detection model, which can detect an event, typically, within 4-8 minutes after the event is first mentioned and can track it in real time. A graph based approach is employed, where each node is an individual token that appeared in a tweet and edges represent the co-occurrence frequencies of the tokens. The detection task is accomplished by conducting the following four steps. First, a fast and efficient divergence model is used to identify unusual activity in the usage of words. Second, we build a co-occurrence graph around those words with unusual activity. Third, candidate events are extracted from the graph using a combination of fast and efficient graph pruning techniques and a graph clustering method. Fourth, spurious clusters (non-events) are eliminated via an event evolution model, which requires candidate events to be discussed for certain duration of time before being considered a real event. Evaluation of our approach, compared to similar work [44], shows that the proposed method detects a greater percentage of known true events and a greater number of true events. Moreover, events are detected earlier.

VIII. Research Challenges

Visual analytics of TEG is an emerging discipline. Given the demand and challenges, there are several emerging paradigms in data management, analysis and visualization aspects of these graphs. Below are some of the research challenges.

Scalability: The growing dynamic data is pushing the size of these graphs, and this naturally introduces challenges in every stage of visual analytics, pre-processing, graph loading, mining, and visualization. The layout algorithms for visualizing the large scale graphs need to take the dynamic evolution of these graphs – which is a hard problem because it is hard to estimate the layout of the graphs in domains such as social media where topic and event evolution are so rapid and unpredictable. Better user controlled graph simplification approaches are needed for filtering, sampling, and aggregation of the graphs.

A. Graph processing and interaction: Most of the existing graph processing and visual analytics techniques employ black box techniques where the user has no knowledge or control over the analysis process. System should be designed to allow the user to guide and control the parameters during the analysis.

B. Perception in visualization: Human perception plays a major role in visualization of TEGs as it supports the cognitive associated process. Visualizations should support exploration and stimulate the capabilities of human visual system.

C. In situ analysis: Traditional approaches for storing the data into secondary storage and analyzing later are not feasible with large scale TEGs, especially for fast evolving graphs. Visual analytics system should explore the idea of in situ analysis and process the data as much as it can while still the data is in memory. Major challenge to address with in situ analysis is to effectively share computing resources and collaborate with overall process flow and other user interactions [42].

D. Parallel Algorithms: To be on pace with the ever increasing size of graphs and their evolution speed, parallel processing should be explored. As computing resources are getting cheaper and equipped with multiple cores it is necessary to redesign most of the graph processing and visualization algorithms to support parallel processing.

E. Applications: Designing graph visual analytics frameworks that adapts fast across different application domains is necessary as each application has specific analysis focus and data type. Building such an integrated unified visual analytics framework for TEGs is a difficult task.

F. Availability of APIs and other development libraries: Lack of resource libraries supporting integrated visual analytics for TEGs hinders the rapid application development in this scenario. Most of the graph algorithms are designed to support static graphs and some of these have limitations while adapting for TEGs and in most cases needs to developed from scratch which is time consuming and costly.

IX. Conclusion

The work presented in this paper is the state-of-the-art in visual analytics of time evolving graphs – which is a growing and active discipline given the explosion of datasets arriving from real-world sources. We first cover background definition of Time evolving graphs, how they are represented, stored and visualized, then we discuss visual analytics as a framework for time evolving graphs. Then we discuss sine ongoing research and tools available for building analytics engine component. Then various visualization techniques and tools for visualization and interaction are discussed. Reference visual analytics sandbox architecture is presented that is currently being developed by the authors. Various research case studies are presented that leverage this sandbox. Finally research challenges are presented.

X. Acknowledgements

This material is based upon work supported by: NSF Grant No.1429526 and NSF Grant No. 1160958

XI. References

[1] Greene, Derek, and Pádraig Cunningham. "Producing a unified graph representation from multiple social network views." Proceedings of the 5th Annual ACM Web Science Conference. ACM, 2013.
[2] Scott, John. Social network analysis. Sage, 2012.
[3] Danon, Leon, et al. "Networks and the epidemiology of infectious disease." Interdisciplinary perspectives on infectious diseases (2011).
[4] Iliofotou, Marios, et al. "Graption: Automated detection of P2P applications using traffic dispersion graphs (TDGs)." University of California, Riverside Report, UCR-CS-2008096080 (2008).
[5] Holme, Petter. "Modern temporal network theory: a colloquium." The European Physical Journal B 88.9 (2015): 1-30.
[6] Holme, Petter, and Jari Saramäki. "Temporal networks." Physics reports 519.3 (2012): 97-125.
[7] Casteigts, Arnaud, et al. "Time-varying graphs and dynamic networks." International Journal of Parallel, Emergent and Distributed Systems 27.5 (2012): 387-408.
[8] Cardillo, Alessio, et al. "Evolutionary dynamics of time-resolved social interactions." Physical Review E 90.5 (2014): 052825.
[9] Tang, John, et al. "Exploiting temporal complex network metrics in mobile malware containment." World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a. IEEE, 2011.
[10] Dong, Yuxiao, et al. "Link prediction and recommendation across heterogeneous social networks." Data Mining (ICDM), 2012 IEEE 12th International Conference on. IEEE, 2012.
[11] Wehmuth, Klaus, Artur Ziviani, and Eric Fleury. "A unifying model for representing time-varying graphs." arXiv preprint arXiv:1402.3488 (2014).
[12] V. Kostakos, "Temporal graphs," Physica A: Statistical Mechanics and its Applications, vol. 388, no. 6, pp. 1007–1023, Mar. 2009
[13] Aggarwal, Charu, and Karthik Subbian. "Evolutionary network analysis: A survey." ACM Computing Surveys (CSUR) 47.1 (2014): 10.
[14] Wehmuth, Klaus, Artur Ziviani, and Eric Fleury. "Model for Time-Varying Graphs." Workshop on Dynamic Networks. 2013
[15] Tang, John, et al. "Analysing information flows and key mediators through temporal centrality metrics." Proceedings of the 3rd Workshop on Social Network Systems. ACM, 2010.
[16] Cook, Kristin A., and James J. Thomas. Illuminating the path: The research and development agenda for visual analytics. No. PNNL-SA-45230. Pacific Northwest National Laboratory (PNNL), Richland, WA (US), 2005.
[17] Keim, Daniel A., et al., eds. Mastering the information age-solving problems with visual analytics. Florian Mansmann, 2010.
[18] Von Landesberger, Tatiana, et al. "Visual analysis of large graphs: state-of-the-art and future research challenges." Computer graphics forum. Vol. 30. No. 6. Blackwell Publishing Ltd, 2011.
[19] Beck, Fabian, et al. "The state of the art in visualizing dynamic graphs." EuroVis STAR (2014).
[20] Miller, Justin J. "Graph Database Applications and Concepts with Neo4j." Proceedings of the Southern Association for Information Systems Conference, Atlanta, GA, USA March 23rd-24th. 2013.
[21] Aasman, Jans. Allegro graph: RDF triple database. Technical report. Franz Incorporated, 2006.url:http://www. franz. com/agraph/allegrograph/(visited on 10/14/2013)(cited on pp. 52, 54),
[22] InfiniteGraph: The Distributed Graph Database, a performance and distributed performance benchmark of InfiniteGraph and a Leading Open Source Graph Database using synthetic data, 32 Infinite Graph, white paper from Objectivity, http://www.objectivity.com/wpcontent/uploads/Objectivity_WP_IG_Distr_Benchmark.pdf, 2012.
[23] Pienta, Robert, et al. "Scalable graph exploration and visualization: Sensemaking challenges and opportunities." Big Data and Smart Computing (BigComp), 2015 International Conference on. IEEE, 2015.
[24] Archambault, D., T. Munzner, and D. Auber. "Visual exploration of complex time-varying graphs." Visualization and Computer Graphics, IEEE Transactions on 12.5 (2006): 805-812.
[25] Gaertler, Marco, and Dorothea Wagner. "A hybrid model for drawing dynamic and evolving graphs." Graph Drawing. Springer Berlin Heidelberg, 2006.
[26] Burch, Michael, Fabian Beck, and Stephan Diehl. "Timeline trees: visualizing sequences of transactions in information hierarchies." Proceedings of the working conference on Advanced visual interfaces. ACM, 2008.
[27] Hao, Ming C., et al. "Importance-driven visualization layouts for large time series data." Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on. IEEE, 2005.
[28] Tekušová, Tatiana, and Tobias Schreck. "Visualizing time-dependent data in multivariate hierarchic plots-design and evaluation of an economic application." Information Visualisation, 2008. IV'08. 12th International Conference. IEEE, 2008.
[29] Saraiya, Purvi, Peter Lee, and Chris North. "Visualization of graphs with associated timeseries data." Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on. IEEE, 2005.
[30] Greilich, Martin, Michael Burch, and Stephan Diehl. "Visualizing the evolution of compound digraphs with TimeArcTrees." Computer Graphics Forum. Vol. 28. No. 3. Blackwell Publishing Ltd, 2009.
[31] Kerren, Andreas, and Falk Schreiber. "Toward the role of interaction in visual analytics." Proceedings of the Winter Simulation Conference. Winter Simulation Conference, 2012.
[32] Heer, Jeffrey, and Ben Shneiderman. "Interactive dynamics for visual analysis." Queue 10.2 (2012): 30.
[33] Yi, J.S., Y. a. Kang, J. Stasko, and J. Jacko. "Toward a deeper understanding of the role of interaction in information visualization". IEEE Transactions on Visualization and Computer Graphics 13(6) 1224-1231, 2007.
[34] Gephi: http://gephi.github.io/ (accessed - 17 November 2015)
[35] Cytoscape: http://www.cytoscape.org/ (accessed - 17 November 2015)
[36] Palantir: https://www.palantir.com/ (accessed - 17 November 2015)
[37] Dato (GraphLab): https://dato.com (accessed - 17 November 2015)
[38] D3.js: http://d3js.org/ (accessed - 17 November 2015)
[39] Sigma.js: http://sigmajs.org/ (accessed - 17 November 2015)
[40] VivaGraph.js: https://github.com/anvaka/VivaGraphJS (accessed - 17 November 2015)
[41] http://www.kdnuggets.com/2015/06/top-30-social-network-analysis-visualization-tools.html (accessed - 17 November 2015)
[42] Wong, Pak Chung, et al. "The top 10 challenges in extreme-scale visual analytics." IEEE computer graphics and applications 32.4 (2012): 63.
[43] MEDLINE: http://www.ncbi.nlm.nih.gov/pubmed/ (accessed - 17 November 2015)
[44] M. Cataldi, L. Di Caro, and C. Schifanella, "Emerging topic detection on twitter based on temporal and socialterms evaluation," in Proc. of the Tenth International Workshop on Multimedia Data Mining, 2010, p. 4.