This chapter appeared in "Matthias Dehmer, Frank Emmert-Streib, Stefan Pickl, Andreas Holzinger (Editors): Big Data of Complex Networks, pp.293-312, Chapman and Hall/CRC, 2016" – ISBN 9781498723619

http://www.crcpress.com/Big-Data-of-Complex-Networks/Dehmer-EmmertStreib-Pickl-Holzinger/9781498723619

Note that the final formatting is slightly different in print.

Chapter 12

Visualizing Life in a Graph Stream

James Abello, David DeSimone, Steffen Hadlak, Hans-Jörg Schulz, and Mika Sumida

12.1 Abstract

We introduce a simple and useful view for observing graph streams. They are viewed as collections of edge events where each edge has associated a set of time-dependent statistics that include firing rate, recency, and persistence. The activity rate of any subgraph is expressed as an aggregation of its corresponding edge statistics. Salient subgraphs are detected by isolating through

time those edges whose activity rate deviates substantially from the activity rate of the entire stream. These salient subgraphs exhibit some peculiar "herding" and "straying" behaviors that are humanly interpretable. The vertices involved in the creation of these salient behaviors cover a substantial portion of the entire graph stream. This coverage can be subject to both human and computer verification. All our computations are incremental and are accompanied by a visualization platform that integrates dynamic node link views of "recent" graph substreams with a tape view of the Top-K edge statistics to provide a compact overview of the graph stream. This platform has also been coupled to our modular Degree-of-Interest system for a closer investigation of those patterns found in the overview. We use Twitter data to illustrate our tools, but our approach is by no means confined to microblog data.

12.2 Introduction

When exploring a data stream it is natural to ask how to relate current stream snapshots to past snapshots. Depending on the data semantics and the task at hand different interpretations are possible. For example, in the case of microblog data (like Twitter) making sense of conversations and discussions related to a particular topic may entice users to join the discussion. For data analysts, a usual task is to discern how tweets information patterns spread with the possible goal of intuitively explaining their findings. In monitoring traffic scenarios, teasing out those communication patterns that deviate from a considered normal behavior can be used as proxies for intrusion detection. In general social networks, identifying influential nodes in a "volatile" graph stream is of considerably interest. We report here a useful approach to identify trends and exceptional nodes in a graph stream. The fundamental idea is to view a graph stream as a collection of "elementary" time-stamped events whose aggregation through time generates "salient" patterns whose activity rate is incrementally maintained. We are able to isolate group "herding" and "straying" as peculiar behaviors that can be subject to both human and computer verification. We quantitatively estimate the overall behavior of the detected salient edges as a convex combination of their "herding" or "straying" tendencies and their firing rates and recency "profiles". All our computations are accompanied by a visualization platform that integrates dynamic node link views of "recent" subgraphs with a tape view of their Top-K edge statistics (Figure 12.1). The approach discussed here has been coupled with a Degree-of-Interest (DoI) based exploration system [2] to provide the user with the functionality to take a closer look at particular keywords of interest identified with our novel approach. Currently, such DoI-based systems are not equipped to operate on the graph streaming setting proposed here.

Our main contributions are:

- An adaptive and simple approach to graph stream processing that is based on the "firing rates" of edge co-occurrences.
- Use of the notion of "recency" as a mechanism to measure the decay of an edge or vertex in a graph stream.
- Isolation of "herding" and "straying" patterns in a graph stream as proxies of group behavior.
- Quantification of the notion of "persistent" and "statistically salient" behaviors in a graph stream which can be automatically verified by a human or a computational agent. This is possible by the incorporation of an incremental maximal matching where the matching edges are weighted by the sum of the overall frequency of their endpoints.
- Incorporation of visual cues that correspond directly to the notion of "recency" and "firing rates". This is facilitated by coupling a force directed node link layout with an intuitive Top-K tape representing the most "salient" graph stream elements in a dynamically adjusted time window.

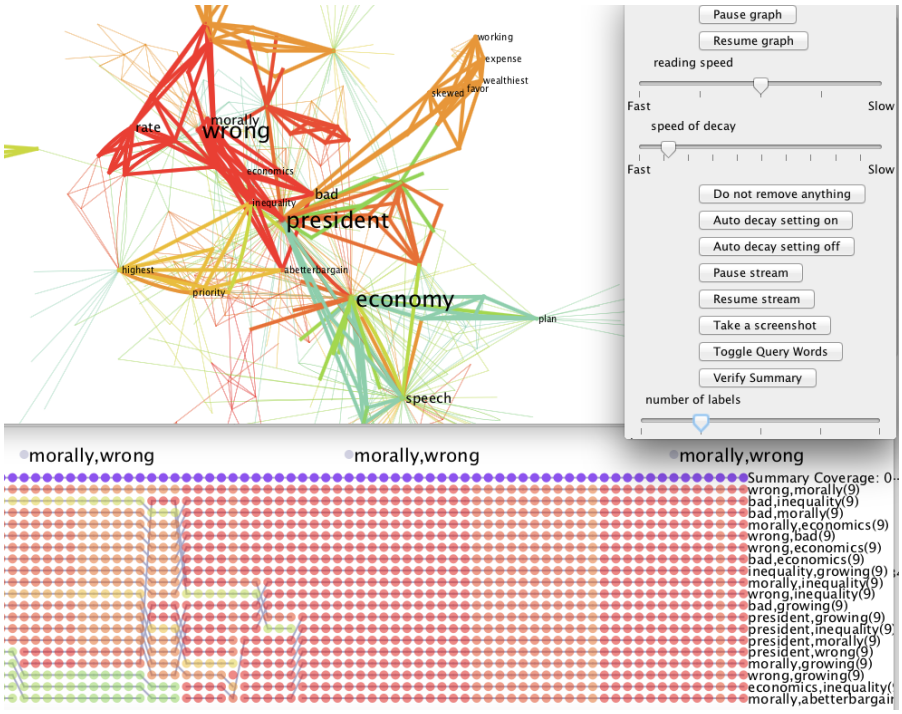

FIGURE 12.1: Complete view of the graph stream visualization interface featuring tweets from President Obama's speech on the U.S. economy on 07/24/2013

The chapter layout is as follows: Section 12.3 describes related work. Section 12.4 introduces the general data model and the fundamental graph stream statistics on which we base our co-occurrence graph stream processing. They are: recency, firing rate, and persistence. Section 12.5 describes the decay mechanisms used to maintain the most "recent" co-occurrence subgraph and presents the statistical mechanisms to extract the most "salient" edges in the stream. Section 12.6 describes the visualization of the recent subgraph as an animated graph visualization and its most salient edges as a collection of time plots that we call the **Top-K Tape**. Section 12.7 details the life cycle of a vertex in a graph stream. Section 12.8 describes the different states of the "life" of a graph stream edge and how they determine the "life persistence" of those vertices, which have co-occurred prominently in the graph stream. Prominent edge states are: trending, untrending, not trending, herding, and straying. Section 12.9 illustrates the application of our approach to the processing of Twitter data, however this work is by no means confined to microblog data. Section 12.10 illustrates the coupling with the DoI-based system. Finally, Section 12.11 concludes this chapter by outlining possible avenues for further research.

12.3 Related Work

During the last decade, the visualization of dynamic graphs has become a quite active and diverse interdisciplinary research field. Recent surveys of the area [5, 9, 11] discuss in a very comprehensive manner the existing variety of approaches and insights. We refer the reader to these publications for a more in-depth treatment of this area. When analyzing time varying graphs, common approaches are to choose a single point in time in the sense of an animation (e.g., [4]) or to aggregate longer time spans into a super or union graph [8], a static structure that can be more easily visualized. A common concept regarding the use of animation to visualize dynamic graphs is the user's "mental map". Issues concerning the preservation of the "mental map" are discussed in [3, 12, 16]. Layout algorithms taking these issues into account are discussed in [6, Sec.3]. However, these approaches have several limitations. The user has either to inspect each time point individually or loose the temporal context altogether. Alternatives show the structure over multiple time points in a single image to overcome these limitations. They either use small multiples in which the structure is shown individually for each time point [13, 14, 17] or embed a representation of time points into the node or edge representation [18, 19, 20]. As they try to convey every piece of information (all nodes, edges, and time points) they cannot scale to large dynamic graphs.

The research described here can be placed in the context of statistically driven extraction of salient subgraphs with bounded resources. We concentrate

on the formulation of a principled approach based on the novel notions of firing rate and recency distribution. They are the result of viewing graph streams as time co-occurrence graphs from which only recent and Top-K prominent subgraphs are extracted and subsequently visualized. To our knowledge such an approach has not been pursued before.

12.4 General Definitions - Data Model and Statistics

Definition 1 ***Graph Streams as Co-occurrence Graphs.*** *A graph stream, on a set of vertices (or nodes) V, is a collection of time-stamped pairs $\langle (x, y), t_i \rangle$ where x and y are elements of V and t_i indicates a time point when the pair of vertices x and y co-occur. For each edge $e = (x, y)$, we let $T_{e,t}$ denote the set of time points ($t_i \leq t$) in which the pair of vertices x and y co-occur. The cardinality $|T_{e,t}|$ is the frequency of co-occurrence of the edge $e = (x, y)$. We denote the collection of co-occurring node pairs up to time t by $E_t = \{e = (x, y) : T_{e,t}$ is non-empty$\}$, and the corresponding set of vertices V_t.*

Definition 2 ***Firing Rates.*** *Following [1], each edge in E_t has a firing rate: $fr(e, t) = \frac{|T_{e,t}|}{t}$ and its corresponding firing sequence is $fr(e) = \langle fr(e, t) \rangle$. The firing rate of any subset E' of E_t is just the sum of the firing rates of the edges in E' up to time t; and its corresponding firing sequence will be denoted by $fr(E') = \langle fr(E', t) \rangle$. The firing rate of a vertex x (up to a particular time t) is the sum of the firing rates of its incident edges up to that time t. With these conventions, a graph stream is a graph sequence $\{G_t = (V_t, E_t)\}$ with a corresponding firing sequence $\langle fr(E_t) \rangle$. We will refer to $fr(E_t)$ as the firing rate of G_t. Note that firing rates can be thought of being analogous to velocities.*

Definition 3 ***Instantaneous Events.*** *An instantaneous snapshot of a graph stream at time point t is the collection of edges that co-occur at time t. An instantaneous event is a maximal connected subgraph of an instantaneous snapshot. In other words, an instantaneous event at time point t is a maximal connected co-occurrence subgraph.*

Definition 4 ***Average Vertex Firing Rate.*** *The* vertex average firing rate *at time point t is the average firing rate of all vertices in the cumulative graph stream at time t. If there are N_t different vertices in the graph up to time point t,*

$$AFR(t) = \frac{\sum_{i=1}^{N_t} FR(w_i, t)}{N_t}$$

We let $\sigma_v(t)$ denote the standard deviation of the set of firing rates of all vertices in the co-occurrence graph up to time t.

Since the arrival distribution of different edges is in general quite different we keep track of what we call edge (or vertex) **Recency**, which is defined as follows:

Definition 5 ***Edge and Vertex Recency.*** *If e is an edge in the graph stream,*
$Recency(e,t) =$

$$\begin{cases} 1 & \text{if } e \text{ appears in the graph stream at time } t \\ \dfrac{1}{t - t_{last}} & \text{otherwise} \end{cases}$$

where t_{last} = the immediate previous time e appeared in the graph stream. The recency of a vertex is defined in an analogous way.

The aforementioned mathematical framework allows us to formulate questions related to the "behavior" of either vertices or edges in a graph stream $\{G_t\}$ in terms of their associated firing rates. The overall approach consists of comparing the firing sequence of an edge or a vertex with the firing sequence of the graph stream in which they reside. We **selectively label** from the graph stream, incrementally, those edges (or vertices) whose firing rate is substantially above the Average Firing Rate of the subgraph edges (or vertices).

Definition 6 ***Vertex Label Select.*** *The* label select *value of a vertex is a non-negative integer when its firing rate is above $AFR(t)$. When the value is positive, it is equal to the number of standard deviations by which its firing rate exceeds $AFR(t)$.*
$LabelSelect(w,t) =$

$$\begin{cases} -1 & \text{if } FR(w,t) \leq AFR(t) \\ \left\lfloor \dfrac{FR(w,t) - AFR(t)}{\sigma(t)} \right\rfloor & \text{otherwise} \end{cases}$$

12.5 Recency and Top K Filters

12.5.1 Recency Filtering.

In order to focus on relatively recent subgraphs, we systematically remove edges and vertices from the graph stream when they have been inactive for some data-driven time interval. It is based on another time-dependent parameter of the graph stream, that we call $ScreenTime_{CG}(t)$ which is a function of the processor intake rate and the screen display capacity. The intention is to have an edge removal control mechanism. Namely, if $Recency(e,t) < \frac{1}{ScreenTime_{CG}(t)}$, we remove the edge. If all the edges connected to a node are removed, we remove the node from the recency graph.

12.5.2 Top-K Tape Filtering.

Although the recency graph gives a comprehensive snapshot of what the graph stream looks like at a particular time, it does not keep a record of past events. Once a vertex disappears from the recency graph, no record of it is left in the recency graph regardless of how prominent it may have been. There is no way to tell how long the vertices have been in the recency graph or whether and how often they have been labeled. We would like to know not only about the present state of the data stream but also have a description of important past events. In order to do this, we introduce the **Top-K Tape**, which encodes a window into the past activity of the recency graph.

Let $\mu_e(t)$ be the average edge firing rate for all edges in the co-occurrence graph up to time point t. Let $\sigma_e(t)$ be the standard deviation of the firing rate of all edges up to time t. If an edge's firing rate drops below $\mu_e(t) + 2\sigma_e(t)$, the edge is removed from the Top K Tape.

We define the **Top-K Tape** (Bottom of Figure 12.1) to be the set of all edges in the recency graph, up to time point t, with firing rate greater than $\mu_e(t) + 2\sigma_e(t)$. The Top-K Tape is represented by a series of horizontal rows of dots. Each of these rows represents an edge. Each column represents a separate time point. The tape has a width that represents a number of time units into the past from the current time. The current time is on the right of the tape and the most distant time point is on the left. The rows are sorted, in non-increasing order, according to their firing rate on the overall graph stream. The Top-K Tape also has two special *virtual* edges, one representing the overall **Graph Stream Firing Rate**, and the other representing the **Top-K Firing Rate**. Both of these edges compete for a slot in the Top-K. These virtual edges are shown in Figure 12.2 in purple at the top row and in black at the start of the bottom row.

Definition 7 ***Edge Persistence.*** *Persistent edges are those edges that have a high appearance rate for an extended period of time, whereas non-persistent edges are those edges that may appear in the graph stream in concentrated bursts, but not enough to be considered persistent. We define edge persistence to be the average length of contiguous edge lifetime segments. Edges with a higher persistence value have either longer total lifespans, or a lower number of contiguous lifetime segments. See next subsection for how these quantities are displayed in the Top-K Tape.*

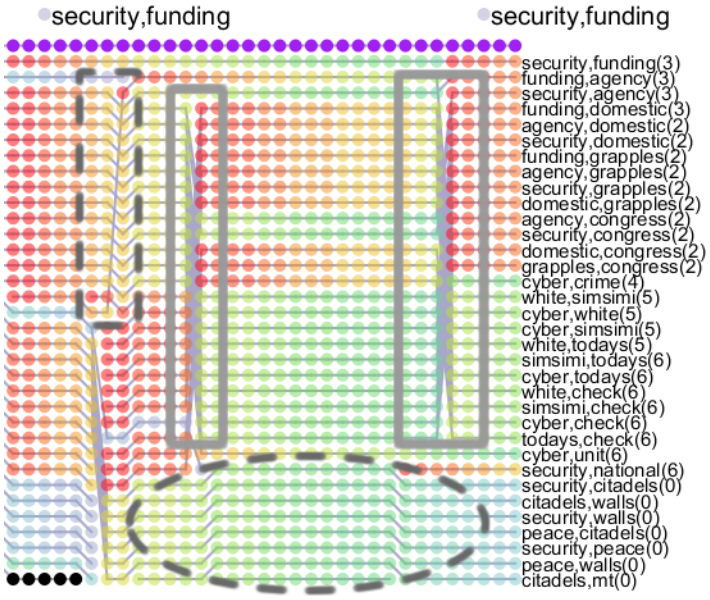

FIGURE 12.2: Tweets captured during a maritime security conference, 3/2/2015 - 3/4/2015

12.6 Graph Stream Visualization

Based on these definitions, we have designed a visualization platform for graph streams as shown in Figure 12.1 that integrates two views. First, a dynamic node link view of "recent" subgraphs that combines an animation with a special supergraph of the past time points. And second, a tape view of the Top-K edge statistics serving as an overview of only the K most important edges within these recent subgraphs for multiple time points.

12.6.1 Recency Graph Visualization

The visualization of the recency graph as depicted in Figure 12.3 has the following visual attributes, which are determined by the previously defined parameters.

Edge Recency Color: Edges in the graph are colored using the diverging spectral color scheme from Colorbrewer2.org [10]. The color of an edge reflects its recency value. An edge is colored red when $Recency(e, t) = 1$. If its recency value decreases, its color begins to fade towards blue. For nodes, we average the recency values of all its incident edges and use that value to determine its color.

Edge Thickness as Edge Firing Rate: The thickness of an edge encodes that edge's present firing rate.

Node Size as Vertex Firing Rate: The size of a node reflects its firing rate. If a node has a high firing rate, it will be larger on the screen than other nodes with lower firing rates.

Vertex Label Size: A node is unlabeled if its Label Select value is -1. Nodes with higher Label Select values are assigned larger fonts.

Visually, nodes with large textual labels are those whose average firing rate is at least one standard deviation above the overall firing rate of the graph stream.

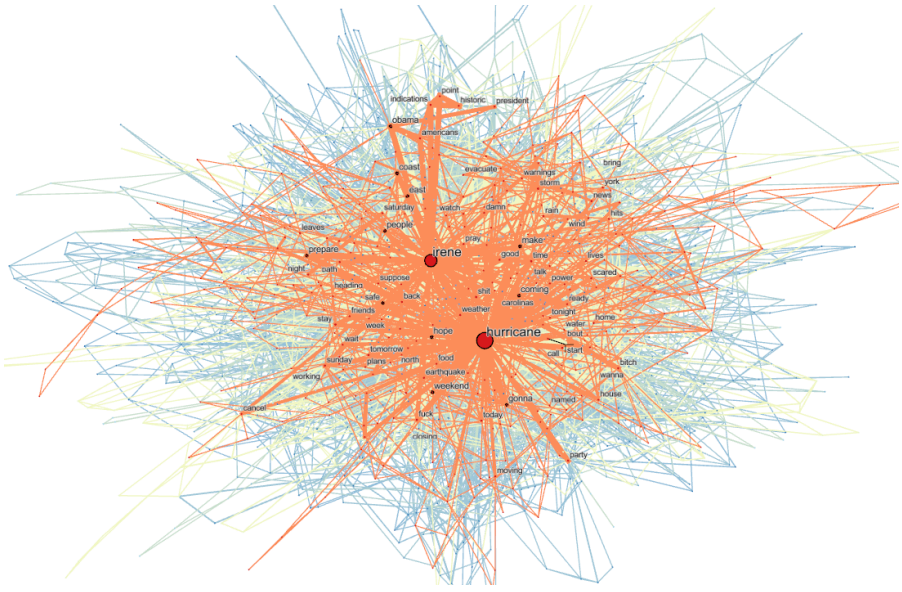

FIGURE 12.3: The recency graph of a stream of tweets on Hurricane Irene

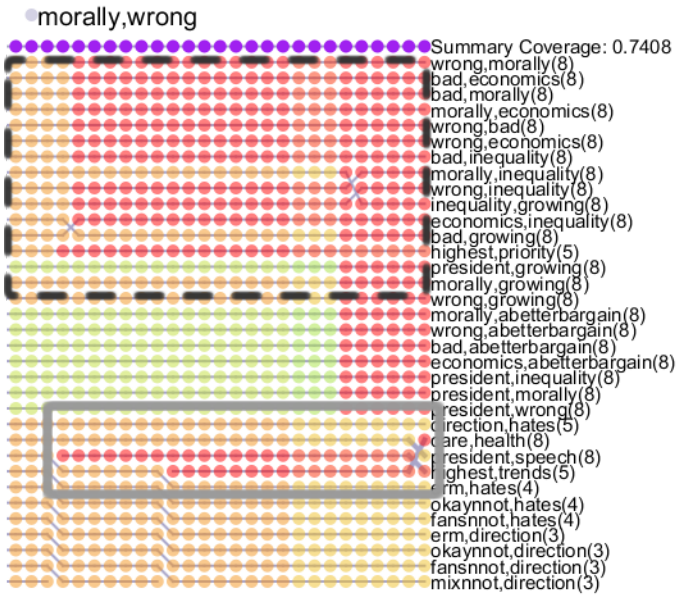

FIGURE 12.4: Speech on the economy by President Obama, 07/24/2013

12.6.2 Top-K Tape Visualization

An edge appearing in the Top-K Tape has the following visual attributes as shown in Figure 12.4.

Dot Color: As in the recency graph, dot color indicates recency. There is a direct correspondence between the color of a dot on the tape, and the edge occurrence it represents in the recency graph.

Dot Size: The size of a dot is a variable controlled by the user to increase visibility.

Dot Label: The dots at the right-most column are labeled with the two vertices of the edge they represent. The order of these vertices in the label is determined by the vertices' total frequency in the co-occurrence graph. Each dot label also contains an additional number in parenthesis. This number is the edge's persistence value in the co-occurrence graph.

12.7 Life Cycle in a Graph Stream

The life cycle of a vertex can be described as follows. When a vertex first appears in the graph stream, it is added to the recency graph. If a vertex is inactive in the graph stream for too long, then it is removed from the recency graph. If that vertex appears again in the graph stream, it is treated like the first occurrence. Otherwise, the vertex becomes labeled when its firing rate exceeds $AFR(t)$. Note that a vertex can cycle between the two states "labeled" and "unlabeled" if its firing rate fluctuates between above and below $AFR(t)$. The life of a vertex in the recency graph is succinctly shown in the transition diagram in Figure 12.5. Orthogonally, an edge has its own life cycle in a specially selected Top-K sub-stream, from which we can identify those vertices which cover prominently a substantial portion of the overall graph stream. The main mechanism for this identification is based on edge group behavioral patterns. They include *trending, untrending, herding, and straying.* Their details are discussed in the next section.

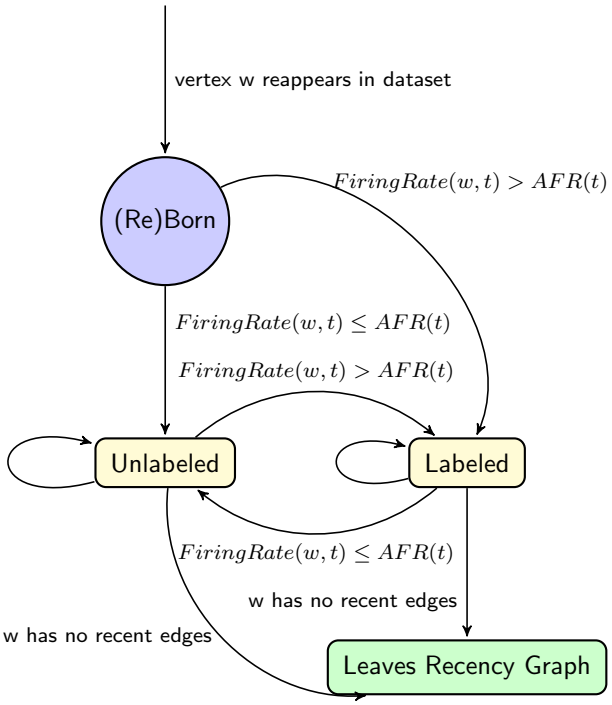

FIGURE 12.5: Vertex life in the Recency Graph

12.8 Top K Edge Group Patterns

12.8.1 Trending and Untrending

Definition 8 ***Trending and Untrending.)*** *The Top-K edges are ranked in non-increasing order by their firing rate. If an edge increases in rank between times t and $t+1$, we say that edge is trending. Likewise, if an edge decreases in rank between times t and $t+1$, we say that edge is untrending. Since it may be the case that an edge is trending/untrending, even though its velocity remains unchanged, we introduce next the notion of* ***significance*** *in trending/untrending.*

Definition 9 ***Significant Trending.)*** *An edge e exhibits significant trending if $rank(e,t) < rank(e,t+1)$ and $fr(e,t) < fr(e,t+1)$.*

Trending/Untrending can be detected visually by examining the lifetime path of an edge in the Top-K Tape. If an edge is trending, it will be shown by a visual upward line between an edge's representative dots. If there is an increase in color while an edge increases in rank, this signifies Significant Trending. If there is a decrease in color while an edge decreases in rank, this signifies Significant Untrending (see dashed gray rectangle in Figure 12.2).

12.8.2 Herding and Straying

Edges that commonly co-occur share similar trending/untrending behavior relative to one another. We introduce the concepts of **Herding** and **Straying** (see light gray square in Figure 12.6).

Definition 10 ***Herding.*** *Two edges, e and f, are defined to be herding if $diff(rank(e,t), rank(f,t)) = diff(rank(e,t+1), rank(f,t+1))$, where diff signifies the difference between two quantities.*

Certain edges that are herding across multiple subgraphs, may break away from their herd. An edge, e, is said to be **straying at time point** t (refer to light gray rectangle in Figure 12.4) if e was herding with **at least** h other edges at time point $t-1$ and e is not herding with any edge at time point t.

12.8.3 Pattern Identification

At any time, an edge participates in a combination of Trending/Untrending and Straying or Herding patterns, as defined in the previous subsections. The possible transitions between these patterns can be formalized by a set of finite state transitions (Figure 12.7). These patterns can be used to generate summaries for a graph stream.

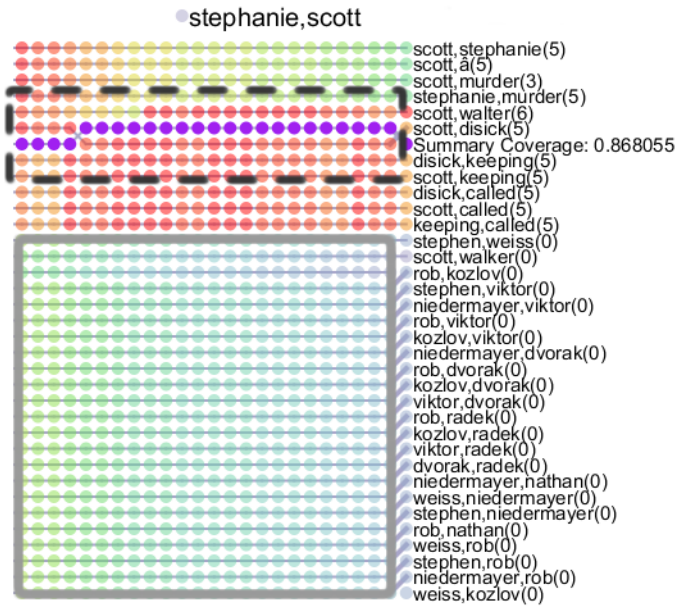

FIGURE 12.6: Tweets from the day following the Walter Scott shooting, 4/9/2015 - 4/12/2015

Definition 11 ***Pattern Score.*** *If an edge is Trending/Untrending over a time interval $t_{pattern}$, we assign it pattern points equal to its net difference in rank during $t_{pattern}$. If an edge is Herding, we subtract points equal to the number of edges it is herding with. If an edge is straying, we add to its pattern score points equal to the size of the herd it strayed from.*

If an edge's score is strictly greater than 1, we place the edge into our data summary. At this point, we verify if the summary is meaningful, in the sense that it provides a large edge cover of the graph stream.

12.8.4 Verification and Evaluation

One useful concept involved in analyzing a graph stream is persistence. The Top-K Tape records the history of recency and firing rate over time. However, for longer data sets, the Top-K Tape may become too long with respect to screen capacity, and not viable to view in its entirety. Thus, we would like to summarize the graph stream via patterns that we observe over the Top K Tape's lifetime. These patterns (Trending, Untrending, Herding, Straying) can be used to generate a collection of edges that represent a summary of the Top-K Tape.

In early implementations, we had a human recording these events by hand.

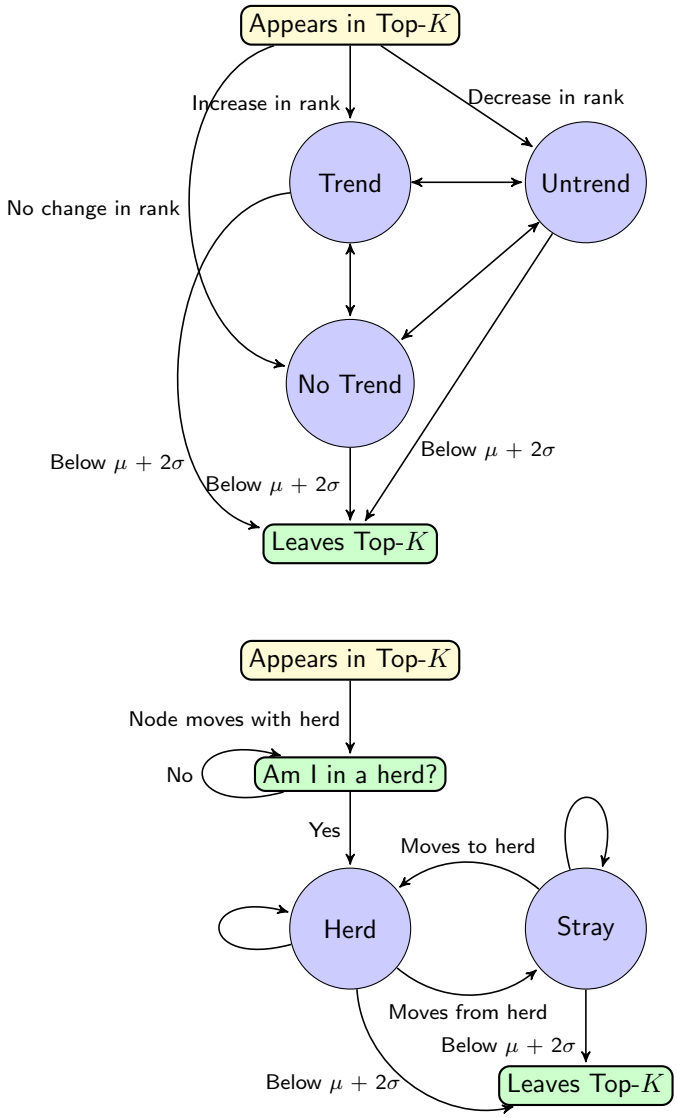

FIGURE 12.7: Parallel life cycles of a Top-K edge

Later, we automated the process to have a computer agent record these pattern occurrences. To compare the two approaches, we needed to develop a verification metric for a summary of a graph stream. We defined a summary to be a collection of vertices. A "good" summary would be a collection of vertices that covers a large portion of the graph stream. A "great" summary would be a minimum vertex cover for the entire graph stream. With this in mind, we defined a normalized metric to compare pattern summaries. As finding a minimum vertex cover is *NP*-Complete, we will instead use a maximal matching, which has a 2-approximation ratio to the a minimum vertex cover. However, in extremely large graph streams (in terms of number of vertices and edges), even taking the approximation may be too expensive. We take a different approach by incrementally taking the union of several smaller maximal matchings.

Algorithm 1 Incremental summary ratio calculation algorithm

Data: $graphStream_t$, the graph stream up to time t
Result: Normalized Summary Ratio VertexCover = {}
  for each time point t **do**
    Let TK be the set of edges in the Top-K at time t
    ACOVER = approxCoverViaMaximalMatching(TK)
    VertexCover = VertexCover ∪ ACOVER
    record numberOfVerticesCoveredBy(VertexCover) / numberOfNodes($graphStream_t$)
  end for

We incrementally construct a cover by selecting a maximal matching for the subgraph induced by the members of the Top-K Tape at the current time point t. We union the maximal matching of this subgraph with our running vertex cover. We record the ratio of coverage size over the total number of vertices (see Algorithm 1).

Complexity. Ultimately, our proposed approach to graph stream processing depends on

- the graph stream edge arrival rate,
- the processing speed of the available computing platform, and
- the amount of RAM buffer space available, for incrementally maintaining
  - A fixed number of cumulative vertex graph stream statistics that include vertex firing rate and their most recent active timestamps.
  - The statistically selected Top-K edges and the corresponding new set of vertices used to extend our greedy maximal matching cover ratio.

Assuming that the processing speed ps is at least twice the graph stream

edge arrival rate, and that the available RAM is $O(|V_t|)$ we can process any graph stream $G_t = (V_t, E_t)$ in the worst case in time $O(|V_t|)$. This conforms to the semi-external model of computation. However, in the arguably realistic scenario where, at each time point t, the size of each instantaneous subgraph is bounded, we can incrementally update the Top-K Edge buffer in time proportional to its size.

12.8.5 Holes

Another useful pattern is what we call a **hole**. A hole at time point t occurs when the number of edges with velocity greater than $\mu_e(t) + 2\sigma_e(t)$ is less than the number of edges with velocity greater than $\mu_e(t+1) + 2\sigma(t+1)$. As the number of edges included in the Top-K is dependent on the velocity distribution of the graph stream, a hole signifies a change in the mean and standard deviation of this distribution.

12.9 Twitter Data Sample Results

In order to test our graph stream abstraction, we created an implementation to process Twitter data. We provide to the Twitter API a set of *query words* and obtain a collection of time-stamped tweets containing the input keywords. We first remove non-alphanumeric words, stop words, convert all letters to lowercase, and use a stemming algorithm [15] to represent words with similar stem by the same representative. Each word in a tweet is mapped to a vertex in the graph stream. If the set of words in a tweet is considered ordered according to their order of appearance in the tweet, it makes sense to consider two words w-connected by a tweet if they are separated by no more than w tweet words. The subgraph corresponding to a tweet is the graph with vertices consisting of all words in a tweet and edges drawn between all pairs of w-connected words. These cliques of w-connected words become our instantaneous events (see definition 3).

The TwitterMap interface has several customization options for exploring a given dataset. The user has the ability to pause/resume the incoming graph stream for more detailed observation. They can also change the speed at which tweets are read into the system, and the rate at which tweets will decay after they have been processed in the system. In addition to this, the user can control the number of labels present in the recency graph, and the visible size of both the nodes and the edges within the Top-K Tape. We remove query words from the analysis via a toggle button.

12.9.1 Sample Results

Figure 12.2 shows tweets recorded during a major international conference on "Maritime and Cyber security" under the query [maritime, security], on 03/02/2015 through 03/03/2015. Figure 12.4 shows a tape segment recorded during a speech by President Obama on the subject of economic reform, under the query [president, obama] on 07/24/2013. Figure 12.6 shows tweets recorded the days following the police shooting of an unarmed African American man under the query [walter, scott] on 04/09/2015 through 04/11/2015. Each of the shown tape segments differ from one another in terms of the resulting visible color scheme and the types of visible Top-K patterns.

These visible patterns give us an intuitive way to visually describe the graph stream over this time window. For example, the Obama tape segment has

- few straying word pairs (light gray rectangle in Figure 12.4),
- heavy herding (light gray rectangle in Figure 12.4), and
- little color variation (both rectangles in Figure 12.4).

The fact that most of the word pairs are not changing in rank shows that the conversation is dominated by a set of common words (see Figure 12.4 dashed dark gray rectangle). There are few straying words (see Figure 12.4 light gray rectangle) most likely caused by a group of select tweets being heavily retweeted during this time window. This may be a characteristic of large live public events where many commentators are sharing their views over social media. We can also notice that the highest ranked words are also negative in connotation. We see word pairs such as "morally wrong" and "bad economics" ranked at the top of our view.

We can contrast the Obama set with the Walter Scott set. One may notice that the Walter Scott set features

- a varied color palette (dashed gray rectangle in Figure 12.6),
- large size herding (light gray rectangle in Figure 12.6), and
- a group of word pairs ranked above the Top-K purple virtual edge line (see top of Figure 12.6).

The varied color palette shows that this conversation exhibited a more diverse collection of activity patterns than the Obama speech set. The word pairs have more time between their occurrences, and thus we see more of the color scale (see light gray rectangle in Figure 12.6). The heavy herding tells us that this conversation is also dominated by several groups of words with similar behavior, most likely due to several tweets being heavily retweeted in this time frame. The group of words above the Top-K purple virtual line indicates that this particular set of words recently had a spike in activity (see

top of Figure 12.6). We see that these word pairs are focused on the victim, Walter Scott, and his wife Stephanie.

The security tape segments exhibit the following characteristics

- varied color palette (see entirety of Figure 12.2),
- a mix of both straying and herding (notice the contrast between straying in the dashed rectangle and herding in the dashed oval in Figure 12.2),
- the black virtual edge is visible (see bottom left of Figure 12.2), and
- a relative lack of rank stability (light gray rectangles in Figure 12.2).

The varied color palette still signifies that this conversation has a more diverse activity set of patterns than the Obama speech set (see dashed oval in Figure 12.2). The visibility of the black virtual edge is indicative of a rise in system acceleration (i.e., increasing velocity in a small period of time). Observe that the word pair "security funding" strayed away from its herd (see dashed rectangle in Figure 12.2), signifying a change in the focus of the conversation.

12.10 Degree-of-Interest-based Visual Exploration

The previous sections have looked at the big picture of graph streams. Once the viewer is familiar with that big picture, the question remains what else is there? Are there further interesting characteristics that are drowned out by the most prevalent features? This can be seen in the recency view in Figure 12.3, where the two most common keywords "hurricane" and "irene" are dominating the visualization making it very hard to take a closer look at the other identified keywords. In this section, we take a closer look at these "undercurrents" of the graph stream.

The method we apply for doing so is based on Degree-of-Interest (DoI) functions. We can use these functions to capture those aspects that interest us in the graph stream and increase the DoI of the involved nodes and edges, while at the same time decreasing the DoI of nodes and edges involved in aspects that are not of interest to us. Using the modular interface for specifying DoI functions that we have previously presented [2], we can zero in on features of interest in a step-by-step fashion depending on what we find in the process.

As a starting point, we use Figure 12.3 that shows the recency graph of the tweets from 26-AUG-2011 with the two dominant keywords "hurricane" and "irene". We then define a DoI function that assigns a low DoI to these two keywords (see DoI-module A in Figure 12.8), as well as to keywords with very low firing rates (see DoI-module B in Figure 12.8) to cut down on the clutter. The result of these removals can be seen on the right side of Figure 12.8.

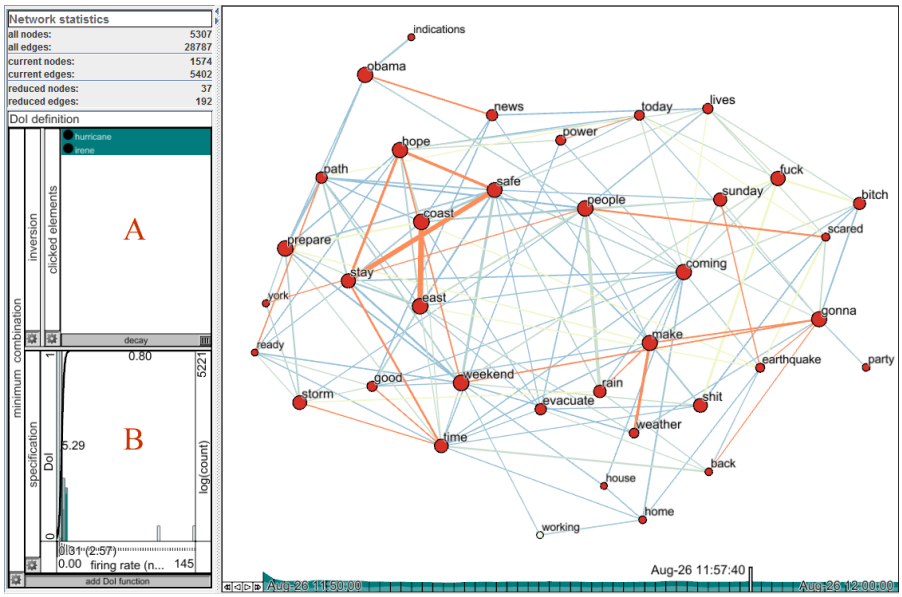

FIGURE 12.8: The first step of the DoI definition: Building a DoI function that removes the dominant keywords by manual selection (DoI module A) and edges of low firing rate (DoI module B) by assigning them low DoI values. The result can be seen on the right side. $doi_1(x_i) = min(\{inv(select(x_i)), inter(firing\_rate(x_i)))\})$

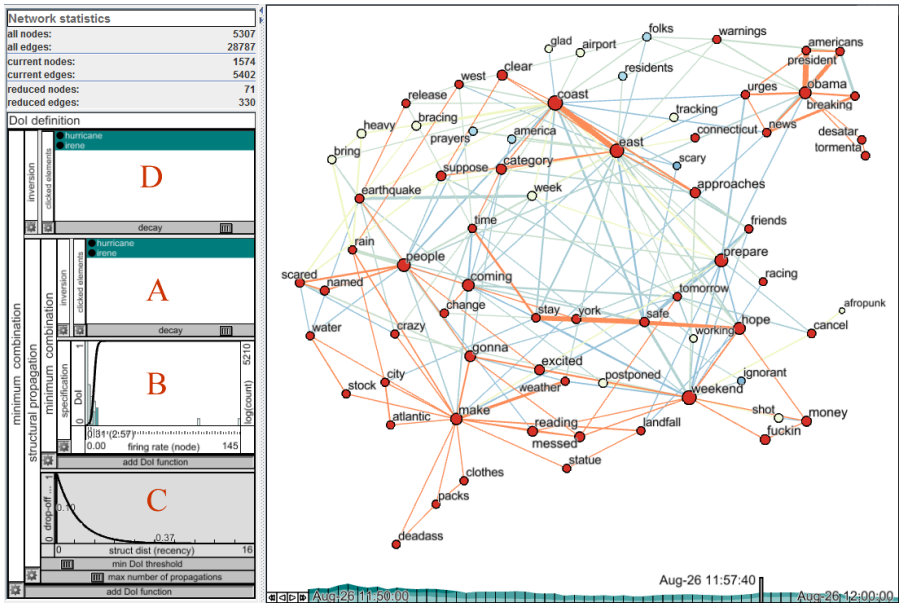

FIGURE 12.9: The second step of the DoI definition: Enhancing the DoI function by adding a structural propagation (DoI module C) that distributes high DoI values to neighboring nodes. The result can be seen on the right side. $doi_2(x_i) = min(\{inv(select(x_i)), prop_s(doi_1(y_i))\})$

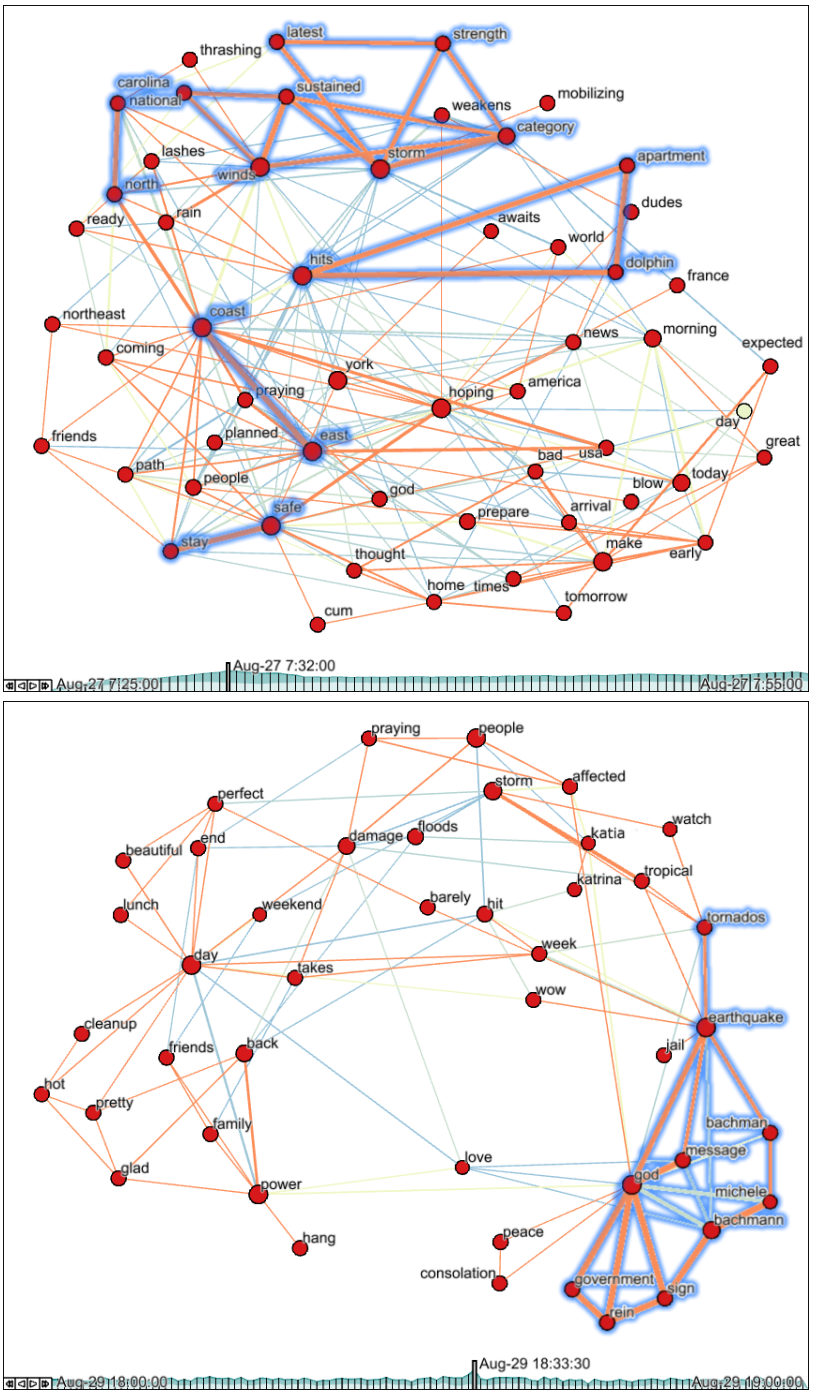

FIGURE 12.10: Additional results of the DoI function previously defined in Figure 12.9 for two time points: 27-AUG-2011 (top) and 29-AUG-2011 (bottom).

Smaller patterns emerge now and we can already see from this figure that people were scared, president Obama was on the news, and that there were many tweets about staying safe. One can also see a number of swear words that appear frequently in the context of the hurricane tweets. A secondary analysis using the Linguistic Inquiry and Word Count (LIWC)[1] confirms that these tweets are overwhelmingly emotionally negative with a 37.9 score on emotional tone. Scores below 50 denote emotional negativity, scores above 50 denote emotional positivity [7]. To put this score in context, according to the LIWC website the average score on emotional tone in social media lies at 63.35.

This first step allowed us already to form a more differentiated picture than the original image. The drawback of the defined DoI function is that it only focuses on keywords that occur often, but neglects those words that appear in the context of these frequent keywords – i.e., words that are not as frequent by themselves but regularly tweeted alongside the frequent words. These contextual keywords can help to further derive meaning from the observed structures. Hence in a second step, we add them back in by applying a structural propagation of the DoI values derived from the first step to their neighboring nodes, which is depicted as DoI-module C in Figure 12.9. Since this propagation also adds the keywords "hurricane" and "irene" back in, we have to subtract them again as shown by DoI-module D in Figure 12.9. The result of applying this revised DoI function is depicted on the right side of the figure. While at this point, it does not generate additional insights, one can clearly see that certain keywords are now shown with more context – e.g., "obama" is now connected with "president" and "americans", and the hurricane apparently "approaches" the "east" "coast".

Prepared in such a way, we can now take a look at later time points using the very same DoI function to examine the patterns that lie beneath the overall Twitter hype around "hurricane" "irene", which we filtered out. Figure 12.10 shows two time points: 27-AUG-2011 (top) and 29-AUG-2011 (bottom). In the snapshot from 27-AUG-2011, one can see a pattern surrounding the "category" of the "storm", as its "strength" "weakens". This captures nicely the time point at which it was announced that hurricane Irene was downgraded to a Category 1 hurricane. This is also reflected in the emotional tone of these tweets, which score with 59.5 a much higher and even slightly positive emotional tone than the day before. Yet this score plunges again right after the landfall of Irene on the East Coast that brought flooding and power outages in its wake. The emotional tone reaches a minimum of 34.92 on 29-AUG-2011, when then presidential candidate Michele Bachmann commented on the disastrous situation on the East Coast that hurricanes and earthquakes are god's warning to Washington. Mentions of this quote were frequently retweeted, so that a corresponding pattern shows in Figure 12.10 (bottom). The overall negative reception of this quote on Twitter together with the first mentions

[1] see http://www.liwc.net

of the next "tropical" "storm" of the 2011 hurricane season "katia" seems to have brought the emotional tone down to this minimum.

From these examples, it becomes apparent that the DoI-based inspection of the streaming data is a valuable tool to define just the parts of the data that are of interest while cutting down on the noise (i.e., keywords of low interest cluttering the view) and the already known facts (i.e., the most dominant keywords). Since the DoI function can be adjusted at any time throughout the exploration, found facts – i.e., prominent keywords – can be added to the exclusion list to bring out even more subtle details.

12.11 Conclusions

We have introduced a simple and useful view of graph streams as co-occurrence graphs. The fundamental driving statistics are edge firing rate, recency and edge persistence. They allow us to isolate edge group patterns like "herding" and "straying" that can be used as proxies of interesting graph stream behaviors. Salient nodes in the graph stream pop up as those vertices with persistent firing rates substantially above the average firing rate of the entire graph stream. Their selection is certified by their coverage ratio of the entire stream. This ratio can be incrementally verified by either a human or a computer. Drowned-out "undercurrents" of the graph stream can be brought to light by using a modular Degree-of-Interest function that filters out the known patterns in the stream and enhances the unknown ones.

The approach suggested here presents several directions of future research. They include:

- Summarization of graph streams by transition diagrams
- Collaborative exploration of graph streams
- Detection of verifiable graph stream properties. Candidates include: streams entropy, entropy norms, and discrepancy.

12.12 Acknowledgments

We acknowledge the partial support of the US National Science Foundation (DIMACS REU NSF Grant CCF-1263082, 2015) and of the federal state of Mecklenburg-Vorpommern within the EFRE project "Basic and Applied Research in Interactive Document Engineering and Maritime Graphics" (Grant no: ESF/IV-BM-B35-0006/12). The version of TwitterMap presented here was created with the Java visualization library Graph Stream (http://graphstream-project.org).

Bibliography

[1] James Abello, Tina Eliassi-Rad, and Nishchal Devanur. Detecting novel discrepancies in communication networks. In Geoffrey I. Webb, Bing Liu, Chengqi Zhang, Dimitrios Gunopulos, and Xindong Wu, editors, *ICDM'10: Proceedings of the International Conference on Data Mining*, pages 8–17. IEEE Computer Society, 2010.

[2] James Abello, Steffen Hadlak, Heidrun Schumann, and Hans-Jörg Schulz. A modular degree-of-interest specification for the visual analysis of large dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 20(3):337–350, 2014.

[3] Daniel Archambault, Helen Purchase, and Bruno Pinaud. Animation, small multiples, and the effect of mental map preservation in dynamic graphs. *IEEE Transactions on Visualization and Computer Graphics*, 17(4):539–552, 2011.

[4] Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. GraphDiaries: Animated transitions and temporal navigation for dynamic networks. *IEEE Transactions on Visualization and Computer Graphics*, 20(5):740–754, 2014.

[5] Fabian Beck, Michael Burch, Stephan Diehl, and Daniel Weiskopf. The state of the art in visualizing dynamic graphs. In Rita Borgo, Ross Maciejewski, and Ivan Viola, editors, *EuroVis'14: State-of-the-Art Reports of the Eurographics/IEEE Symposium on Visualization*, pages 83–103. Eurographics Association, 2014.

[6] Ulrik Brandes, Natalie Indlekofer, and Martin Mader. Visualization methods for longitudinal social networks and stochastic actor-oriented modeling. *Social Networks*, 34(3):291–308, 2012.

[7] Michael A. Cohn, Matthias R. Mehl, and James W. Pennebaker. Linguistic markers of psychological change surrounding September 11, 2001. *Psychological Science*, 15(10):687–693, 2004.

[8] Stephan Diehl, Carsten Görg, and Andreas Kerren. Foresighted graphlayout. Technical Report A/02/2000, Universität des Saarlandes, 2000.

[9] Steffen Hadlak, Heidrun Schumann, and Hans-Jörg Schulz. A survey of multi-faceted graph visualization. In Rita Borgo, Fabio Ganovelli, and Ivan Viola, editors, *EuroVis'15: State-of-the-Art Reports of the Eurographics/IEEE Symposium on Visualization*, pages 1–20. The Eurographics Association, 2015.

[10] Mark Harrower and Cynthia A. Brewer. Colorbrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[11] Natalie Kerracher, Jessie Kennedy, and Kevin Chalmers. The design space of temporal graph visualisation. In Niklas Elmqvist, Mario Hlawitschka, and Jessie Kennedy, editors, *EuroVis'14: Short Paper Proceedings of the Eurographics/IEEE Symposium on Visualization*, pages 7–11. Eurographics Association, 2014.

[12] Kazuo Misue, Peter Eades, Wei Lai, and Kozo Sugiyam. Layout adjustment and the mental map. *Journal of Visual Languages and Computing*, 6(2):183–210, 1995.

[13] Adam Perer and Jimeng Sun. MatrixFlow: Temporal network visual analytics to track symptom evolution during disease progression. In *AIMA'12: Proceedings of the American Medical Informatics Association Annual Symposium*, pages 716–725, 2012.

[14] Bruno Pinaud, Guy Melançon, and Jonathan Dubois. PORGY: A visual graph rewriting environment for complex systems. *Computer Graphics Forum*, 31(3pt4):1265–1274, 2012.

[15] Martin F. Porter. An algorithm for suffix stripping. In Karen Sparck Jones and Peter Willett, editors, *Readings in Information Retrieval*, pages 313–316. Morgan Kaufmann Publishers Inc., 1997.

[16] Helen Purchase, Eve Hoggan, and Carsten Görg. How important is the "mental map"? - An empirical investigation of a dynamic graph layout algorithm. In Michael Kaufmann and Dorothea Wagner, editors, *GD'06: Proceedings of the International Symposium on Graph Drawing*, volume 4372 of *Lecture Notes in Computer Science*, pages 184–195. Springer, 2007.

[17] Sébastien Rufiange and Michael J. McGuffin. DiffAni: Visualizing dynamic graphs with a hybrid of difference maps and animation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2556–2565, 2013.

[18] Purvi Saraiya, Peter Lee, and Chris North. Visualization of graphs with associated timeseries data. In John Stasko and Matthew O. Ward, editors, *InfoVis'05: Proceedings of the IEEE Symposium on Information Visualization*, pages 225–232. IEEE Computer Society, 2005.

[19] Klaus Stein, Rene Wegener, and Christoph Schlieder. Pixel-oriented visualization of change in social networks. In Nasrullah Memon and Reda Alhajj, editors, *ASONAM'10: Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*, pages 233–240. IEEE Press, 2010.

[20] Andrea Unger and Heidrun Schumann. Visual support for the understanding of simulation processes. In Peter Eades, Thomas Ertl, and Han-Wei Shen, editors, *PacificVis'09: Proceedings of the IEEE Pacific Visualization Symposium*, pages 57–64. IEEE Computer Society, 2009.