Discovering and Summarising Regions of Correlated Spatio-Temporal Change in Evolving Graphs

Jeffrey Chan, James Bailey, Christopher Leckie
NICTA VRL, Department of Computer Science and Software Engineering
The University of Melbourne, Australia
{jkcchan, jbailey, caleckie}@csse.unimelb.edu.au

Abstract— **Graphs are adept at describing relational data, hence their popularity in fields including network management, webpage analysis and sociology. However, most of the current graph mining work regards graphs as static and unchanging, even though many graphs are dynamic. In this paper, we introduce a new pattern to discover from evolving graphs, namely regions of the graph that are evolving in a correlated manner. These regions of correlated spatio-temporal change group together graph changes that are topologically near (spatial) and evolve similarly (temporal) to each other. The regions can be used to summarise changes, particularly for graphs that have many simultaneous changes. We have developed an algorithm called cSTAG to summarise changes in dynamic graphs. This new algorithm discovers these regions of correlated change and identifies events that caused these changes. As a demonstration of the effectiveness of our algorithm, we applied cSTAG to summarise the changes to the Border Gateway Protocol connectivity graph during the 2005 Hurricane Katrina Disaster. cSTAG was able to identify the reported failures in Louisiana, as well as other simultaneous events.**

I. Introduction

There is growing interest in analysing changes in graphs that evolve with time. Examples include analysing changes in global characteristics of evolving graphs [1][2], and discovering patterns of change within graphs [3]. Other studies have analysed incremental changes in graphs. Examples include detecting anomalous changes using regions of the greatest change [4], and an incremental PageRank algorithm [5].

Although most of this prior work is promising, none have considered the problem of finding patterns that summarise significant changes in an evolving time series of snapshots. This is particularly important in large graphs where there are many simultaneous changes that are caused by different underlying events/causes. Consider a large network such as the routing topology of the Internet. A single cause, or event, such as congestion in a router, can affect a large region of the network, such as changes in the routing topology that are triggered by delays caused by the congested router. To complicate matters, these changes may emerge gradually over time as the problem develops, and there can be several problems occurring simultaneously in the network. Hence, we require a technique that can group related changes together into regions, so that users can analyse the underlying events that caused those changes. This grouping of changes needs to take into consideration both the topological and temporal extent of the changes, as well as discriminating between unrelated changes.

In this paper, we introduce the problem of how to identify regions of a graph that are evolving in a correlated spatial and temporal manner. These regions of correlated spatio-temporal change can then be used to identify events that caused these changes. We have developed an algorithm called cSTAG (*clustering for Spatial-Temporal Analysis of Graphs*) to discover regions of correlated spatio-temporal change, and use these regions to characterise the events that caused them. Our approach is to first represent graph changes as waveforms, where a change waveform is associated with each link in the graph. We then cluster these waveforms into regions of similar change, based on their spatial and temporal similarity. Finally, we correlate the evolution of these regions of similar change in order to characterise the underlying events that have caused these changes.

In contrast to previous approaches that detect incremental changes in graphs [4][5], our approach considers the evolution of changes over an extended period of time. This is particularly useful for identifying the characteristics of the change event, and identifying significant graph changes among a large number of random changes. In contrast to techniques that analyse the overall statistics of the graph, such as the diameter of the graph [1], our approach is able to identify simultaneous but distinct changes that are caused by different events. This is particularly useful for tasks such as fault detection and prioritisation, where it is necessary to correctly identify the number of independent events that are causing changes in the network. For example, it is important to avoid grouping unrelated changes into a single event, or duplicating effort by reporting a single event as multiple changes.

Related prior work includes spatio-temporal mining and dependency analysis. Spatio-temporal mining involves discovering patterns that are correlated in space and time [6]. Although similar in aim, the spatial dimension in this type of mining is geographical, whereas the spatial dimension in our work is related to the topological characteristics of graphs.

Dependency analysis focuses on discovering the dependencies between elements and attempting to find the root causes and their impact [7]. We consider our problem to be an extension of this approach, as it requires discovering the actual symptoms (regions of correlated change), as well as inferring their evolution to summarise the symptoms observed (event identification).

In summary, the contributions of our current work are:

1) We propose a new pattern to mine from evolving graphs, namely regions of correlated spatio-temporal change;
2) We propose a near real-time method, cSTAG, to mine these patterns and summarise the underlying events;
3) We demonstrate the utility of using regions of correlated change in event correlation by analysing the effect of the 2005 Hurricane Katrina Disaster on the Internet routing topology.

II. Methodology

In this section, we formally define the problem of discovering regions of correlated change and then describe our method, cSTAG.

Events have a limited lifetime. Therefore, the region affected by an event will only show correlated behaviour for a limited period. As we do not know a priori the length of each event, we segment the time series of snapshots into a number of windows. Therefore, the problem of event identification by analysing changes across snapshots of a graph consists of two sub-problems:

1) Discover the spatially and temporally correlated regions within each window;
2) Associate the discovered regions across windows, grouping the highly correlated regions to form evolution trees of each underlying event.

More formally, we can formulate the first sub-problem of *discovering regions of correlated spatio-temporal change* as follows. Let $< G_1, \ldots, G_S >$ represent a time series of S snapshots, where G_d denotes the d^{th} graph snapshot, and let $W_\omega =< G_k, G_{k+1}, \ldots, G_{k+W} >$ represent a window of snapshots, $1 \leq k \leq S - W$, where W is the window length. Let $G_{<\omega>}$ be the union graph of W_ω. Let $E(G)$ be the edges of graph G, and $E_{<\omega>c} \subseteq E(G_{<\omega>})$ be the set of edges that have experienced some change in W_ω. Then the problem is to seek a partition of $E_{<\omega>c}$, $\{R_{\omega 1}, \ldots, R_{\omega l}\}, R_{\omega g} \cap R_{\omega h} = \emptyset, g \neq h, 0 \leq g, h \leq l$, such that all edges in each partition have correlated temporal evolution over the window, and they form a region of high spatial proximity. These partitions are the regions of correlated spatio-temporal change.

A. Outline of cSTAG

In this section, we describe cSTAG. Figure 1 rises the main steps in cSTAG. cSTAG initially partitions the time series into a number of overlapping fixed windows. For each window of snapshots, cSTAG discovers the set of regions of correlated change. Once all windows have been analysed, cSTAG finds the set of links that best explain the evolution between regions of different time windows. The regions and links between them form an evolution graph. The components of the evolution graph form DAGs (directed acyclic graphs), which represent the evolution history of each event. In this initial work, we focus on undirected, unweighted, labeled graph snapshots. We concentrate on changes to edges, as changes to vertices will induce changes to their incident edges.

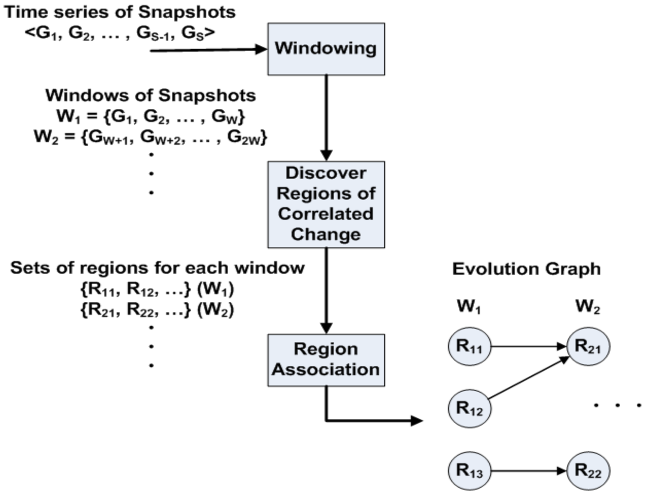


Fig. 1: Summary of the cSTAG process.

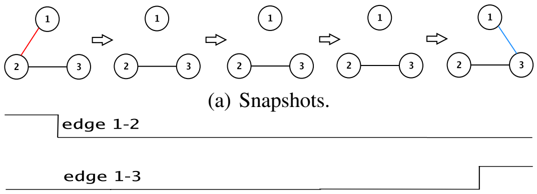
B. Discovering the Regions of Correlated Spatio-Temporal Change within a Window

The problem of discovering regions of correlated change (within each window) is one of clustering in both the spatial and temporal domains, and defining measures to determine the temporal and spatial similarities between the edges that have changed. One solution is to formulate a new measure that combines temporal and spatial similarity, and then use that new measure as the clustering criterion. However, it is difficult to formulate an accurate, yet generic aggregation formula. Hence, we first cluster in the temporal domain, then the spatial domain. This is similar to constrained clustering [8], where we take the spatial domain as the constraining domain. In the following subsections, we describe how we represent and compare the temporal evolution of edges. In addition, we outline the spatial proximity criterion, and how the changed edges are clustered using the temporal and spatial measures.

Computing Temporal Similarity: We represent the temporal evolution of each edge by a change waveform. For unweighted graphs, the only possible edge changes are the appearance and disappearance of edges. Therefore, we construct the change waveform by assigning a value of 1 when the respective edge is present in a graph snapshot, and 0 when it is absent. Formally, let t_i denote the change waveform of changed edge e_i, and $t_i[k]$ denote the k^{th} value of waveform t_i (i.e. its value in the k^{th} snapshot of this window), $1 \leq k \leq W$. Then $t_i[k] = 1$ if $e_i \in G_{x \cdot W+k}$, else $t_i[k] = 0$ for window x. As an example, Figure 2a shows a window of five snapshots. The set of changed edges is {1-2, 1-3}, and the respective change waveforms are displayed in Figure 2b.

To compare the temporal similarity of two edges, we can compare their waveform similarity. The desired waveform similarity measure should be efficient and simple. Therefore, we used a similarity measure based on an edit distance metric and the shape of the waveforms. Edit distance alone does not consider the differences in waveform shapes.

We represent the shape of a binary-valued waveform by a sequence of transitions. Let a $1 \rightarrow 0$ transition in the waveform



(a) Snapshots.

(b) Corresponding change waveforms, for edge 1-2 and 1-3.

Fig. 2: Five snapshots and the corresponding change waveforms.

be denoted as $-$, and a $0 \rightarrow 1$ transition as $+$. Then we define a sequence of transitions $trans_j$ for waveform t_j as a sequence of alternating $-/+$ transitions. Two waveforms are considered to have the same shape if their respective transition sequences are of the same length and the sequence of $-/+$ transitions match. For example, $trans_{1-2} = <->$ and $trans_{1-3} = <+>$ in Figure 2b, so the waveforms have different shapes. The similarity measure based on edit distance is computed as $editSim(t_i, t_j) = W - \sum_{k=1}^{W} t_i[k]\ \mathit{XOR}\ t_j[k]$. Therefore, given two waveforms t_i and t_j (for edges e_i and e_j), the waveform similarity measure is defined as:

$$wavSim(t_i, t_j) = \begin{cases} 0, & trans_i \neq trans_j \\ editSim(t_i, t_j), & \text{otherwise.} \end{cases}$$

Intuitively, the waveform similarity measure determines the number of snapshots in which the two edges are either both present or both absent, while taking into consideration the shape of the waveforms. Two waveforms are similar if both their sequences of changes are the same and their similarity based on edit distance is high (i.e. $editSim()$ is close to W). Consider the example in Figure 2b, where $editSim(t_{1,2}, t_{1,3}) = 3$, but $wavSim(t_{1,2}, t_{1,3}) = 0$ since the waveforms have different shape.

Computing Spatial Similarity: The spatial proximity is computed using a single-linkage criterion. Edges that have experienced change and are topologically connected are considered spatially close. This is a good approximation of spatial proximity of graphs where changes propagate via edges, like computer networks. Therefore regions of the graph that change under the influence of an event are likely to form connected components, whose edges have similar temporal evolution. Note that the definition of topological proximity can be changed. In citation networks, for example, the spatial similarity measure could be high inter-connectedness.

Clustering the Changed Edges: The set of changed edges are clustered to find regions of correlated change. Recall that cSTAG first clusters in the temporal dimension, then the spatial one. The set of changed edges for the window is first partitioned into temporally similar clusters. Then each of the temporally similar clusters are further sub-partitioned, based on the spatial proximity criterion.

The clustering method employed for discovering the temporally similar clusters is a variant of single linkage clustering [9]. For a given threshold T, each changed edge in the window is compared to existing clusters. If the average distance from all edges in a cluster to the compared edge is less than T, then the compared edge joins that cluster. If no existing cluster satisfies this criterion, then a new cluster is created for the edge. An advantage of this clustering method is that it does not require the specification of the number of clusters, and it finds the desired clusters that are temporally similar.

As explained earlier, we define changed edges to be spatially close if they are topologically connected, and temporally similar. Therefore a spatially (and temporally) similar cluster can be found by discovering the connected components among the temporally similar changed edges.

C. Region Association

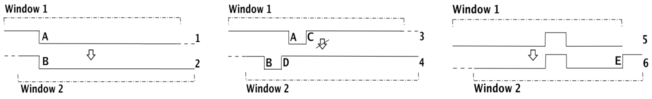
The region association problem involves linking regions of correlated change, which are under the influence of the same events, across windows. There are two issues to consider when solving this association problem. First, an algorithm is required to determine if two regions in different time windows should be linked. Second, a method is required to determine if and how to group together the linked regions as events. In this current work, we propose a simple solution. We shall show that this simple solution can still be successfully used to visually identify events.

Overview of the Region Association Algorithm: Ideally, the decisions of whether to link two regions and how to group the linked regions together as events should be solved simultaneously. However, this global optimisation problem is an extremely difficult problem (it can be mapped to the minimum cut graph partitioning problem, known to be NP-complete[10]). Hence, we first solve the local problem of whether to link two regions, and then identify events by visually identifying the separate DAGs in the resulting evolution graphs.

Fundamentally, we wish to link regions of different windows if a region in one window evolved to another region in the adjacent window, or the two regions represent the same spatio-temporal region (correlation in temporal behaviour longer than a window length). The degree to which two regions of different windows match these cases can be measured by the temporal and spatial similarity between the regions themselves. Two regions of adjacent windows, under the influence of the same event, should have i) similar temporal evolution, ii) be spatially close, as well as iii) be temporally consistent (see Figure 3a and 3b for examples).

As an example of an event causing similar temporal and spatial changes across windows, consider Figure 4, which shows a failure propagation event. In window 1, edges in region R_{11} has failed. Then in window 2, the failure changes have propagated to the edges that enclose region R_{21} (which represents the same region as R_{11}), forming region R_{22}. Note the spatial and temporal similarity of R_{11} to R_{22}. Next, we discuss the measures used and the linking algorithm.

Measures: The temporal similarity between regions, $temSim(R_a, R_b)$, can be measured using the edit similarity measure $editSim()$ between the majority waveforms of R_a and R_b. We do not require two temporally consistent



(a) Consistent waveforms.

(b) Inconsistent waveforms.

(c) Consistent waveforms with different shapes.

Fig. 3: Temporally consistent and inconsistent waveforms.

waveforms to have the same shape, because a region across two windows could have the same waveform in the overlapping parts of the window, but a new change outside the overlapping parts makes the two waveforms have different shapes. Consider Figure 3c, which shows waveforms of two windows for the same (spatial) region of the graph. Waveforms 5 and 6 have different shapes due to the (new) change E, but in the overlapping period between both windows, waveforms 5 and 6 have the same shape.

The spatial similarity measure is based on computing the intersection of the k-neighourhood[1] set of one region with the edge set of the other region. This allows the detection of events where the impact areas move with time, or propagate like in Figure 4, which cannot be detected by the simple intersection of two regions. If $N^k_{R_a}$ denotes the k-neighbourhood edge set of region R_a, then we can define the metric spaSim() as

$$\text{spaSim}(N^k_{R_a}, N^k_{R_b}, R_a, R_b) = \frac{|N^k_{R_a} \cap R_b|}{|R_b|} \cdot \frac{|R_a \cap N^k_{R_b}|}{|R_a|}$$

This metric is high when the two regions overlap or are in close proximity.

Linking Algorithm: To keep the complexity low, the linking algorithm restricts linking to adjacent windows. The algorithm examines each region $R_{\omega i}$ in window ω, and computes the normalised spatial and temporal measures between $R_{\omega i}$ and regions in the next window, $R_{(\omega+1)j}$. A directed link is inserted between the two regions if both the spatial $spaSim()$ and temporal $temSim()$ measures are above their respective thresholds L_{spa} and L_{tem}, and the majority waveform of the regions are temporally consistent. This process is repeated across all pairs of adjacent windows. At the end of this linking process, we have an evolution graph (see Figure 6a for an example). The vertices of the evolution graph, representing the regions of correlated change, are vertically aligned if they are in the same window and scaled according to the size of the region. Time flows from left to right. The other artefacts in the figure will be explained in the next section.

III. Evaluation of cSTAG

In this section, we demonstrate the effectiveness of cSTAG by analysing the effect of the 2005 Hurricane Katrina Disaster on the US portion of the Border Gateway Protocol (BGP) connectivity graph. We compare the output of cSTAG with the reported affected locations and events [11], and compare cSTAG to linking algorithms that only consider either spatial or temporal similarity. Due to space constraints, we do not present our parameter sensitivity or complexity evaluation.

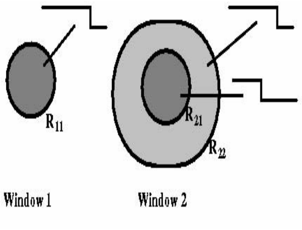


Fig. 4: Propagation of changes from region R_{11}/R_{22} to region R_{22}. Waveforms shown.

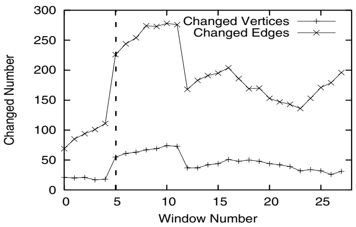


Fig. 5: Number of changed edges and vertices of each window. Dotted line signifies the landfall of Katrina.

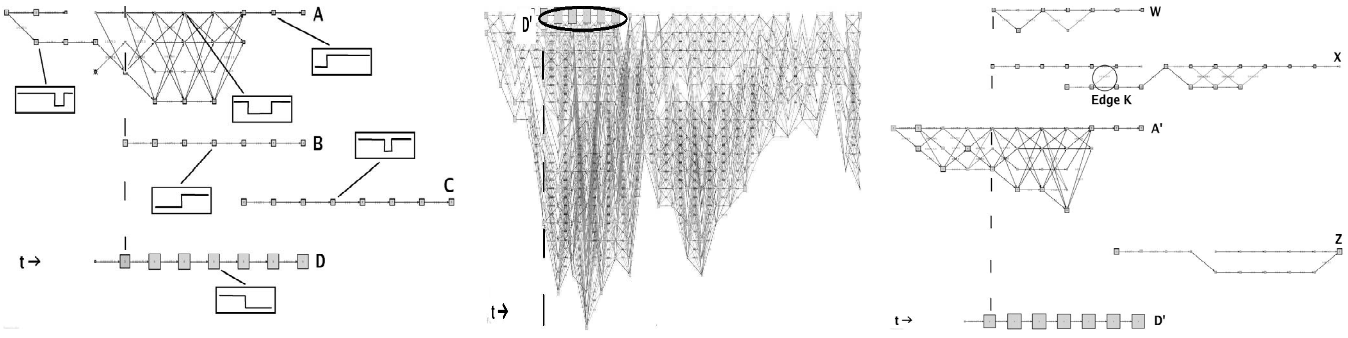
BGP is a routing protocol used to establish the forwarding tables between the routers of organisations (autonomous systems (ASs)) on the Internet. The vertices in the BGP connectivity graph represent the ASs, and the edges represent the existence of a routing path between the ASs. We extracted the BGP graph snapshots from traces of the RouteViews project [12]. When the hurricane hit in August 2005, the US BGP graph had approximately 10000 vertices and 45000 edges. We analysed 66 snapshots, which covered the period from August 28th to 31st. This included the landfall of Hurricane Katrina at New Orleans. We found that a *window size* of 8, a *window increment* of 1, and both L_{spa} and L_{tem} of 0.8 produced the most clear results. Also, to better illustrate the resultant evolution graphs, we pruned away DAGs that had less than three regions or only involved small regions containing five edges or less. The running time was less than 60 seconds, mostly dominated by I/O.

A. Event Separation

To demonstrate the difficulty of analysing the changes individually, we first analyse the number of individual changes during the landfall of Katrina in this section. We then compare and demonstrate the event separation ability of cSTAG.

Consider Figure 5, which shows the number of individual changed edges and vertices for each window. It shows a significant number of changing edges and vertices that need to be analysed, even before the landfall of Katrina. During the window immediately after the landfall of Katrina, the number of changed edges rises to nearly 300. Given that the only knowledge available for each individual changed edge is whether it appeared or disappeared between adjacent snapshots, it is difficult to determine any pattern from the individual changes. Compare this with the evolution graph produced by cSTAG (Figure 6a). The dotted vertical line represents the time of the landfall of Katrina, and each DAG is labeled with the representative waveforms of its regions. Figure 6a clearly shows the different events. For example, the DAG labeled D involves a large failure region. It has been reported [11] that a significant percentage of the Louisiana (and Mississippi) network was knocked out by Katrina. Using the whois service of ARIN [13], we found that of the 26 unique changed edges that are in the regions of DAG D, 20

[1] The k-neighbourhood set of set A is the set of edges that are within k-hops of an edge of set A.



(a) cSTAG.

(b) *temporal-only*.

(c) *spatial-only*.

Fig. 6: Evolution graphs produced by cSTAG, the temporal-only strategy, and the spatial-only strategy.

of them are connected to an AS registered in Louisiana or Mississippi. Therefore it is likely that DAG D corresponds to this reported event. Although not shown in Figure 6a, the regions of DAG B are topologically adjacent to the regions of DAG D. Yet it is correctly identified as a separate event, as DAG B has different temporal evolution (it is a recovery event). DAG A represents failure from the earlier landfall of Katrina in Florida [11], where there was an initial failure, then the event branches, due to partial recovery of different parts of the failed network at different times (see waveforms shown in figure 6a). Finally, DAG C represents another failure centred around Florida, possibly due to the second landfall of Katrina. It also involved failure and subsequent recovery changes, but its changes are delayed from those of DAG A, and hence is considered as a separate event. In summary, this analysis demonstrates cSTAG was able to separate the events and their impact areas, even if the events were either topologically adjacent or temporally similar to each other.

B. Spatial-Temporal Significance

In this section, we compare two naive region linking strategies with cSTAG. The first strategy, *temporal-only*, is to link regions that are temporally consistent and similar. The second strategy, *spatial-only*, is to link regions that are spatially close. cSTAG uses both spatial and temporal similarities. Unless otherwise stated and where applicable, we used the same parameters for the three schemes.

Figure 6b shows the evolution graph for the *temporal-only* algorithm. Apart from two small regions not shown, every other region has been incorrectly grouped into one component. For example, the regions circled and labelled D' are the same regions from DAG D in Figure 6a. Figure 6c shows the evolution graph for the *spatial-only* algorithm. It is similar to the one produced by cSTAG. However, some of the DAGs have additional edges and vertices, because some spatially close but temporally different regions have been linked to each other. For example, DAG X consists of two shorter events, incorrectly joined by the edge K. The waveforms of the source and target regions of edge K are temporally inconsistent, so should not be in the same DAG. Although Figure 6a does not show them due to lack of space, cSTAG actually identified these events as separate DAGs.

The results of these two alternative algorithms demonstrate that both spatial and temporal information needs to be considered when linking the regions.

IV. Conclusion

In this paper, we proposed a novel pattern, regions of correlated spatio-temporal change, to summarise change in evolving graphs. We developed a method, cSTAG, to discover the regions and use them as the basis for event identification. cSTAG represents the graph changes as waveforms, and then clusters these waveforms to find regions of correlated change. The regions are grouped into DAGs, and these DAGs represent the separate events affecting the evolving graph.

Analysing the snapshots taken around the time of the landfall of Hurricane Katrina, we demonstrated that cSTAG can distinguish the multiple simultaneous events occurring in the BGP connectivity graph. We have also demonstrated that using both spatial and temporal information to link regions produces better results. In future work, we hope to extend the evaluation, use adaptive windows, and employ techniques like bayesian inference to infer the evolution links between the regions.

References

[1] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graphs over time: Densification laws, shrinking diameters and possible explanations,” in *KDD*, 2005.

[2] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins, “On the bursty evolution of blogspace,” in *WWW*, 2003.

[3] J. Leskovec, A. Singh, and J. Kleinberg, “Patterns of influence in a recommendation network,” in *PAKDD*, 2005.

[4] P. Shoubridge, M. Kraetzl, W. Wallis, and H. Bunke, “Detection of abnormal change in a time series of graphs,” DSTO, Tech. Rep., 2000.

[5] P. Desikan, N. Pathak, J. Srivastava, and V. Kumar, “Incremental pagerank computation on evolving graphs,” in *WWW*, May 2005.

[6] D. Neill, A. Moore, M. Sabhnani, and K. Daniel, “Detection of emerging space-time clusters,” in *KDD*, 2005.

[7] J. Gao, G. Kar, and P. Kermani, “Approaches to building self healing systems using dependency analysis,” in *NOMS*, 2004.

[8] A. K. H. Tung, R. T. Ng, L. V. S. Lakshmanan, and J. Han, “Constraint-based clustering in large databases,” in *ICDT*, 2001.

[9] L. Portnoy, E. Eskin, and S. Stolfo, “Intrusion detection with unlabeled data using clustering,” in *DMSA*, 2001.

[10] M. Carey and D. Johnson, *Computers and Intractability*. W.H. Freeman and Company, 1979.

[11] J. Cowie, A. Popescu, and T. Underwood, “Impact of hurricane katrina on internet inrastructure,” Renesys Corporation, Tech. Rep., 2005.

[12] University of Oregan, “Route views project,” http://www.routeviews.org

[13] “American registry for internet numbers,” http://www.arin.net