A Statistical Framework for Streaming Graph Analysis

James Fairbanks David Ediger Rob McColl David A. Bader Eric Gilbert
Georgia Institute of Technology
Atlanta, GA, USA

Abstract—**In this paper we propose a new methodology for gaining insight into the temporal aspects of social networks. In order to develop higher-level, large-scale data analysis methods for classification, prediction, and anomaly detection, a solid foundation of analytical techniques is required. We present a novel approach to the analysis of these networks that leverages time series and statistical techniques to quantitatively describe the temporal nature of a social network. We report on the application of our approach toward a real data set and successfully visualize high-level changes to the network as well as discover outlying vertices.**

The real-time prediction of new connections given the previous connections in a graph is a notoriously difficult task. The proposed technique avoids this difficulty by modeling statistics computed from the graph over time. Vertex statistics summarize topological information as real numbers, which allows us to leverage the existing fields of computational statistics and machine learning. This creates a modular approach to analysis in which methods can be developed that are agnostic to the metrics and algorithms used to process the graph.

We demonstrate these techniques using a collection of Twitter posts related to Hurricane Sandy. We study the temporal nature of betweenness centrality and clustering coefficients while producing multiple visualizations of a social network dataset with 1.2 million edges. We successfully detect vertices whose triangle-forming behavior is anomalous.

I. Introduction

Social networks such as Twitter and Facebook represent a large portion of information transfer on the Internet today. Each time a new post is made, we gain a small amount of new information about the dynamics and structure of the network of human interaction. New posts reveal connections between entities and possibly new social circles or topics of discussion. Social media is a large and dynamic service; at its peak, Twitter recorded over 13,000 Tweets per second [1] and revealed recently that the service receives over 400 million Tweets per day on average [2].

Social media events – such as two users exchanging private messages, one user broadcasting a message to many others, or users forming and breaking interpersonal connections – can be represented as a graph in which people are vertices and edges connect two people representing the event between them. The nature of the edge can vary depending on application, but one common approach for Twitter uses edges to connect people in which one person “mentions” the other in a post. Edges can be directed or undirected; in this paper we analyze tweets using an undirected graph. The edge is marked with a timestamp that represents the time at which the post occurred. The format of a Twitter post makes this information accessible.

In the *massive streaming data analytics model* [3], we view the graph of social media events as an un-ending stream of new edge updates. For a given interval of time, we have the static graph, which represents the previous state of the network, and a sequence of edge updates that represent the new events that have taken place since the previous state was recorded. An update can take the form of an insertion representing a new edge, a change to the weight of an existing edge, or a deletion removing an existing edge. Because of the nature of Twitter, we do not use deletions, and the edge weights count the number of times that one user has mentioned the other.

Previous approaches have leveraged traditional, static graph analysis algorithms to compute an initial metric on the graph and then a final metric on the graph after all updates. The underlying assumption is that the time window is large and the network changes substantially so that the entire metric must be recomputed. In the massive streaming data analytics model, algorithms react to much smaller changes on smaller time-scales.

Given a graph with billions of edges, inserting 100,000 new edges has a small impact on the overall graph. An efficient streaming algorithm recomputes metrics on only the regions of the graph that have experienced change. This approach has shown large speed-ups for clustering coefficients and connected components on scale-free networks [3], [4].

Despite an algorithmic approach to streaming data, we lack statistical methods to reason about the dynamic changes taking place inside the network. These methods are necessary to perform reliable anomaly detection in an on-line manner. In this paper, we propose analyzing the values of graph metrics in order to understand the dynamic properties of the graph. This allows us to leverage existing techniques from statistics and data mining for analyzing time series.

A. Related Work

Previous research has shown that Twitter posts reflect valuable information about the real world. Human events, such as breaking stories, pandemics, and crises, affect worldwide information flow on Twitter. Trending topics and sentiment analysis can yield valuable insight into the global heartbeat.

A number of crises and large-scale events have been extensively studied through the observation of Tweets. Hashtags, which are user-created metadata embedded in a Tweet, have been studied from the perspective of topics and sentiment. Hashtag half-life was determined to be typically less than 24 hours during the London riots of 2011 [5]. The analysis of Twitter behavior following a 2010 earthquake in Chile revealed differing propagation of rumors and news stories [6]. Researchers in Japan used Twitter to detect earthquakes with high probability [7]. Twitter can be used to track the prevalence of influenza on a regional level in real time [8]. Betweenness centrality analysis applied to the H1N1 outbreak and historic Atlanta flooding in 2009 revealed highly influential tweeters in addition to commercial and government media outlets [9].

Many attempts at quantifying influence have been made. Indegree, retweets, and mentions are first-order measures, but popular users with high indegree do not necessarily generate retweets or mentions [10] These first-order metrics are traditional database queries that do not take into account topological information. PageRank and a low effective diameter reveal that retweets diffuse quickly in the network and reach many users in a small number of hops [11]. Users tweeting URLs that were judged to elicit positive feelings were more likely to spread in the network, although predictions of which URL will lead to increased diffusion were unreliable [12].

B. Graph Kernels and Statistics

We define a *graph kernel* as an algorithm that builds a data structure or index on a graph.[1] We define a *vertex statistic* as a function from the vertex set to the real numbers that depends on the edge set and any information contained by the edges, *i.e.* edge weight. Specifically, a vertex statistic should not depend on the vertex labels in the data structure that the graph is stored in. For example, a connected components algorithm is a graph kernel because it creates a mapping from the vertex set to the component labels. The function that assigns each vertex the size of its connected component is a vertex statistic. Graph kernels can be used as subroutines for the efficient computation of graph statistics. Any efficient parallel implementation of a vertex statistic will depend on efficient parallel graph kernels. Another example of a kernel/statistic pair is breadth-first search (BFS) and the eccentricity of a vertex, which is the maximum distance of v to any vertex in the connected component of v. The eccentricity of v can be computed by taking the height of the BFS tree rooted at v.

Vertex statistics are mathematically useful ways to summarize the topological information contained in the edge set. Each statistic compresses the information in the graph; however by compressing it differently, an ensemble of statistics can extract higher-level features and properties from the graph.

One implication of this framework for graph analysis is that the computation of these vertex statistics will produce a large amount of derived data from the graph. The data for each statistic can be stored as an $|V| \times |T|$ array, which is indexed by vertex set V and time steps $T = t_1, t_2, \ldots, t_T$. These dense matrices are amenable to parallel processing using techniques from high performance linear algebra. Once we have created these dense matrices of statistics, we can apply large scale data analysis techniques in order to gain insight from the graph in motion. These statistics can also be visualized over time, which is a challenge for graphs with more than thousands of vertices. These statistics can give an analyst a dynamic picture of the changes in the graph without showing the overwhelming amount of topological information directly.

[1]This is distinct from a kernel function that compares the similarity of two graphs.

Section II takes a stream of Twitter posts ("Tweets") from the time surrounding the landfall of Hurricane Sandy, a tropical storm that hit the Northern Atlantic coast of the United States, and forms a temporal social network. We compute graph metrics, including betweenness centrality, in a streaming manner for each batch of new edges arising in the network. A traditional network analysis finds that betweenness centrality selects news media and politicians as influential during an emergency. This corroborates the prior work regarding the historic Atlanta floods [9]. We show that the logarithm of betweenness centrality follows an exponential distribution for the more central vertices in the network. We show that the derivative of logarithm of betweenness centrality is also statistically well-behaved. This informs our streaming statistical analyses, and we quantify a linear relationship in statistic values over time. Section III studies the changes in the network using derivative analysis and the correlation of statistic with its past values. Section IV uses triangle counts and local clustering coefficients to find rare vertices by combining topological and temporal information. Multivariate techniques are used to look for anomalous behavior in the network. A related approach is taken in [13] using non-negative matrix factorization; however, we separate the graph computation from the statistical learning computation.

II. Global Views of the Data

While it is straightforward to generate large, synthetic social networks for exploring the scalability and performance characteristics of new algorithms and implementations, there is no substitute for studying data observed in the real world. For the experiments presented in this work, we study real social network data. Specifically, we assembled a corpus around a single event maximizing the likelihood of on-topic interaction and interesting structural features. At the time, there was great concern about the rapid development of Hurricane Sandy. Weather prediction gave more than one week of advanced notice and enabled us to build a tool chain to observe and monitor information regarding the storm on a social network from before the hurricane made landfall through the first weeks of the recovery effort.

In order to focus our capture around the hurricane, we selected a set of hashtags (user-created metadata identifying a particular topic embedded within an individual Twitter post) that we identified as relevant to the hurricane. These were *#hurricanesandy, #zonea, #frankenstorm, #eastcoast, #hurricane,* and *#sandy*. Zone A is the evacuation zone of New York City most vulnerable to flooding.

We used a third party Twitter client to combine our hashtags into a single stream containing a Java Script Object Notation

(JSON) representation of the Tweets along with any media and geolocation data contained in the Tweets. Starting from the day before the storm made landfall, we processed nearly 1.4 million public Twitter posts into an edge list. This edge list also includes cases where a user "retweeted" or reposted another user's post, because retweets mention the author of the original Tweet similar to a citation. The dataset included over 1,238,109 mentions from 662,575 unique users. We construct a graph from this file in which each username is represented as a vertex. The file contains a set of tuples containing two usernames which are used to create the edges in the graph. The temporal ordering of the mentions was maintained through the processing tool chain resulting in a temporal stream of mention events encoded as graph edges.

The edge stream is divided into batches of 10,000 edge insertions. As each batch is applied to the graph, we compute the betweenness centrality, local clustering coefficient, number of closed triangles, and degree for each vertex in the graph. Each algorithm (statistic) produces a vector of length $|V|$ that is stored for analysis. The computation after each batch considers all edges in the graph up to the current time. As the graph grows, the memory will eventually become exhausted, requiring edges to be deleted before new edge insertions can take place. We do not yet consider this scenario, but propose a framework by which we can analyze the graph in motion at a given moment in time. [2]

A. *Betweenness Centrality*

Centrality metrics on static graphs provide an algorithmic way to measure the relative importance of a vertex with respect to information flow through the graph. Higher centrality values generally indicate greater importance or influence. Betweenness centrality [14] is a specific metric that is based on the fraction of shortest paths on which each vertex lies. The computational cost of calculating exact betweenness centrality can be prohibitive; however, approximating betweenness centrality is tractable and produces relatively accurate values for the highest centrality vertices [9], [15]. It is known that betweenness centrality follows a heavy tail distribution. In order to account for the heavy tail, we examine the logarithm of the betweenness centrality score for each vertex. This is analogous to measuring earthquakes on the Richter magnitude scale. Since many vertices have zero or near zero betweenness centrality we add one before taking the logarithm and discard vertices with zero centrality.

For this data set at time 98 we find that the right half of the distribution of logarithm of betweenness centrality is exponential with location 5.715 and $\lambda = 1.205$ Since betweenness centrality estimates are more accurate for the high centrality vertices [16], we focus our analysis on the vertices whose centrality is larger than the median. The cumulative distribution function (*CDF*) is $1 - exp\,[-\lambda x]$. Figure 1 shows both the empirical *CDF* and the modeled *CDF* for the log(betweenness centrality). It is apparent in the figure that

[2]See http://www.stingergraph.com/ for code and data

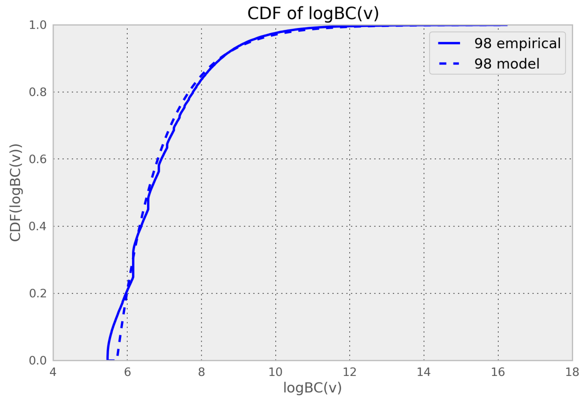

Fig. 1. The cumulative distribution function for logarithm of betweenness centrality empirical (solid) and exponential best fit (dashed)

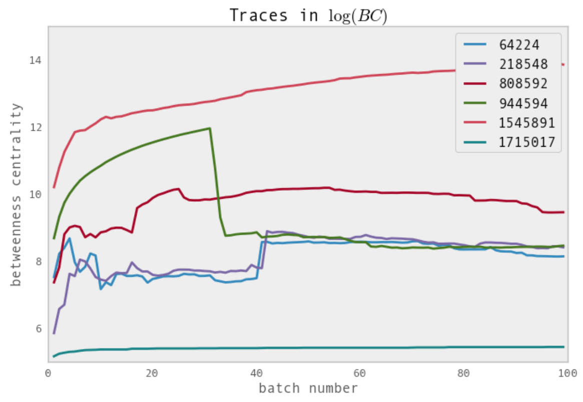

Fig. 2. Traces of betweenness centrality value for selected vertices over time.

the exponential distribution is a good fit for the right tail. We can use the *CDF* to assign a probability to each vertex, and these probabilities can be consumed by an ensemble method for a prediction task. This will allow traditional machine learning and statistical techniques to be combined with high performance graph algorithms while, maintaining the ability to reason in a theoretically sound way.

III. Temporal Analysis

A. *Observing a Sample of Vertices*

In Figure 2, we trace the value of betweenness centrality for a selection of vertices over time. Each series in this figure is analogous to a seismograph, depicting the fluctuations in centrality over time for a particular vertex. In the sociological literature, this corresponds to a longitudinal study. It is clear that there is a significant amount of activity for each vertex. Such a longitudinal study of vertices can be performed for any metric that can be devised for graphs.

B. *Analysis of derivatives*

Tracking the derivatives of a statistic can provide insight into changes that are occurring in a graph in real time. We

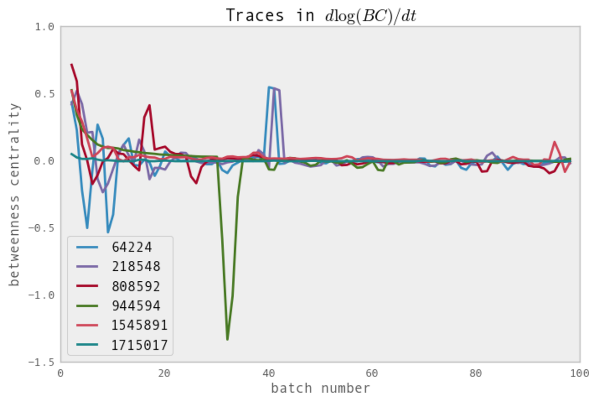

Fig. 3. The derivative of the logarithm of betweenness centrality values for selected vertices.

define the (discrete) derivative of a vertex statistic for a vertex v at time t using the following equation where $b(t)$ is the number of edges inserted during batch t. Note that the derivative of a vertex statistic is also a vertex statistic.

$$\frac{df}{dt}(v,t) = \frac{f(v,t+1) - f(v,t-1)}{b(t) + b(t-1)}$$

When concerned about maximizing the number of edge updates that can be processed per second, fixing a large batch size is appropriate. However when attempting to minimize the latency between an edge update and the corresponding update to vertex statistics, the batch size might vary to compensate for fluctuations in activity on the network. Dividing by the number of edges per batch accounts for these fluctuations. For numerical or visualization purposes one can scale the derivative by a constant.

For example, Figure 3 shows the derivative of logarithm of betweenness centrality. These traces indicate that changes in the betweenness centrality of a vertex are larger and more volatile at the beginning of the observation and decrease in magnitude over time. The reason for taking logs before differentiation is that it effectively normalizes the derivative by the value for that vertex.

Because the temporal and topological information in the graph is summarized using real numbers, we can apply techniques that have been developed for studying measurements of scientific systems to graphs. This includes modeling the sequences of measurements for prediction. We can then use robust statistics to determine when a vertex differs significantly from the prediction produced by the model. Here we can apply techniques from traditional time series analysis to detect when the time series for a vertex has changed dramatically from its previous state. This might indicate that the underlying behavior of the vertex has also changed. Others have been able to directly detect changes in a time series using streaming density estimation [17]. This method makes weak assumptions about the data and thus could be useful in practice. If the time series of a vertex has a change point, then the vertex can be flagged as interesting for this time step and processed by an analyst or more sophisticated model.

One can model the time series for each vertex as a stochastic process with a set of parameters for each vertex. These stochastic processes could be used to predict future values of the time series. These predictions would allow an advertiser to strategize about the future of the network. Modeling vertices in this fashion would also allow detection of when vertices deviate from their normal behavior.

The data can be examined in a cross sectional fashion by examining the distribution of the derivatives at a fixed point in time. By grouping the vertices by the sign of their derivatives and counting, we can see that more vertices decrease in centrality than increase in a given round.

Since the derivative of a vertex statistic is another vertex statistic, these derivatives can be analyzed in a similar fashion. By estimating the distribution of $\frac{df}{dt}$ for any statistic we can convert the temporal information encoded in the graph into a probability for each vertex. These probabilities can then be used as part of an ensemble method to detect anomalous vertices. Modeling these differences in aggregate allows for the detection of vertices that deviate from the behavior of typical vertices. When f is a measure of influence such as centrality, these extreme vertices are in the process of “going viral”, since their influence is growing rapidly.

C. Correlation

If we are to predict statistic values into the future based on current and past history, then we must identify a pattern in the temporal relationship that can be exploited. For any statistic, we can look at the Pearson correlation between the values at time t and time $t+k$ for various values of k and quantify the strength of a linear relationship. This is not a method to predict statistic values into the future. Instead this quantifies the success of linear regression. In order to demonstrate that this technique is agnostic to the statistic that has been measured on the graph, we proceed using the local clustering coefficient metric [18]. Local Clustering of a vertex v is the number of triangles centered at v divided by $degree(v)(degree(v)-1)$. The clustering coefficient is a measure of how tightly knit the vertices are in the graph.

We can leverage linear regression to find vertices that are significant. Once we have computed the statistic at two distinct time-steps, the line of best fit can be used to determine what a typical change was due to the edges that were inserted. The distance from each vertex to that best-fit surface could be used as a score for each vertex. The vertices with large scores could then be processed as anomalous. The correlation function $\rho_f(t,t+k)$ is a measure of how well a linear surface fits the data.

Let $\rho_f(t,t+k)$ denote the correlation between $f(v)$ measured at time t and $f(v)$ measured at time $t+k$. For this graph, $\rho_f(t,t+k)$ is increasing in t and decreasing in k. There is a linear decay in the Pearson correlation coefficient as the gap size k increases.

We define relative impact of a set of insertions as the number

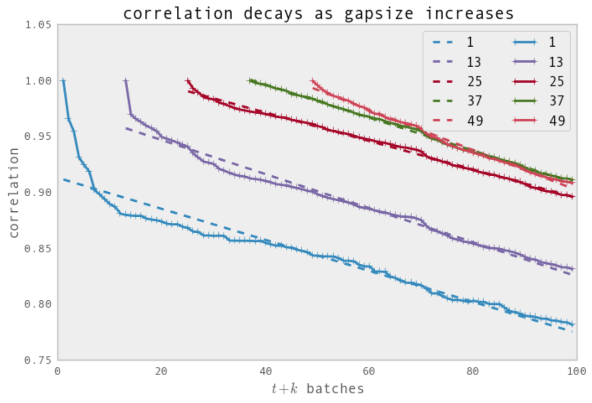

Fig. 4. Pearson correlation of local clustering coefficient values for inter-batch gaps of 10 (top) and a line of best fit.

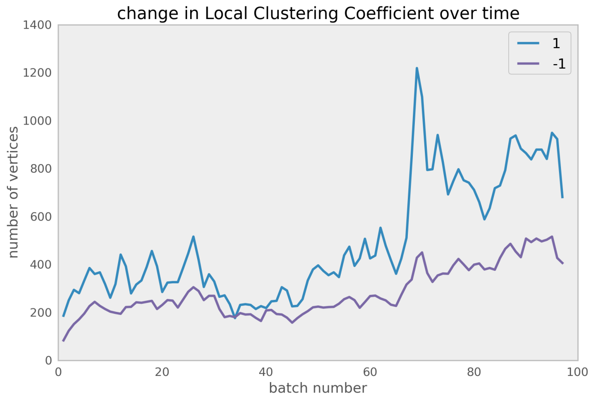

Fig. 5. Counting vertices by sign of their derivative at each time step.

of edge changes in the set divided by the average size of the graph during the insertions. By fixing the batch size to a constant b, and the initial graph size to 0, we obtain the following equations for the relative impact of the batch at time t after a gap of k batches.

$$RI(t,k) = \frac{2bk}{NE_t + NE_{t+k}} = \frac{k}{t+k/2} = \left(\frac{t}{k} + \frac{1}{2}\right)^{-1}$$

Considering these equations enables reasoning about the correlation of a statistic over time because relative impact measures the magnitude of the changes to the graph. For a fixed gap k, as t grows the relative impact of k batches converges to 0. For a fixed time t, as the gap k grows, the relative impact of those batches grows. If we assume that the correlation between a statistic at time t and time $t+k$ depends on $RI(t,k)$ then we expect $\rho_f(t,t+k)$ to increase as t grows and decrease as k grows. Figure 4 shows the correlation in clustering coefficient for the graph under consideration. The curves shown are $\rho_f(t,t+k)$ where the series labels indicate t and the horizontal axis indicates the time of the second measurement which is $t+k$. The dashed lines are the best linear fits for each series. Since moving to the right decreases the correlation and moving to increasing series increases the correlation, the model is validated.

Another way to analyze a streaming graph using these statistics, is to look at the derivative. We can measure the overall activity of a graph according to a statistic such as clustering coefficient by counting the number of vertices that change their value in each direction. For clustering coefficient this is shown in Figure 5. One observation is that more vertices have increasing clustering coefficient than decreasing clustering coefficient. We also learn that only a small fraction of vertices change in either direction. Monitoring these time series could alert an analyst or alarm system that there is an uptick in clustering activity in the graph. The increase in the number of vertices with increasing clustering coefficient around batch 70 could correspond to communities forming around a set of vertices which all increase their clustering coefficient.

IV. Multivariate methods for outlier detection

Because anomaly detection is a vague problem, we focus on outlier detection, which is a more well-defined problem. The outliers of a data set are the points that appear in the lowest density region of the data set.

One method for finding outliers is to assume that the data are multivariate Gaussian and use a robust estimate of mean and covariance [19] – a method known as the elliptic envelope. This is appropriate when the data is distributed with light tails and one mode. The one class support vector machine (SVM) can be used to estimate the density of an irregular distribution from a sample. By finding the regions with low density, we can use an SVM to detect outliers [20].

We seek a method to apply these multivariate statistical methods to our temporal graph data. Because we have been computing the triangle counts and local clustering coefficient for each vertex in an on-line fashion, each vertex has a time series. This time series can be summarized by computing moments.

We extract the mean and variance of the original local clustering coefficient series. In order to capture temporal information, we use the derivative of the local clustering coefficient and extract the mean and variance. Summary statistics for the derivatives are taken over non zero entries because most vertices have no change in local clustering coefficient at each time step. These summary statistics are used as features that represent each vertex. This is an embedding of the vertices into a real vector space that captures both topological information and the temporal changes to the network. This embedding can be used for any data mining task. Here we use outlier detection to illustrate the usefulness of this embedding.

Once these features are extracted, the vertices can be displayed in a scatter plot matrix. This shows the distribution of the data for each pair of features. These scatter plots reveal that the data is not drawn from a unimodal distribution. Because the robust estimator of covariance requires a unimodal distribution this eliminates the elliptic envelope method for outlier detection.

Using a single class support vector machine with the Gaussian Radial Basis Kernel, we are able to estimate the support of the data distribution. The ν parameter of the SVM selects a fraction of the data to label as outliers. Because the SVM is sensitive to scaling of the data, we whiten the data so that is has zero mean and unit standard deviation. By grouping the data into inliers and outliers, we see that the two distributions are distinct in feature space.

Figure 6 shows a scatter matrix with the inlying vertices in blue and the outliers in red. We can see that in any pair of dimensions some outliers are mixed with inliers. This indicates that the SVM is using all of the dimensions when forming a decision boundary. The diagonal plots show normalized histograms in each dimension with inliers in blue and outliers in red. These histograms show that the distribution of the inliers differs significantly from the distribution of the outliers. This indicates that the SVM is capturing a population that is distinct from the majority population.

V. Conclusion and Future Work

Social media events are an insight into events occurring in the real world. Processing data in a streaming, or on-line, manner adds value to the insight. The goal of this research is to quickly provide actionable intelligence from the stream of graph edges. Modeling the graph directly using standard machine learning techniques is difficult because of the enormous size, volume, and rate of the incoming data. Rather, we embed the graph in a Real vector space by computing multiple graph algorithms that each capture a different aspect of the network topology. Then we can apply computational statistics and machine learning in order to extract insight. Our contribution is a framework for connecting machine learning techniques with graphs arising from massive social networks using high performance graph algorithms that combines topological and temporal information.

We studied a collection of Tweets observed during Hurricane Sandy. A preliminary study of the entire corpus of Tweets revealed that betweenness centrality isolated local news and government agencies. A parametric analysis revealed an exponential distribution for logarithmic of betweenness centrality for vertices greater than the median.

Taking discrete derivatives of the values over time encodes temporal graph information in an efficient manner. We exposed an opportunity to model individual vertices. From the derivatives, we determined that the distribution of positive changes differs from negative changes for betweenness centrality and clustering coefficients. We reveal that relatively few vertices change their values for any given batch of edge insertions.

We defined the relative impact of a batch of edge updates on a graph and postulated that this will measure the relationship between a vertex statistic and its future values. This analysis predicted two trends in correlation of local clustering coefficient, and we observed these trends in the Twitter corpus. This suggests that it is possible to develop predictive models for vertex statistics.

Detecting anomalous activity in a network is a key capability for streaming graph analysis. Two approaches include labeling events or actors as anomalous; we focus on labeling actors in a statistically rigorous manner. One difficulty is giving a precise definition of anomalous behavior. We use vertex statistics (and their derivatives) to define vertex behavior, and then use outlier detection in the traditional way to detect vertices whose behavior differs significantly from the majority. We demonstrate this approach on a Twitter corpus by using a one class support vector machine where the behavior of interest is formation of triangles. This approach finds a partition of the vertices such that the inliers are tightly clustered and the outliers are diffuse.

Acknowledgments

This work was supported in part by the Pacific Northwest National Lab (PNNL) Center for Adaptive Supercomputing Software for MultiThreaded Architectures (CASS-MT) and by DARPA-BAA-11-64.

References

[1] Twitter, "#Goal," April 2012, http://blog.uk.twitter.com/2012/04/goal.html.

[2] ——, "Celebrating #Twitter7," March 2013, http://blog.twitter.com/2013/03/celebrating-twitter7.html.

[3] D. Ediger, K. Jiang, J. Riedy, and D. A. Bader, "Massive streaming data analytics: A case study with clustering coefficients," in *4th Workshop on Multithreaded Architectures and Applications (MTAAP)*, Atlanta, Georgia, Apr. 2010.

[4] D. Ediger, E. J. Riedy, D. A. Bader, and H. Meyerhenke, "Tracking structure of streaming social networks," in *5th Workshop on Multithreaded Architectures and Applications (MTAAP)*, May 2011.

[5] K. Glasgow and C. Fink, "Hashtag lifespan and social networks during the london riots," in *Social Computing, Behavioral-Cultural Modeling and Prediction*, ser. Lecture Notes in Computer Science, A. Greenberg, W. Kennedy, and N. Bos, Eds. Springer Berlin Heidelberg, 2013, vol. 7812, pp. 311–320.

[6] M. Mendoza, B. Poblete, and C. Castillo, "Twitter under crisis: can we trust what we rt?" in *Proceedings of the First Workshop on Social Media Analytics*, ser. SOMA '10. New York, NY, USA: ACM, 2010, pp. 71–79.

[7] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," in *Proceedings of the 19th international conference on World wide web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 851–860.

[8] A. Signorini, A. M. Segre, and P. M. Polgreen, "The use of twitter to track levels of disease activity and public concern in the u.s. during the influenza a h1n1 pandemic," *PLoS ONE*, vol. 6, no. 5, p. e19467, 05 2011.

[9] D. Ediger, K. Jiang, J. Riedy, D. A. Bader, C. Corley, R. Farber, and W. N. Reynolds, "Massive social network analysis: Mining twitter for social good," *Parallel Processing, International Conference on*, pp. 583–593, 2010.

[10] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, "Measuring user influence in twitter: The million

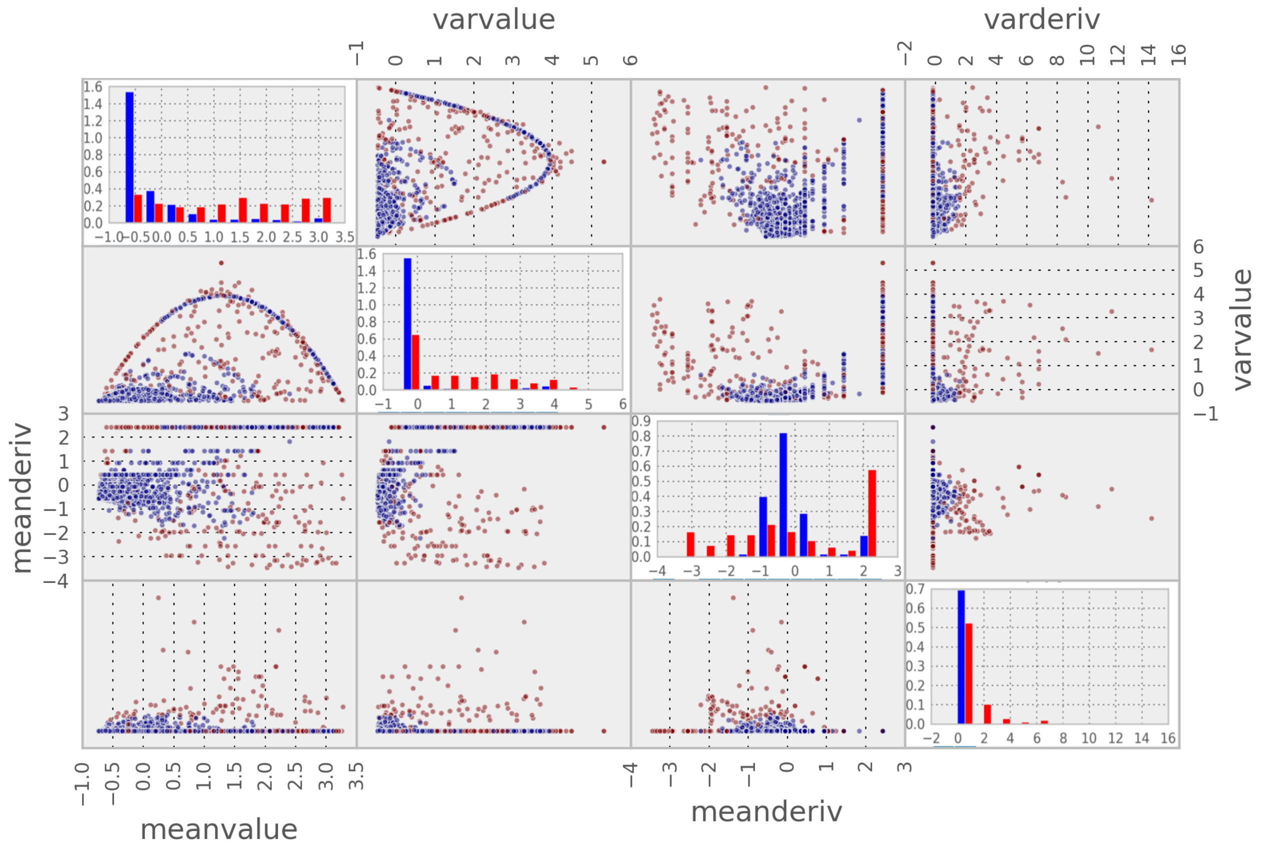

Fig. 6. Scatter plot matrix showing the outliers (red) and normal data (blue)

follower fallacy," in *in ICWSM Š10: Proceedings of international AAAI Conference on Weblogs and Social*, 2010.

[11] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *19th World-Wide Web (WWW) Conference*, Raleigh, North Carolina, Apr. 2010.

[12] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: quantifying influence on twitter," in *Proceedings of the fourth ACM international conference on Web search and data mining*, ser. WSDM '11. New York, NY, USA: ACM, 2011, pp. 65–74.

[13] R. A. Rossi, B. Gallagher, J. Neville, and K. Henderson, "Modeling dynamic behavior in large evolving graphs," in *WSDM*, 2013, pp. 667–676.

[14] L. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

[15] D. Bader, S. Kintali, K. Madduri, and M. Mihail, "Approximating betweenness centrality," in *Proc. 5th Workshop on Algorithms and Models for the Web-Graph (WAW2007)*, ser. Lecture Notes in Computer Science, vol. 4863. San Diego, CA: Springer-Verlag, December 2007, pp. 134–137.

[16] R. Geisberger, P. Sanderst, and D. Schultest, "Better approximation of betweenness centrality," in *Proceedings of the Tenth Workshop on Algorithm Engineering and Experiments and the Fifth Workshop on Analytic Algorithmics and Combinatorics*, vol. 129. Society for Industrial & Applied, 2008, p. 90.

[17] Y. Kawahara and M. Sugiyama, "Change-point detection in time-series data by direct density-ratio estimation," in *Proceedings of 2009 SIAM international conference on data mining (SDM2009)*, 2009, pp. 389–400.

[18] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[19] P. J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, no. 3, pp. 212–223, 1999.

[20] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.