EVENT DETECTION IN TIME SERIES OF MOBILE COMMUNICATION GRAPHS

Leman Akoglu*, Christos Faloutsos
Carnegie Mellon University
Pittsburgh, PA, 15213

ABSTRACT

Anomaly and event detection has been studied widely for having many applications in fraud detection, network intrusion detection, detection of epidemic outbreaks, and so on. In this paper we propose an algorithm that operates on a time-varying network of agents with edges representing interactions between them and (1) spots "anomalous" points in time at which many agents "change" their behavior in a way it deviates from the norm; and (2) attributes the detected anomaly to those agents that contribute to the "change" the most. Experiments on a large mobile phone network (of 2 million anonymous customers with 50 million interactions over a period of 6 months) shows that the "change"-points detected by our algorithm coincide with the social events and the festivals in our data.

1. INTRODUCTION

Anomaly detection has been studied widely in many settings such as "anomalous point detection" on clouds of multi-dimensional points, spatio-temporal "anomalous pattern detection", "change-point detection" on a sequence of time series of data, and so on with many applications such as intrusion detection in networks [Sequeira et. al. 2002], detection of medical insurance claim fraud, credit card fraud, electronic auction fraud [Bolton et. al. 2002, Chau et. al. 2006], fault detection in engineering systems [Fujimaki et. al. 2005] as well as many others. In this paper, we focus on change-point detection in time-varying graph data.

The problem of discovering change-points at which properties of time-series data change significantly has also attracted a lot of interest in the research community [Basseville et. al. 1993, Brodsky et. al. 1993, Gustafsson 2000, Yamanishi et. al. 2002 Kifer et. al. 2004, Kawahara et. al. 2007]. This problem is also referred as event detection [Guralnik and Srivastava 1999].

Although the change-point detection problem has been actively studied in the statistics and the data mining communities over the last several decades, there has been much less focus on change-point detection particularly *in graph data*. More recently, [Ide and Kashima 2004] developed an eigen-vector based algorithm to detect faults in multi-tier Web-based systems represented as a time sequence of graphs. Another set of research [Bunke et. al. 1998, Shoubridge et. al. 2002] derives "distance functions" between a pair of graphs, compute distances between consecutive graphs in a given sequence, and finally apply traditional anomaly detection methods on the time series of distance values. [Sun et. al. 2007] propose a parameter-free algorithm to discover communities in streams of graph and flag points in time as discontinuity points when the community structure changes significantly.

In this paper, we propose an algorithm to spot change-points in a time-varying graph at which many nodes deviate from their normal "behavior". In a nut-shell our method works as follows. We first extract time sequence of several network features for all nodes in the graph. Next we build a correlation matrix representing the correlation of "behavior" between all pairs of nodes in the graph over a certain time window. Then, we derive a "behavior" vector of all nodes and compare it to recent past "behavior" vectors detected over several previous time windows. If the current "behavior" is found to be significantly different than recent past, we flag the current time window as anomalous and report as an event has occurred.

To demonstrate the effectiveness of our method, we study the texting behavior of users of an anonymous mobile network in a large city in India. In this *who-texts-whom network*, nodes represent the users and edges represent the SMS interactions between them. The data consists of six months' of activity and is therefore time-varying. Also, the edges are weighted, weights denoting the total number of SMSs sent/received between individual pairs. More specifically, the SMS network constructed from the SMS records includes over 2 million users with 50 million SMS interactions between them over a period from Dec. 1, 2007 to May 31, 2008. Given this large, time-varying network, the main questions we answer in this paper are the following:

(1) *At what points in time many of the nodes in a given time-varying graph change their behavior significantly?*

(2) *Can we attribute the change to specific nodes, that is, can we characterize which nodes change in behavior the most?*

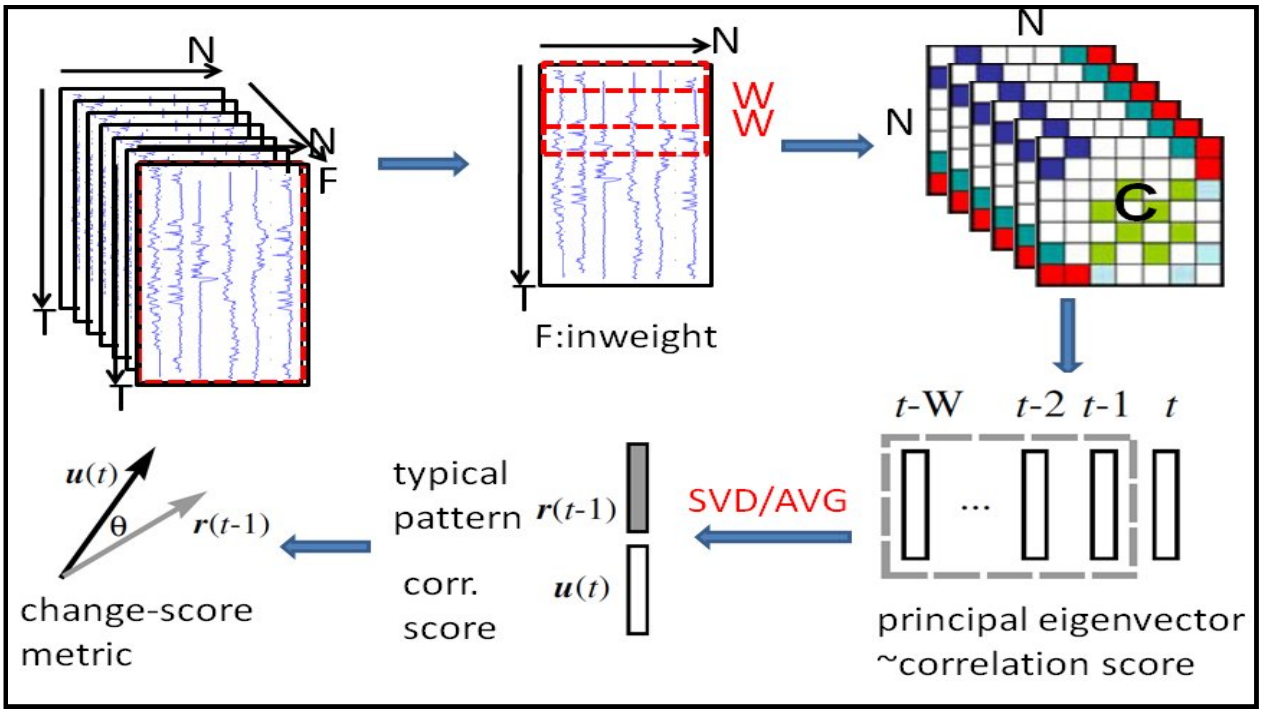

Fig. 1. The Work-flow of Change-Point Detection

3. OUR METHOD

3.1 Feature Extraction from Nodes

In order to find patterns that nodes of a graph follow, we characterize the nodes with several features so that each node becomes a multi-dimensional point. In particular, *each node is summarized by a set of features* extracted from its *egonet* (egonet of a node includes the node itself, its neighbors, and all the interactions between these nodes). The 12 features considered in this work are as follows: 1) in-degree, 2) out-degree, 3) in-weight, 4) out-weight, 5) number of neighbors, 6) number of reciprocal neighbors, 7) number of triangles, 8) average in-weight, 9) average out-weight, 10) maximum in-weight, 11) maximum out-weight, and finally 12) maximum weight ratio on reciprocated edges in the egonet.

3.2 Change-Point Detection

The flow of our method to detect change-points in the behavior of nodes is illustrated in Figure 1. This method is similar to [Ide and Kashima 2004], but differs in the construction of the "dependency" matrices C (See top right in Figure 1) as follows.

Here, the data we study looks like the **3-D** tensor on the top left of Figure 1, where **T** denotes the number of time ticks (T=183 days), **N** denotes the number of nodes in our graph (N=2M customers), and **F** denotes the number of features extracted for each node (F=12 as described in Section 3.1). To start with, we take one "slice" of this **3-D TxNxF** tensor for a particular feature F_i, say in-weight, which is a **TxN** matrix (See top middle in Figure 1). Next, we define a window of size W over the time-series of values of all nodes for that particular feature F_i. Then for pair of nodes, we compute the correlation between their time-series vectors over the window of size W using Pearson's rho as follows.

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

In the above equation, X and Y are the length-W vectors for node pair (X, Y). So, for each window we construct a correlation matrix **C**, where $C_{x,y}$ = rho(x,y) over window W. Next, we slide the window down one time tick (day) and compute the correlations over the next window of W time ticks. Similarly we keep repeating this process until we reach the end of our data. To be representative and given the periodic behavior of human nature, we chose the size W as 7 days (one week). As a result, we end up constructing 177 **C** matrices (See top right in Figure 1).

By the Perron-Frobenius theorem (1907, 1912), the largest (principal) eigenvector of each of the **C** matrices is positive. The value for each node in the eigenvector can be thought as the "activity" of that node; that is, the more correlated a node is to the majority of the nodes, the higher its "activity" value will be. Here, we call each such

eigenvector as the "eigen-behavior" of all the nodes in the graph on the whole.

3.3 Metric to Score Time Points for "anomalousness"

After finding all the eigenvectors for all the 177 **C** matrices, the change-point in the "eigen-behavior" of the nodes is found as follows. For the eigenvector computed at time say **t** denoted by **u(t)**, we compute a "typical eigen-behavior" denoted by **r(t-1)** from the last **W'** eigenvectors back in time (See bottom right in Figure 1). We experimented with two different ways to compute the mentioned typical eigen-behavior. Firstly, we simply took the arithmetic average of all previous **W'** eigenvectors. Secondly, we constructed a new **NxW'** matrix, each column being an eigenvector over that window, and then we computed the left singular vector of that new matrix using SVD decomposition. Similar to the principal eigenvector for square positive matrices, the left singular vector of a positive non-square matrix yields an average "behavior" score for all nodes. One can think of the left singular vector as the *weighted* average of eigenvectors in window W'.

Finally, after we obtain the "typical eigen-behavior" for each **C** matrix (for each week) using either SVD or regular averaging, the "eigen-behavior" **u(t)** computed at time **t** is compared to the "typical eigen-behavior" **r(t-1)** by taking the dot-product of those two unit vectors. The change metric we used is $Z = (1-u^T r)$. Here, if **u(t)** is perpendicular to **r(t-1)**, then their dot-product gives a value of 0 (Z=1), whereas if **u(t)** is exactly the same as **r(t-1)**, then their dot-product gives a value of 1 (Z=0). Therefore, Z takes values between 0 and 1 and a higher value of Z indicates a change-point and is flagged accordingly (See bottom left in Figure 1).

4. EMPIRICAL STUDY

We start by looking at the distribution of correlation values $C_{x,y}$ in the **C** matrices. Figure 2 shows the histogram as well as the CDF of $C_{x,y}$ values for two different days, Dec. 1 and Dec. 26, for F:in-weight.

Here, one observation is that the distribution of correlations between time-series of nodes is skewed as might be expected. Surprisingly, though, it is skewed towards large values. That is, there are lots of pairs with correlation score close to or equal to **1**. This happens because over the time window **W** of 7 days, most of the nodes have no activity –their W-length vectors are all 0's and thus the pair-wise correlations of such 0 vectors are computed to be 1. This suggests that the nodes have bursty activity where nodes have no activity for several weeks and have activity at bursts.

Another observation from Figure 2 is that the total number of correlation scores 1 reduces in Dec. 26 compared to Dec. 1, suggesting for no activity weeks for fewer nodes, that is, more nodes become active during the week of Dec. 26. This is expected as this week is the New Year week. We note that the CDF distributions for these two days also look different. These observations strengthen our belief of studying correlations between behaviors of nodes would be important in detecting the change-points in our data.

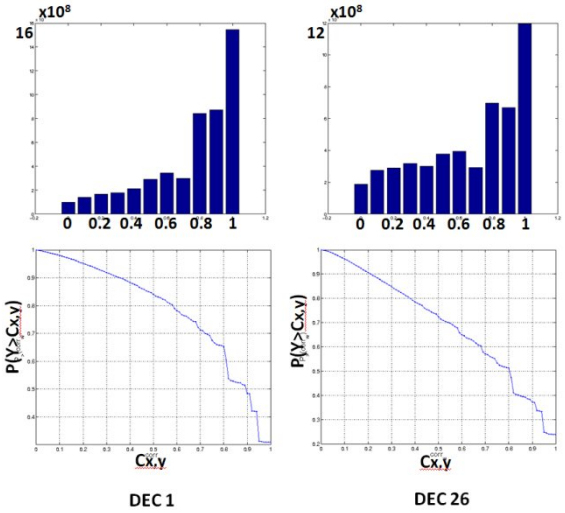

Fig. 2. (top) Histogram and (bottom) CDF distribution of correlation scores $C_{x,y}$ for (left) Dec. 1 and (right) Dec. 26 using F: in-weight.

Next, we compare the results when using SVD versus taking the regular average (AVG) for computing the "typical eigen-behavior" **r(t-1)** of earlier eigenvectors over a window of **W'** (See Section 3.3). Figure 3 shows the so-called **Z** scores computed (1) when **r(t-1)** is computed with SVD (in blue bars) versus (2) when **r(t-1)** is computed by simply taking the average (in red lines) for four different values of **W'**, (from left to right, top to bottom) {5, 7, 20, 50}. Notice that the red line almost exactly follows the blue bars. This means that SVD is giving equal weight (importance) to all W' eigenvectors in the past same as the AVG does. Therefore, since computing the average is less expensive, we will use the AVG to compute **r(t-1)** in the rest of our experiments.

We also note that in Figure 3 the Z scores follow somewhat a similar trend when different window sizes are considered. However, the larger the window gets, the more aggregated the results become (notice that the four spikes for W={5,7} reduce to two spikes only for W={20,50}). Thus in the rest of our experiments, we will use W = 5 over which the **r(t-1)** vector is computed by using AVG.

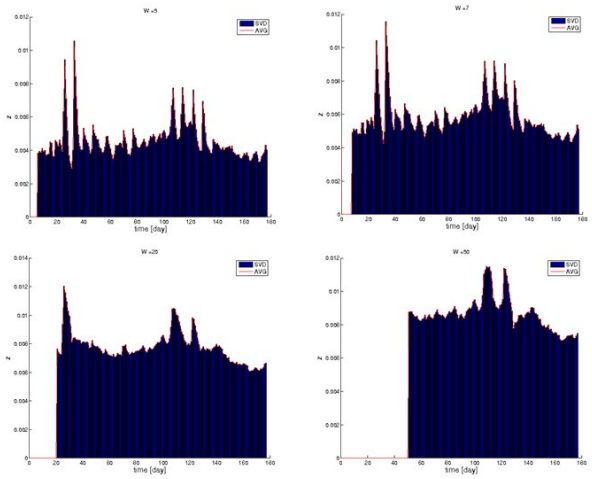

Fig. 3. Z scores computed when the typical eigen-behavior vector r(t-1) is computed by taking the SVD (blue bars) versus the regular AVG (red lines) for (from left to right top to bottom) W = {5, 7, 10, 20}. Notice AVG is very similar to SVD.

4.1 Detected Change-Points in Time

Here, after computing the Z scores for all time ticks as was explained above, we simply report the top k (k=10) days with the highest Z scores. In an online setting, this would be picking a threshold t and flag the days with a Z score greater than t.

In Figure 4 we show the top 10 time ticks with the highest Z scores in red bars when feature values F are taken to be the "in-weight". Here, we observe that the week of Christmas (Dec. 26 - Jan. 1) is marked with the *second* highest Z score. This shows that during this week many people changed their behavior in terms of receiving SMSs, probably started receiving more than usual number of SMSs. Interestingly, although our data is collected in India and most people are not Christian, many would indeed "celebrate" the "Christian New Year". We note that the reason the week starting on Jan 2 is marked with the *highest* Z score is because the week following Christmas is yet another change-point at which things go back to "normal".

Another surprising finding is the time tick with the *third* highest Z score which is April 7. Similar to the week of Jan 2, this week is also a change-point at which things turned back to "normal". The actual interesting day here is indeed April 6. Our data spans months both in 2007-2008, and according to http://www.infoplease.com/ipa/A0777465.html, April 6 in 2008 corresponds to the "Hindi New Year".

These results suggest that our method is effective in finding points in time at which the collective behavior of the nodes deviate from the recent past.

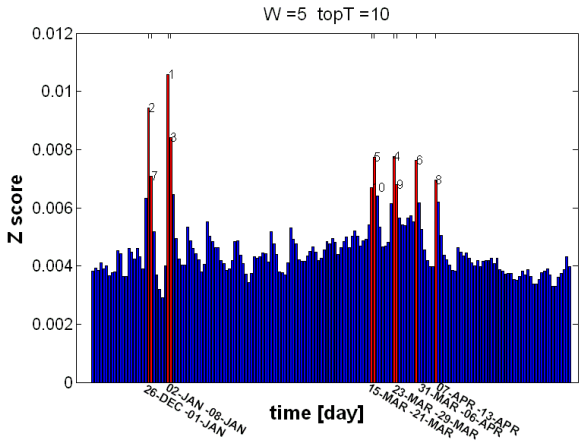

Fig. 4. Top 10 time points flagged by our method (red bars) for feature F: in-weight. "Christmas" and "Hindi New Year" were successfully detected as major change-points.

For sanity check, we also performed experiments using other features such as "numrecip": number of reciprocated edges and "out-degree": number of contacts SMSs sent. Here we conjectured that these two features would be correlated with "in-weight" and for example in New Year not only people would 'receive' many SMSs (affecting in-weight) but they would also 'reply' to them returning good wishes (affecting both numrecip and out-degree). In that sense, we wanted to check whether our method would flag similar time ticks when these two features are used.

Figure 5 shows that our method in fact flags the same time points including the weeks of Dec 26 and April 6 also when "numrecip" and "out-degree" features are used. Moreover, the spikes in the Z scores are even clearer when these features are used. This is intuitive in the sense that even though the "in-weight" (number of SMSs received) is expected to increase on days such as Christmas and New Year, the number of reciprocated interactions are expected to increase even more (people tend to reply to celebration messages on such days much more than to messages for regular communication).

We also compared the results of our method to the results we obtain by using the sheer volume at each time tick. In particular, we computed the total number of SMSs received (in-weight) per day and marked the top k=10 time points for which the most number of SMSs were received in total. We repeated the same process for total number of reciprocal replies (numrecip) and total number of out-going contacts (out-degree). We show the results in Figure 6 where the red bars depict the top 10 time points with highest volume. We observe that just the total volume for all three features was enough to detect change on for example Dec. 31 and Jan 1. However, we realize that the points reported for each feature also partly differ

from each other and are not as consistent as the earlier results. For instance April 6, even though was detected as a change-point using features "numrecip" and "out-degree", was not detected using "in-weight". The main reason behind these observations is that our method considers every person in the network individually and flags change-points if the majority of them change their "normal" behavior whereas the total volume considers the aggregated behavior. The aggregated data loses information in the individual level and thus is prone to flag change-points if only a few people change their behavior sufficiently a lot.

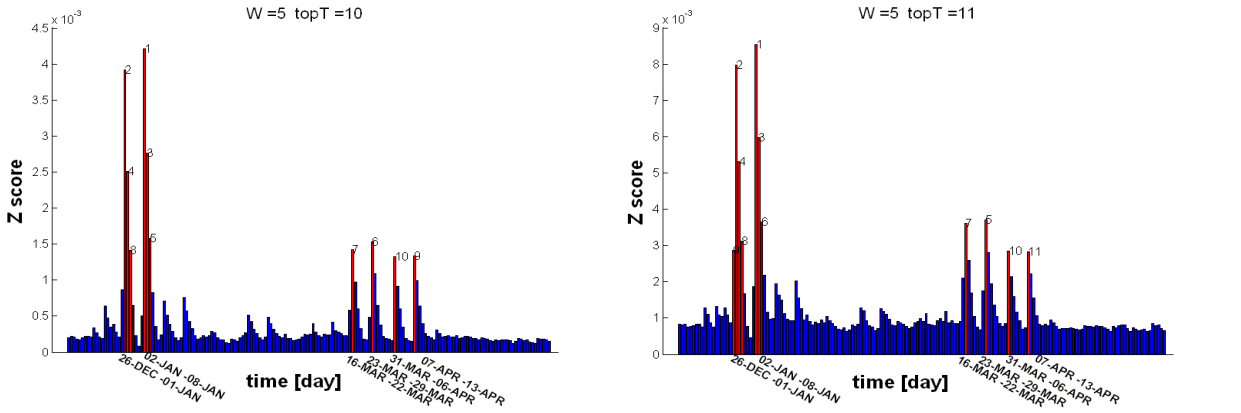

Fig. 5. Top time points flagged by our method (red bars) for features (left) F: numrecip and (right) F: out-degree. Notice that using two correlated features our method detected the same time points as with F: in-weight (Fig. 4).

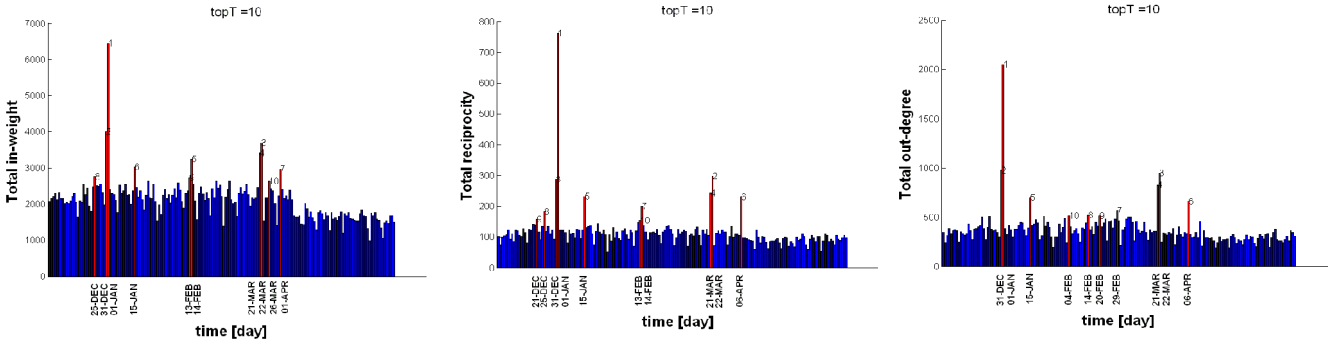

Fig. 6. Top time points flagged by using total volume (red bars) for features (left) F: in-weight, (middle) F: numrecip, and (right) F: out-degree.

4.2 Attributing Change to Nodes

Here the question we try to answer is that for a given change-point detected, can we go back and characterize which node(s) contributed to the change the most?

Figure 7 shows the scatter plot of the values of the eigen-scores **u(t)** versus the typical pattern scores **r(t-1)** for all the nodes on Dec. 26. Here, we observe that most of the values lie on the diagonal, which shows that a majority of the nodes did not change much on their typical behavior ($u_i(t) \sim r_i(t-1)$). On the other hand, some points that are far off-diagonal are marked in red that contribute to the Z score the most.

Similarly, Figure 8 shows the amount of change ratio (%) for 10K randomly picked nodes (bottom row shows the values in sorted order). Again, the same top 5 nodes as in Figure 7 are marked in red.

Since our data does not contain any labels about any type of anomalies or change-points, in Figure 9 we depict the time series of total SMSs received (in-weight) over 183 days for the top 5 nodes (that are marked in red on Figures 7 and 8) each row representing a node. Here we observe that, three of the nodes (rows 1, 4 and 5) have *no* activity on the week of Dec. 26. They are marked because they are observed to have some activity over the previous weeks. On the other hand the other two nodes (rows 2 and 3) have the opposite behavior. They start receiving SMSs *after* the Christmas week. Interestingly, we also observe that these two sets of nodes lie in different sides of the diagonal in Figure 7, indicating an opposite change in their behaviors.

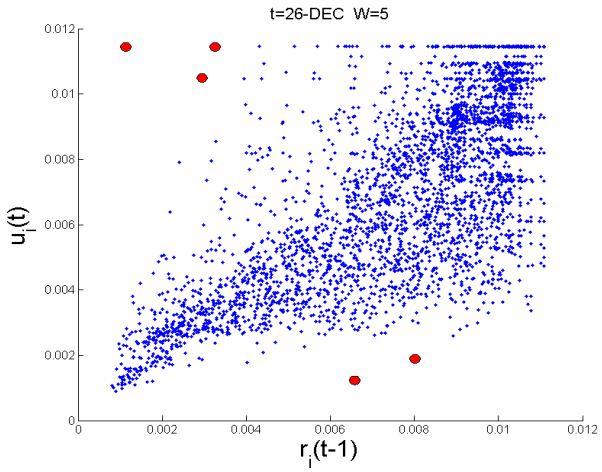

Fig. 7. Scatter plot **u(t)** versus **r(t-1)**. Each blue dot indicates a node. Nodes far away from the diagonal change in "behavior" the most (top 5 marked in red).

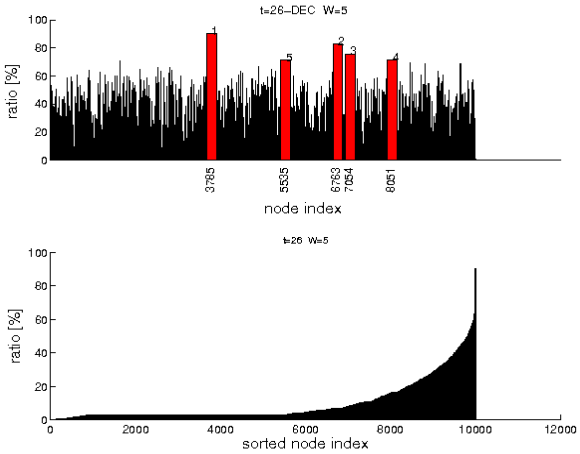

Fig. 8. (top) Change ratios (%) of 10K nodes in **u(t)** and **r(t-1)**. Each bar indicates a node (top 5 shown in red). (bottom) Ratio values sorted.

In addition, Figures 10 and 11 show corresponding results for April 7 (the change point with the third highest Z score) (as before, we use F: in-weight). Note especially the drop in activity over the week starting on April 7 after high activity on April 6 (Hindi New Year in 2008).

Finally, we also mention that similar results are obtained when we use other features such as "numrecip" and "out-degree" but we omit them here for brevity. The comparison of nodes detected by using different features is subtle: Although one can look at the overlap of nodes in the top k ranked list of result, we choose to show the time series of the top 5 nodes detected to contribute to "change" the most. Here, our goal is to show that our results make intuitive sense, that is, the behaviors of nodes indeed do change significantly for the flagged time intervals.

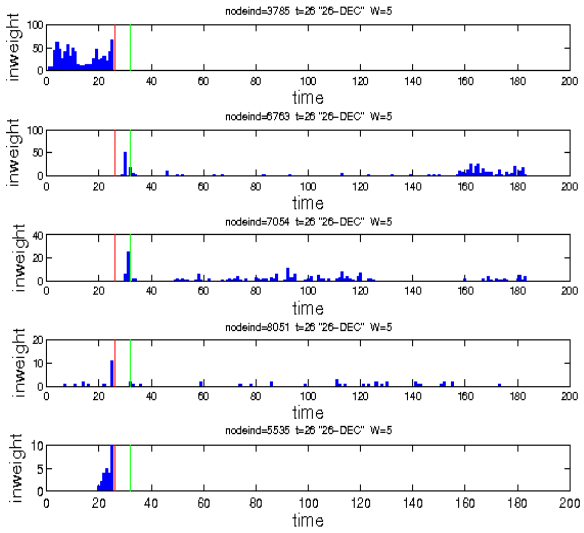

Fig. 9. Time series of inweight values of top 5 nodes marked in Figures 6 and Figure 7.

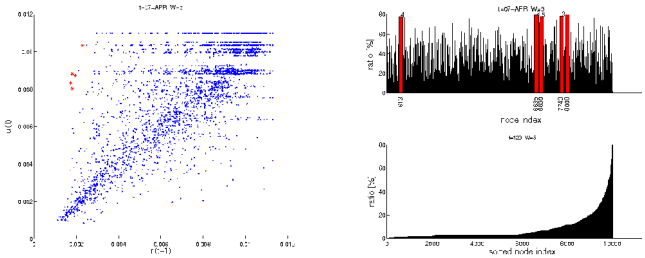

Fig. 10. (left) Scatter plot **u(t)** versus **r(t-1)** on April 7. (right) Change ratios (%) of 10K nodes in **u(t)** and **r(t-1)** on April 7th.

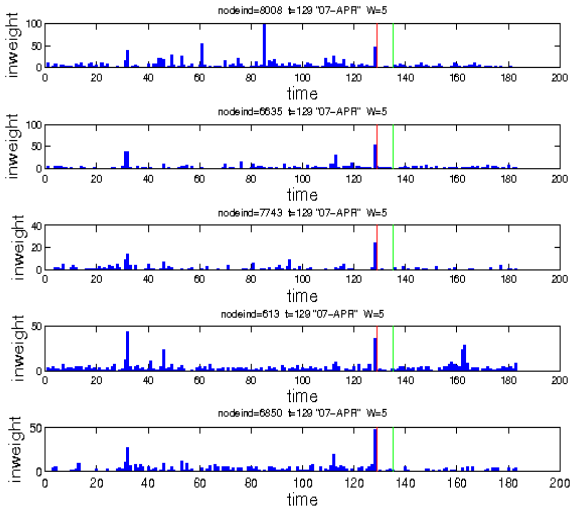

Fig. 11. Time series of in-weight values of top 5 nodes marked in Figure 9.

4.3 Detecting Change-Points per Node

Next, we switch to applying the same method on the other dimension of the data tensor I started with. In particular, here instead of looking at the TxN matrix for a particular feature F, I take the TxF matrix for a particular Node X and try to detect "interesting"/change-points for that particular node only.

Since applying the method on all 2 million nodes is not practical, we choose the top 2 nodes with the highest number of SMSs received. The first user is:

X=84332250336|MR|11-OCT-73|400099|20-FEB-03|ACTIVE|RCVALUE

where X denotes the anonymous customer ID, MR is gender, next comes the listed birth-day and the rest are some extra information that are not relevant in this study.

The second user is:

X= 84354376140|Mr|01-JAN-70|400037|07-DEC-07|ACTIVE|RCVALUE
84354376140|Mr|01-APR-82|400010|07-DEC-07|ACTIVE|RCVALUE

Here, we observe that some customers share the same ID –maybe two people sharing the same phone-line/service.

We conjecture that the birth day in our data would be informative in the sense that these days are expected to be flagged as change-points for these nodes. Unfortunately, for the first node, the birthday is in Oct. and our data spans only until May. Also, for one of the second users the listed birth day of 01-Jan-70 is the default date indicating the user indeed did not set his birthday correctly.

Figure 12 shows the top 10 change-points detected for these two users. Unfortunately, it is hard to make any argument about how valid these results are because this time they are more subjective.

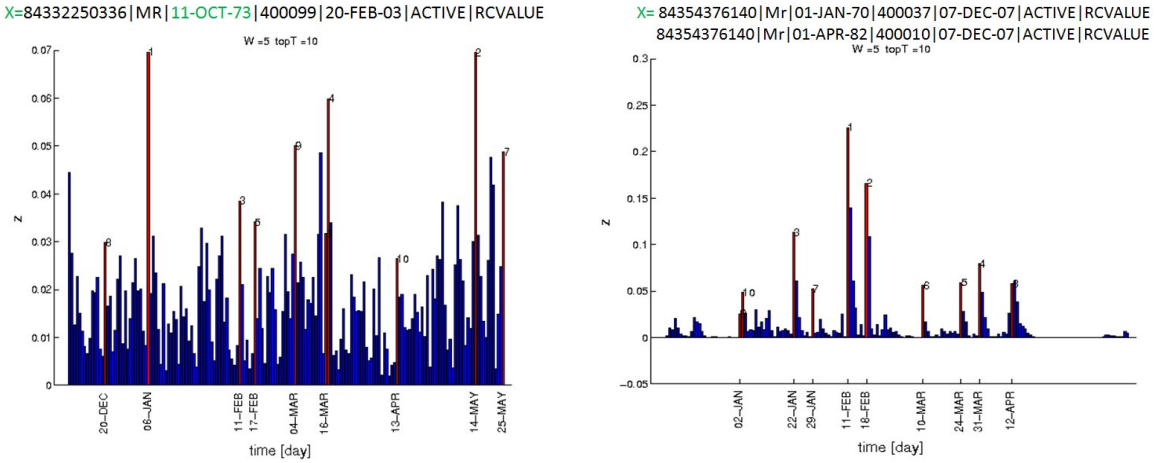

Fig. 12. Top 10 time points flagged by our method (red bars) for two users with the most number of SMSs received.

For the first node, the week of Jan 6 is found to be the most important change-point. Figure 13 shows the ratio of change for all the 12 features at that time. We observe that "in-weight", "average in-weight" and "maximum in-weight" contribute to the change the most. Although these three are correlated features, it is surprising that on the contrary, the "in-degree" itself does not change much (first bar). This suggests for receiving many more SMSs but from the same set of contacts (high "in-weight", constant "in-degree"). Also, the forth most changing feature is "numrecip". This also shows that that user is replying many more SMSs received than usual.

Figure 14 on the other hand shows the time series of all the 12 features for node X=84332250336 on Jan 6. The start and end of the week is marked with red and green vertical bars in the time line, respectively. Unfortunately, it is hard to notice any change compared to recent past for this week since the node is highly active over the whole period of 6 months.

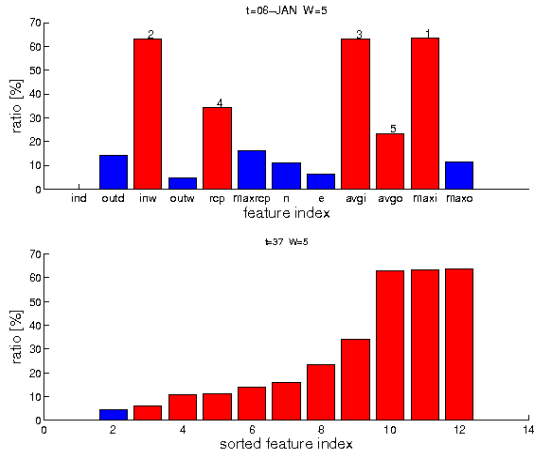

Fig. 13. (top) Change ratios (%) of 12 features in **u(t)** and **r(t-1)**. Each bar indicates a feature (top 5 shown in red). (bottom) Ratio values sorted.

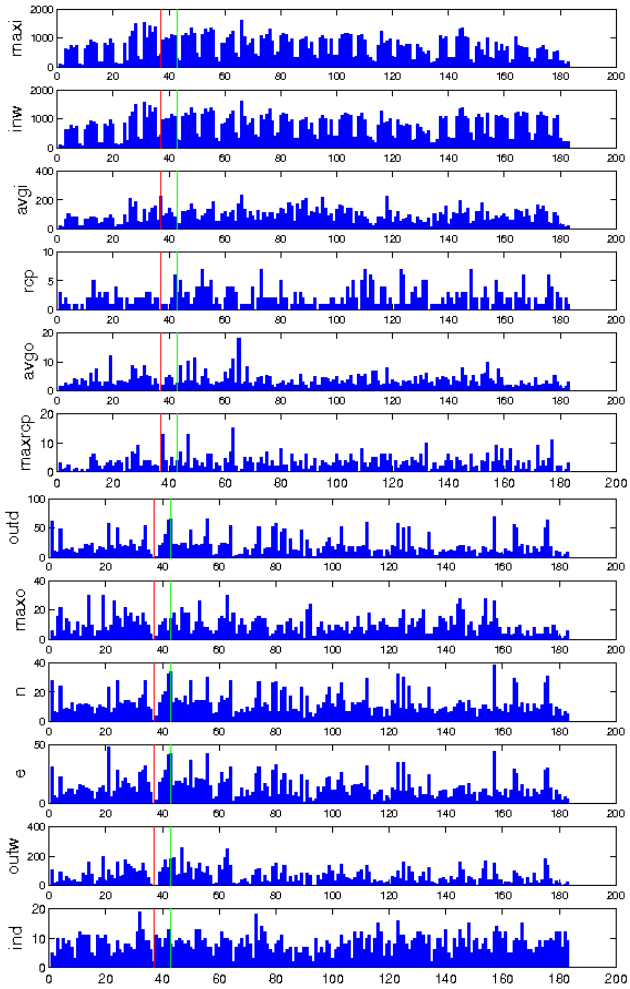

Fig. 14. Time series of all the 12 features for Node X=84332250336 on Jan 6th. Red line marks the start whereas the green line marks the end of the week.

CONCLUSIONS

We propose an algorithm based on "eigen-behavior" analysis to (1) spot "change-points" in time at which the majority of the nodes in a given network deviate from their normal behavior; and (2) point out the specific nodes that are most related to the cause of a detected change-point. We validate the effectiveness of our method on a network dataset of millions of mobile phone users and their SMS interactions over half a year. Although there exists no ground truth information in our SMS data analyzed, the experimental results suggest that our method is able to detect interesting time points such as Christmas and the Hindi New Year. On the other hand, identifying the nodes that contribute to a change the most is harder to evaluate and needs more analysis for further evaluation purposes. Moreover, given the 3-mode characteristics of our data (nodes, features, time), we were also able to find change-points for a given node by tracking the eigen-behavior based on its network features.

ACKNOWLEDGMENTS

This material is based upon work supported by the Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053, the National Science Foundation under Grant No. IIS0808661, iCAST, and an IBM Faculty Award. Any opinions, findings, and conclusions or recommendations in this material are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory, the U.S. Government, the National Science Foundation, or other funding parties. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

REFERENCES

Basseville, M. and Nikiforov, V., Detection of Abrupt Changes: Theory and Application, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1993.

Bolton, R. J. and Hand, D. J., Statistical Fraud Detection: A Review, Statistical Science, 17(3): 235-255, 2002.

Brodsky, B. and Darkhovsky, B., Nonparametric Methods in Change-Point Problems, Kluwer Publishers, 1993.

Bunke, H. and Shearer, K., A graph distance metric based on the maximal common subgraph, Pattern Recognition Letters, 19 (3/4), 1998, pp.255-259.

Chau D. H., Pandit S., Faloutsos C., Detecting fraudulent personalities in networks of online auctioneers, PKDD 2006.

Fujimaki, R., Yairi, T. and Machida, K., An Approach to Spacecraft Anomaly Detection Problem Using Kernel Feature Space, ACM SIGKDD 2005, pp.401-410.

Guralnik, V. and Srivastava, J., Event Detection from Time Series Data. ACM SIGKDD 1999, pp.33-42.

Gustafsson, F., Adaptive Filtering and Change Detection, John Wiley & Sons Inc., 2000.

Ide, T. and Kashima, H., Eigenspace-Based Anomaly Detection in Computer Systems, ACM SIGKDD 2004, pp.440-449.

Karlton Sequeira and Mohammed Javeed Zaki. Admit: anomaly-based data mining for intrusions. KDD, 2002.

Kawahara, Y., Yairi, T. and Machida, K., Change-Point Detection in Time-Series Data Based on Subspace Identification. IEEE ICDM 2007, pp.559-564.

Kifer, D., Ben-David, S and Gehrke, J., Detecting Change in Data Streams, VLDB 2004, pp.180-191.

Shoubridge P., Kraetzl M., Wallis W. D., Bunke H., Detection of Abnormal Change in a Time Series of Graphs. Journal of Interconnection Networks, 2002 Volume 3, pp.85-101.

Sun J., Faloutsos C., Papadimitriou S., Yu P. S.: GraphScope: parameter-free mining of large time-evolving graphs. KDD 2007.

Yamanishi, K. and Takeuchi, J., A Unifying Framework for Detecting Outliers and Change Points from Non-Stationary Time-Series Data, ACM SIGKDD 2002, pp.676-681.