Identifying Groups of Interest through Temporal Analysis and Event Response Monitoring

Jonathan Mugan, Eric McDermid, Abbey McGrew, and Laura Hitt
21CT, Inc.
6011 W. Courtyard Drive
Building 5, Suite 300
Austin, TX 78730
http://www.21CT.com
Email: {jmugan,emcdermid,amcgrew,lhitt}@21ct.com

***Abstract*—We present a method for finding groups of interest in large social networks. A temporal group detection algorithm identifies tightly connected groups by analyzing communications as they unfold over time. Since the number of groups found through temporal group detection may be too large to allow for manual analysis of their behavior, we also present an algorithm to identify groups of interest within an existing set of groups. This algorithm works by observing how groups react to key events and finds groups of interest by noting which groups respond anomalously. We demonstrate this approach on two social media datasets collected from Twitter. The first dataset involves tweets from Afghanistan when Afghanistan signed a letter of cooperation with India. The second dataset involves tweets surrounding the death of Steve Jobs. In both cases, our algorithm was able to identify appropriate groups.**

I. Introduction

We seek to understand and predict our world, but we often must rely on indirect evidence because we cannot observe the world directly. This indirect evidence often comes with high volume, variety, and velocity [1], and we therefore must develop frameworks to characterize these flows so that they are amenable to human and automated analysis. In some circumstances, it is helpful to describe the world in terms of actors and the relationships between them. In particular, we may want to identify groups of actors who exhibit particular patterns of interactions over time.

With the ability to identify particular groups of actors, we may be able to find groups of people who are likely to engage in violent demonstrations or form a disruptive flash mob. We may also wish to find threatening groups such as terrorists, or alternatively, we may wish to find people who could be helped by some particular health care service. Marketers may wish to pinpoint groups of people who would be interested in their product, instead of having to annoy masses of consumers who are not interested in hearing their message.

Social media can be a useful tool for finding groups of interest. It not only serves as a real-time window on the events, thoughts, and communications of distant areas [2], but also serves as an abundant, publicly available proxy for sensitive operational data. The data is live and is collected as it is happening. The results obtained from any analysis may be noisy, but they give the analyst a sense of how information processing algorithms will work on real, operational data.

The contribution of this paper is the presentation of a pair of algorithms that, when put together, are capable of finding groups with properties of interest in large datasets. We evaluated these algorithms on social media data and found that groups of interest could be identified. We first discuss how temporal group detection finds groups whose communications are relatively stable over time. We then discuss how particular groups of interest can be identified within the set of groups that were found through temporal group detection. This process works by observing how each group responds to an event and identifying groups of interest by noting which ones respond anomalously. We conclude with experimental results.

II. Related Work

Our work lies at the intersection of social network analysis [3], static community detection, and temporal community evolution. There has been enormous interest in the study of these topics over the last decade, hence, we present a brief overview with representative references of each, paying particular attention to the work that is most related to our own.

A. Static community detection

The goal of *static community detection* algorithms is to discover unusually dense subgraphs of a given graph. Tang and Liu [4] have categorized the various community detection into four different types of approaches: node-centric, group-centric, network-centric, and heirarchy-centric. For our purposes, it suffices to review the first three of these. In *node-centric* community detection, a set of nodes is a community if every node satisfies a given set of properties (see, e.g., [5, Chapter 7] for an overview). Many of these problems can be solved through combinatorial techniques, although often at very high cost, due to their computational complexity.

Group-centric community detection considers a set of nodes to be a community if the set as a whole satisfies a given property. Examples include density-based and quasi-clique communities [6]. Under this community definition, an individual node may be sparsely connected with the rest of the community, but is included because the community collectively satisfies the given criteria.

The majority of research has been done in the area of *network-centric* community detection. The state-of-the-art approaches are based on latent space models, in which the nodes (edges) of a network are mapped into a low-dimensional Euclidean distance space (using, e.g., matrix factorization) such that the proximity between nodes (edges) based on network connectivity is preserved. The nodes (edges) can then be clustered in the low-dimensional space using straightforward clustering schemes such as k-means. Representative approaches include [7], [8], [9]. Borg and Groenen [10] also provide a broad overview of these techniques. More recent work has extended the latent-space models to consider not only the structure of the underlying graph, but also additional information such as edge and node content (see e.g., [11], [12]).

B. Temporal community evolution

While static community detection aims to find meaningful dense structures in static graphs, *temporal community evolution* (also called dynamic community evolution) additionally aims to discover the emergence of a community, and quantify its growth, decay, or combination with other temporal communities. These insights are lost – or at best misrepresented – when, instead, the nodes and edges of the social network are aggregated into a static graph.

A common framework for temporal community detection and evolution is given by Asur et al [13]. First, a series of "snapshots" g^1, ..., g^k of the input graph are computed. Next, a static community detection algorithm is applied, resulting in a set of communities c_1^i, ..., c_z^i for each snapshot g^i. The appropriate communities are then linked across each g^i. Such a framework has been used, for example, to study a community over time to detect different events or observe the reaction of a group to a known event [14], [15], [16], [13].

A different but related topic to temporal community evolution is that of graph anomaly detection. These approaches typically either consider the evolution of a single graph over time or seek to identify anomalous subgraphs with in a very large graph. We refer the reader to the tutorial of Akoglu and Faloutsos [17] for a thorough overview of such techniques.

III. TEMPORAL GROUP DETECTION

In this section, we present the Temporal Group Detection algorithm.[1] Temporal Group Detection takes a directed graph with time stamps associated with the edges and returns the set of densely connected groups of nodes that persist through a sufficiently significant period of time. We begin this section with a few definitions and notation that we require in order to describe the algorithm; however, for space reasons and ease of presentation, we are occasionally informal in our description.

A *directed temporal graph* $D = (V, E, T)$ is a directed graph with a node set V (a set of people) and an edge set E (a set of communications between them). Associated with each edge $e \in E$ is a time label $t(e)$; the set T is the union of these time labels. Given a directed temporal graph, a temporal series of directed graphs $D_S = (D_0, D_1, \ldots, D_z)$ naturally arises by partitioning the time stamps of the edges of D into time intervals. A *temporal group* is, informally, a subset of vertices $V' \subseteq V$ such that V' is a sufficiently densely connected cluster throughout a sufficient subsequence of a temporal series of directed graphs D_S.

The *Temporal Group Detection* algorithm (TGD) is presented in Algorithm 1. It takes parameters D and S, where $D = (V, E, T)$ is a directed temporal graph and $S = \{(s_0, t_0), (s_1, t_1), \ldots, (s_z, t_z)\}$ is a set of time intervals. The goal of the TGD is to find the temporal groups in the temporal series of directed graphs implied by D and S. The process consists of the three steps described below.

A. Step 1: Detect Static Groups

Given D and $S = \{(s_0, t_0), (s_1, t_1), \ldots, (s_z, t_z)\}$, Step 1 of the TGD partitions E into z subsets $E_1, E_2, \ldots, E_z$, where E_i is the set of edges e_j with $s_i \leq t(e_j) \leq t_i$. The series of graphs D_S is computed by setting the i^{th} graph in the sequence to be the graph induced by the edges E_i. Next, a group detection algorithm is used to find the densely connected subgraphs within each $D_i \in D_S$. A number of possible algorithms are available in the literature (see e.g., [19]); we used the "Best Friends" algorithm of Moy [20]. The output of this step is the union Q of all groups detected over each D_i.

Algorithm 1 Temporal Group Detection

```
Require: a directed temporal graph $D = (V, E, T)$.
Require: a set of time intervals $S = \{(s_0, t_0), (s_1, t_1), \ldots, (s_z, t_z)\}$.
 1: *** Step 1: Detect Static Groups ***
 2: compute the temporal series of directed graphs $D_S$
 3: for each $D_i \in D_S$ do
 4:     compute the set $Q_i$ of (static) groups in $D_i$.
 5: end for
 6: *** Step 2: Match Groups Across Time Intervals ***
 7: initialize an auxiliary graph $H$ to contain one node for each group in $Q$
 8: for each pair of nodes $(q_i, q_j)$ where $q_i$ appears in an earlier time interval that $q_j$ do
 9:     if sufficient overlap between members of $q_i$ and $q_j$ then
10:         make $(q_i, q_j)$ a directed edge of $H$
11:     end if
12: end for
13: *** Step 3: Merge and Refine Matched Groups ***
14: compute the partition of paths $p_1, \ldots, p_k$ of $H$
15: merge the set of groups along each $p_i$, resulting in a new set of groups $\mathcal{G}$
16: discard the unstable groups from $\mathcal{G}$
17: while some pair of groups $(q_i, q_j)$ in $\mathcal{G}$ is sufficiently similar do
18:     merge $q_i$ and $q_j$ into one group $q_k$
19:     remove $q_i$ and $q_j$ from $\mathcal{G}$ and insert $q_k$ into $\mathcal{G}$
20: end while
21: return $\mathcal{G}$
```

B. Step 2: Match Groups Across Time Intervals

Having obtained a set of groups Q, Step 2 of the TGD attempts to identify the evolution of a group over D_S. To accomplish this, a directed acyclic auxiliary graph $H = (Q, A)$ is computed, where Q is the set of groups discovered in Step 1, and A is the set of all directed edges (q_i, q_j) where group q_i appears in an earlier time interval than q_j, and the

[1]This algorithm, implemented as DSP 1.12, has not been previously presented in the scientific literature. We reference the user manual [18].

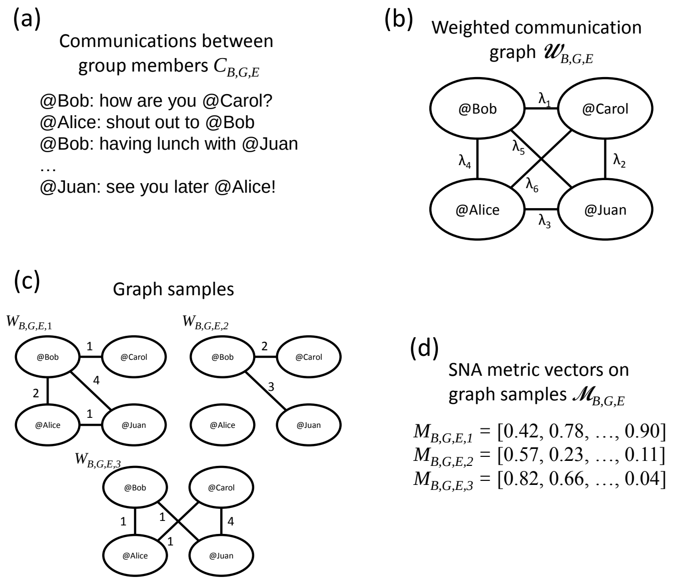
(a)
Communications between group members $C_{B,G,E}$
@Bob: how are you @Carol?
@Alice: shout out to @Bob
@Bob: having lunch with @Juan
…
@Juan: see you later @Alice!
(b)
Weighted communication graph $\mathcal{W}_{B,G,E}$
@Bob
@Carol
@Alice
@Juan
λ_1
λ_2
λ_3
λ_4
λ_5
λ_6
(c)
Graph samples
$W_{B,G,E,1}$
$W_{B,G,E,2}$
$W_{B,G,E,3}$
(d)
SNA metric vectors on graph samples $\mathcal{M}_{B,G,E}$
$M_{B,G,E,1}$ = [0.42, 0.78, …, 0.90]
$M_{B,G,E,2}$ = [0.57, 0.23, …, 0.11]
$M_{B,G,E,3}$ = [0.82, 0.66, …, 0.04]

Fig. 1. (a) A set of communications before event E between a group of four Twitter users: @Bob, @Carol, @Alice, and @Juan. (b) The weighted communication graph $\mathcal{W}_{B,G,E}$, where λ_i represents the strength of communication ties between two Twitter users. (Shown as undirected for ease of presentation.) (c) Sample graphs taken from the weighted communication graph with edge i appearing according to the Poisson distribution with parameter λ_i. (Shown as undirected for ease of presentation.) (d) A set of vectors $\mathcal{M}_{B,G,E}$ of SNA metric values computed for each sample graph.

node set of q_i and q_j is sufficiently similar according to user-specified parameters. Intuitively, the paths of this graph trace the evolution of underlying stable groups through sequences of observed groups in D_S; a group is stable if there is sufficient overlap of members between time slices. The output of this step is the auxiliary graph H.

C. Step 3: Merge and Refine Matched Groups

The final step of the TDG is to produce a final set of groups $\mathcal{G}$ that, intuitively, are considered to have been sufficiently present throughout the period of time of the analysis. In particular, the algorithm partitions H into disjoint paths, $p_1, \ldots, p_k$. The set of groups along each path p_i are merged into a single group, resulting in a new set of groups $\mathcal{G}$. Each group in $\mathcal{G}$ is then considered in turn, and the groups that are not considered to be "stable" enough are discarded. Next, pairs of groups are continually merged until no two groups are sufficiently similar. The set of groups $\mathcal{G}$ that remain at the end of this process are returned by the TDG algorithm.

IV. Monitoring Responses of Groups to Events

Within the set of groups found through Temporal Group Detection, we want to find particular groups of interest by identifying which groups respond to certain classes of events. For each group, we have a set of communications between the members. We convert those communications into a representation of group behavior that we then use to measure if that behavior changes in response to an event. We present two different methods for measuring how anomalous a behavior change is in response to an event. The first method is Behavior Plotting. As the name suggests, Behavior Plotting plots the

Algorithm 2 Compute Group Behavior Representation

```
Require: A set of groups $\mathcal{G}$ found through Temporal Group Detection (Section III).
Require: A set of communications $\mathcal{C}$ between the members of the groups $G \in \mathcal{G}$.
Require: An event $E$
 1: for each $G \in \mathcal{G}$ do
 2:     Let $C_{B,G,E} \subseteq \mathcal{C}$ be the set of communications for group $G$ that occur before event $E$, and let $C_{A,G,E} \subseteq \mathcal{C}$ be the set of communications for group $G$ that occur after event $E$.
 3:     Create weighted communication graph $\mathcal{W}_{B,G,E}$ to have the nodes of group $G$ and edges between each pair of nodes. Give each edge a weight $\lambda$ reflecting the frequency of communications in $C_{B,G,E}$ as described in Section IV-A1. Create weighted communication graph $G_{A,G,E}$ analogously.
 4: end for
 5: for $i$ in range 100 do
 6:     Let $W_{B,G,E,i}$ be a sample of $\mathcal{W}_{B,G,E}$ acquired by sampling the edges of $\mathcal{W}_{B,G,E}$ using the Poisson distribution with each edge weight $\lambda$ as a parameter.
 7:     Compute a vector $M_{B,G,E,i}$ of social network analysis metrics over $W_{B,G,E,i}$ as described in Section IV-A2.
 8:     Add $M_{B,G,E,i}$ to the set $\mathcal{M}_{B,G,E}$
 9:     Compute $M_{A,G,E,i}$ analogously and add it to the set $\mathcal{M}_{A,G,E}$
10: end for
11: return $\mathcal{M}_{B,G,E}$ and $\mathcal{M}_{A,G,E}$
```

change in behavior of a group in response to an event. Behavior Plotting provides visual information to the analyst, but it is somewhat subjective in its interpretation. The second method for measuring how anomalous the behavior of a group is in response to an event is called Anomaly Ranking. As the name again implies, Anomaly Ranking ranks the groups according to degree of anomalous behavior. Anomaly Ranking lacks the visual information of Behavior Plotting, but it provides objective results.

A. Representing Group Behavior

We represent group behavior by creating a model of the group dynamics. This model is in the form of a graph called a weighted communication graph (Section IV-A1). We then sample graphs from the weighted communication graph and compute SNA metrics on those sampled graphs to get a vector of SNA metrics (Section IV-A2) for each sample.

To compute the behavior change of group G to event E, we compute the set of SNA metric vectors $\mathcal{M}_{B,G,E}$ for the communications before event E, and we compute the set of SNA metric vectors $\mathcal{M}_{A,G,E}$ after event E. An example for the process for computing the group behavior for a group G before an event E is shown in Figure 1 and the algorithm is shown in Algorithm 2, and further additional details are provided in Section IV-A1 and Section IV-A2.

1) Computing the Weighted Communication Graph: The weighted communication graph for a group is a directed graph that represents how often the members of the group communicate with each other each week. To compute the weighted communication graph, we assign weights to the edges between the pairs of nodes in group $G \in \mathcal{G}$ equal to the average frequency of communications between those nodes as they appear in the time interval duration (e.g. Bob talks to Carol 4.2 times per week). For each edge, the weight is used as the λ parameter of a Poisson random variable. These λ parameters are then used to create sample communication graphs from this weighted communication graph. These samples are created so that the frequency of communications has proportional influence in the SNA metric space.

2) Computing Social Network Analysis Metrics: The behavior of a group is characterized using a vector consisting of the following twelve SNA metrics. These metrics are computed over the largest connected component of the graph. If a metric requires an undirected graph, the undirected graph is computed from the directed graph by summing the edge values.

Diameter: the maximum distance between any two nodes in the undirected version of the graph.

Radius: the minimum eccentricity of all nodes in the undirected version of the graph. The eccentricity of a node is the maximum distance from that node to all other nodes in the graph.

Characteristic path length: the average of the lengths of all shortest paths between any two pairs of nodes (excluding self-pairs) in the undirected version of the graph.

Algebraic connectivity: the second smallest eigenvalue of the Laplacian matrix for the undirected version of the graph. The Laplacian matrix of a graph is the diagonal matrix of the node degrees minus the adjacency matrix.

Average betweenness: the average of the betweenness centrality values for each node in the graph. Betweenness centrality is defined as the fraction of the shortest paths between all node pairs that pass through that node, in the undirected version of the graph.

Average closeness: the average of the closeness centrality values for each node in the graph. Closeness centrality for a node is defined as the inverse of the average shortest distance from that node to all other nodes, in the undirected version of the graph.

Weighted average closeness: the average of the closeness centrality values for each node in the weighted version of the graph. Closeness centrality for a node is defined as the inverse of the average shortest distance from that node to all other nodes, in the undirected version of the graph.

Average eigenvector centrality: the average eigenvector value for the largest eigenvalue in the undirected, unweighted graph adjacency matrix.

Weighted average eigenvector centrality: the average eigenvector value for the largest eigenvalue in the undirected, weighted graph adjacency matrix.

Average undirected volume: the average sum of the edge weights for all undirected edges associated with each node.

Average out percent degree: the average of the fraction of directed edges in the graph that go through each vertex that are an 'outgoing' edge.

Percent active members: the percentage of members with at least one edge.

B. *Finding Groups of Interest through Behavior Plotting*

To identify groups of interest, we determine which groups have anomalous changes in their behavior vector [21]. These anomalies can be seen visually by plotting their behavior change, where for any group, the response to the event is represented by a line from the behavior plotted before the event to the behavior plotted after the event. Since plots are easiest to visualize in two-dimensional space, we use Principal Components Analysis (PCA) [22] to perform dimensionality reduction on the behavior vectors, and we plot the behaviors using the first two principal components. This process is shown in Algorithm 3.

Algorithm 3 Behavior Plotting

```
Require: A set of groups $\mathcal{G}$ found through Temporal Group
    Detection (Section III).
Require: An event $E$
Require: Sets of vectors $\mathcal{M}_{B,G,E}$ and $\mathcal{M}_{A,G,E}$ representing
    the behavior for each group $G$ both before and after,
    respectively, for event $E$.
1: Perform PCA over the set $\bigcup_{G \in \mathcal{G}} \mathcal{M}_{B,G,E} \cup \mathcal{M}_{A,G,E}$
2: Plot the centroids of $\mathcal{M}_{B,G,E}$ for each $G \in \mathcal{G}$ in the
   space created by the top two eigenvectors from the PCA
   calculation. These points will be the tails (beginnings) of
   the arrows.
3: Plot the centroids of $\mathcal{M}_{A,G,E}$ for each $G \in \mathcal{G}$ in the
   space created by the top two eigenvectors from the PCA
   calculation. These will be the heads (endpoints) of the
   arrows.
```

C. *Finding Groups of Interest through Anomaly Ranking*

We also identify groups of interest by anomaly ranking [21]. A group signature is computed for each group by subtracting the average of the groups' SNA metric samples before the event from the average of the groups' SNA metric samples after the event. Then a score is computed as the highest magnitude of the signature plotted in the PCA space of those remaining, divided by the eigenvector.

Anomalies are found by comparing each group's signature score with all of the others. Once the most anomalous network is found, it is removed from consideration, and the algorithm looks for the next most anomalous network. This process is shown in Algorithm 4.

V. EXPERIMENTAL PROCEDURE AND RESULTS

Our hypothesis is that if a group has a property x of interest, then it will respond to an event with a related property x' differently than those groups who do not share property x. Evaluating hypotheses in Big Data domains such as social media can be difficult because it can be impractical to manually create ground-truth labels. Instead, one can consider alternative sources of information and evaluate if these disparate sources lead to the same conclusion. We use this approach and present the results of two exploratory experiments.

Our data set consists of communications within Twitter data. Twitter is an online social network where users can post messages of 140 characters for the world to see. We focus on Twitter data because of the public nature of the

Algorithm 4 Anomaly Ranking

```
Require: A set of groups $\mathcal{G}$ found through Temporal Group Detection (Section III).
Require: An event $E$
Require: Sets of vectors $\mathcal{M}_{B,G,E}$ and $\mathcal{M}_{A,G,E}$ representing the behavior for each group $G$ both before and after, respectively, for event $E$.
 1: let $S_E = \{\}$
 2: for each $G \in \mathcal{G}$ do
 3:     Let $S_{G,E} = avg(\mathcal{M}_{A,G,E}) - avg(\mathcal{M}_{A,G,E})$
 4:     Add $S_{G,E}$ to $S_E$ to
 5: end for
 6: for $|G| - 3$ times do
 7:     Compute PCA on $S_E$
 8:     Select $S_{G,E}$ with highest $score(S_{G,E}, S_E)$ where $score(S_{G,E}, S_E)$ is the magnitude of the projection of $S_{G,E}$ in the PCA space of $S_E$, divided by the eigenvalue of $S_{G,E}$.
 9:     Add $G$ to anomalous queue
10:     remove $S_{G,E}$ from $S_E$
11: end for
12: return anomalous queue
```

communications and because of the size of the Twitter social network. We say that a communication between two people has occurred when one person mentions another in a tweet. In Twitter, user names begin with the '@' symbol, so it is easy to determine who is being mentioned. There can also be multiple communications in a single tweet. E.g., if @Alice tweets: "I can't wait to hang out with @Bob and @Carol at the junk yard tomorrow!" We say that one communication has taken place between @Alice and @Bob and another communication has taken place between @Alice and @Carol. This is true regardless of the semantic content of the tweet.

A. *Experiment 1: A Strategic Partnership Signed between Afghanistan and India*

Within an atmosphere of increasing tensions between Afghanistan and Pakistan, Afghanistan signed a strategic partnership with Pakistan's historic enemy India, on October 4, 2011. We used this as the event in our analysis of network responses. To obtain tweets before and after the event on October 4, we steadily collected tweets from Kabul, Afghanistan between September 29, 2011 and October 7, 2011. This collection effort resulted in 7,523 tweets containing 1,913 users and 5,398 communications between those users.

Temporal Group Detection found seven groups, two of which were identified as groups of interest by Behavior Plotting (Algorithm 3). Figure 2 shows how the SNA metrics for the seven groups changed from before the event (base of the arrow) to after the event (head of the arrow). The x and y axes represent the values along the top two principal components of a vector-based SNA representation after PCA was applied. The slope of the vectors indicates that groups 7 and, to a lesser degree, 6 were anomalous. This pattern of anomalous behavior corresponds to the external information shown in Figure 3. Those two groups in the plot in Figure 2 were the same groups who were the least interested in India as measured in the bar graph in Figure 3 by their percentage of tweets about India.

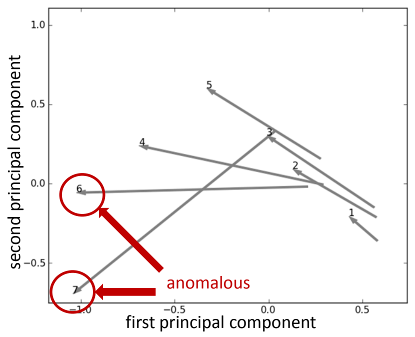

Fig. 2. The response of the seven groups to the event of the signing of the strategic partnership between Afghanistan and India. Groups 6 and 7 appear to respond differently than the others. The principal components are linear combinations of the twelve SNA metrics, as found by the PCA dimensionality reduction algorithm [22].

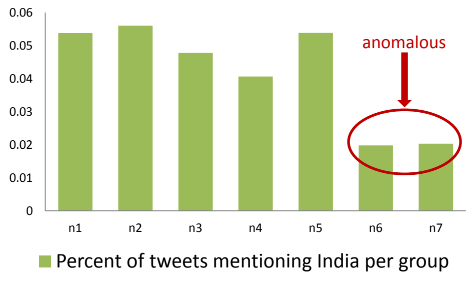

Fig. 3. The frequency of tweeting about India among the seven groups. The groups that reacted differently in Behavior Plotting, groups 6 and 7, mentioned India less often than the others did.

The subjective results shown by Behavior Plotting (Algorithm 3) shown in Figure 2 correspond to the objective ranked results from Anomaly Ranking (Algorithm 4). Anomaly Ranking declared that the most anomalous group was 7, followed by 6, 4, and 1. The remaining three groups were unranked as per the algorithm description.

B. *Experiment 2: The Death of Steve Jobs*

We also examined data collected around the time of the death of Steve Jobs on October 5, 2011. Tweets were collected in two stages. In the first stage, we collected tweets for Temporal Group Detection from each member of a social network of size 784 that was centered on an individual in the United States. We collected tweets from September 25, 2011 until October 7, 2011. The tweets resulted in 460 users and 3,318 communications between users. In the second stage, to determine the proclivity of each user to tweet about Steve Jobs, we collected tweets from each user in each group going backward from October 24, 2011. For each user, we collected tweets (in batches of 20) until Twitter returned 0 tweets, or the last tweet was before September 15, 2011.

Temporal Group Detection again found seven groups, of which group 7 was identified as a group of interest by Behavior Plotting (Algorithm 3), as shown in Figure 4.[2] To determine

[2]It is a moderate coincidence that both experiments produced seven groups with the last group being anomalous. The parameters of Temporal Group Detection can be set to find as few as a handful of groups to as many as dozens of groups, depending on the needs of the user.

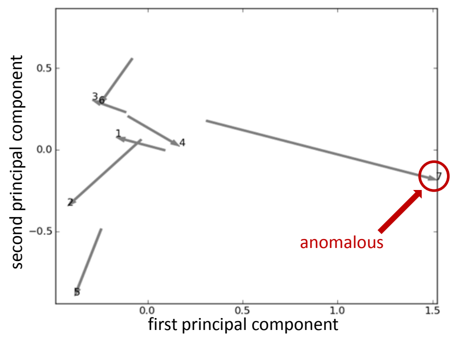

Fig. 4. The response of the seven groups to the event of the death of Steve Jobs. Group 7 appears to respond different than the other groups.

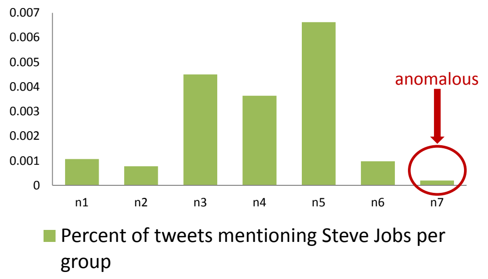

Fig. 5. The frequency of tweeting about Steve Jobs. The group that reacted differently, group 7, mentioned Steve Jobs less than the others did.

if group 7 really was anomalous with respect to the event of the death of Steve Jobs, we counted how often members of all the groups tweeted about Steve Jobs. We see in Figure 5 that group 7 did indeed tweet less about Mr. Jobs. The subjective results from Behavior Plotting (Algorithm 3) shown in Figure 4 again correspond to the objective ranked results from Anomaly Ranking (Algorithm 4). Anomaly Ranking declared that the most anomalous group was 7, followed by 4, 5, and 6. The remaining three groups were again unranked as per the algorithm description.

We also looked at Klout, an online service with a publicly available API that ranks the importance of users in social media. For the groups from Experiment 1, Klout scores did not appear to correlate with how anomalous a group was. However, with the groups arising in experiment 2, the anomalous group 7 had the lowest average Klout score.

VI. Future Work and Conclusion

We presented a pair of algorithms that allow one to find groups of interest in large datasets of communications. We found, in two experiments with social media data, that groups that were different with respect to a key event can be identified within a larger set of groups by analyzing the structural changes of the group in response to the event.

Future work will focus on making these statements more precise by assigning properties to both groups and events and using correspondences between properties to make predictions about group responses to those events. In particular, we will define the concepts of "responding" to an event and what constitutes a "group of interest" with more rigor. One approach is to only consider the magnitude of a response. If each group can be assigned a set of properties based on external information about the members of the group, and each event can also be assigned a set of properties, and if we can define a correspondence between group properties and event properties, then our hypothesis is that the magnitude of response to an event will be greater for those groups that have a property that corresponds to a property of the event. Such a framework of comparing properties between groups and events should allow the algorithms in this paper to make more precise predictions.

Acknowledgments.: We thank the Office of Naval Research for funding this work.

References

[1] D. Laney, "3D data management: Controlling data volume, velocity and variety," *Application delivery strategies*, vol. 949, 2001.

[2] M. Abbasi, S. Chai, H. Liu, and K. Sagoo, "Real-world behavior analysis through a social media lens," *Social Computing, Behavioral-Cultural Modeling and Prediction*, pp. 18–26, 2012.

[3] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

[4] L. Tang and L. H., *Community Detection and Mining in Social Media*. Morgan and Claypool Publishers, 2010.

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

[6] J. Abello, M. Resende, and S. Sudarsky, "Massive quasi-clique detection." in *LATIN, 2002*. Springer-Verlag, 2002, pp. 598–612.

[7] P. Hoff, A. Raftery, and M. Handcock, "Latent-space approaches to social network analysis," *Journal of the American Statistical Association*, vol. 97, pp. 1090 – 1098, 2002.

[8] M. Handcock, A. Raftery, and J. Tantrum, "Model-based clustering for social networks," *Journal of the Royal Statistical Society Series A*, vol. 127, pp. 301 – 354, 2007.

[9] P. Sarkar and A. Moore, "Dynamic social network analysis using latent space models," vol. 7, pp. 31 – 40, 2005.

[10] I. Borg and P. Groenen, *Modern multidimensional scaling: theory and applications*. Springer, 2005.

[11] G. Qi, C. Aggarwal, and T. Huang, "Community detection with edge content in social media networks," in *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*. IEEE, 2012, pp. 534–545.

[12] S. Zhu, K. Yu, Y. Chi, and Y. Gong, "Combining content and link for classification using matrix factorization," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 487–494.

[13] S. Asur, S. Parthasarathy, and D. Ucar, "An event-based framework for characterizing the evolutionary behavior of interaction graphs," in *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, 2007, pp. 913 – 921.

[14] J. Hopcroft, O. Khan, B. Kulis, and B. Selman, "Tracking evolving communities in large linked networks," pp. 5249 – 5253, 2004.

[15] R. Kumar, J. Novak, and A. Tomkins, "On the bursty evolution of blogspace," *World Wide Web*, vol. 8, pp. 159 – 178, 2005.

[16] G. Palla, A. Barabasi, and T. Vicsek, "Quantifying social group evolution," *Nature*, vol. 446, pp. 664–667, 2007.

[17] F. Akoglu and C. Faloutsos, "Acm wsdm 2013 tutorial: Anomaly, event, and fraud detection in large graph datasets," http://www.cs.stonybrook.edu/ leman/wsdm13/.

[18] *DSP: DAGGAR System Prototype V 1.12 User Guide*, 21CT, 2010.

[19] S. Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, Aug. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.cosrev.2007.05.001

[20] M. Moy, "Computer automated group detection," U.S. Patent 7,480,712, 2009.

[21] L. Hitt and M. McClain, "Method and apparatus for identifying a threatening network," U.S. Patent 13/730,191 (utility application), 2012.

[22] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. Wiley, 2001.