

# Algebraic Visual Analysis: The Catalano Phone Call Data Set Case Study

Anna A. Shaverdian  
Department of EECS  
University of Michigan  
annaas@umich.edu

Hao Zhou  
Department of Statistics  
University of Michigan  
zhouhao@umich.edu

George Michailidis  
Department of Statistics  
University of Michigan  
gmichail@umich.edu

H. V. Jagadish  
Department of EECS  
University of Michigan  
jag@umich.edu

## ABSTRACT

While many clever techniques have been proposed for visual analysis, most of these are “one of” and it is not easy to see how to combine multiple techniques. We propose an algebraic model capable of representing a large class of visual analysis operations on graph data. We demonstrate the value of this model by showing how it can simulate the analyses performed by several groups on the Catalano family cell phone call record data set as part of the VAST 2008 challenge.

## 1. INTRODUCTION

As visual analytics has gained importance as a field, there have been many impressive systems constructed and many clever techniques invented to support visual analysis of large data sets. From an application perspective, the ultimate measure of any technique or system has to be how effective it is in the context for which it is designed – does it support the derivation of the desired analytical results. While such a holistic measure may be the ultimate objective, from an engineering perspective, it is useful to break this down into pieces. Perhaps there are aspects of multiple systems that are each superior in their own way – how can we maximize learning from other systems and integration of novel techniques from multiple projects.

To enable this sort of integration, we propose an algebra for visual analysis, with a small number of fundamental operations. The design of specific systems can then be viewed as supporting specific expressions in this algebra. We can mix and match ideas from multiple projects by manipulating these algebraic expressions. Furthermore, we can devise new analysis path by making (often small) changes to these algebraic expressions that are harder to devise at the system level without the algebraic abstraction.

The set of operations in the algebra depends very much on the type of data to be analyzed. We restrict our attention to data that is naturally represented as a graph, with attributes on nodes and on

edges. We describe our data model in Sec. 2.

Given a very large graph, the primary impediment to its visual analysis is size. There are two major ways in which size can be reduced, selection (retain only nodes/edges that satisfy a specified predicate) and aggregation (merge nodes/edges that are in some equivalence class). In Sec. 3, we develop an algebra that formally specifies these operators, and a few additional required “house-keeping” operators.

In Sec. 4, we demonstrate the value of this algebra by showing how it can represent the analyses performed by several researchers on one of the problems in the VAST 2008 challenge [4, 14]. Additional analyses enabled in our algebraic framework are illustrated in Sec. 5.

## 2. MODEL

We start by defining the appropriate data structure: let  $\{\mathcal{D}(t) = [G(t), X(t)]\}_{t \in \mathcal{T}}$  denote a collection of graphs and their attributes, indexed by a finite set  $\mathcal{T}$ ; in the case of the motivating application the index set corresponds to time and  $\mathcal{T} = \{1, 2, \dots, 10\}$ . Further,  $G(t) = (V(t), E(t), A(t))$  is the observed graph at time  $t$ , with node set  $V(t)$ , edge set  $E(t)$  and weighted adjacency matrix  $A(t)$ . The associated attribute structure  $X(t)$  is comprised of three components:  $X(t) = [X_V(t), X_E(t), X_G(t)]$ , where  $X_V(t)$  contains node attributes (e.g. in- and out-degree in the motivating application),  $X_E(t)$  edge attributes (e.g. edge betweenness) and  $X_G(t)$  graph attributes (e.g. diameter). It should be noted that node attributes can be either intrinsic or computed. For example, geographic location or educational level of the nodes correspond to intrinsic attributes, whereas degree or clustering coefficient are computed ones. The same applies to edge and graph attributes.

In order to obtain computed attributes, it is assumed that there exists a collection of functions  $\mathcal{F} = \{\mathcal{F}_X, \mathcal{F}_G\}$ , relevant to the visual analytics problem at hand. Functions in class  $\mathcal{F}_X$  are used for computing quantities of interest from the intrinsic attributes. Such functions are defined as follows:  $f \in \mathcal{F}_X : \mathbb{R}^{|V|} \rightarrow \mathbb{R}^q$ , with  $1 \leq q \leq |V|$ , where  $|\cdot|$  denotes the cardinality of the underlying set. Examples of such functions include sorting an attribute, where  $q = |V|$ , calculating the max or the min of the attribute  $q = 1$ , quantizing (binning) the attribute in which  $q$  corresponds to the number of prespecified bins, etc.

## 3. ALGEBRA

We start by introducing the main operators in the algebra, which include aggregation  $\mathcal{A}$  and selection  $\mathcal{S}$ . All operators in this algebra

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

VAKD'09, June 28, 2009, Paris, France.

Copyright 2009 ACM 978-1-60558-670-0 ...\$5.00.

take a set of graphs as input and produce a set of graphs as output. Therefore, the algebra is closed with respect to the operators we define, and compositions of primitive operators can be used to construct compound operators (or expressions) in this algebra. In our analysis of cases studies (see Section 4) we argue that *most* visual analytics tasks can be captured by expressions in this algebra.

*Set Selection:* Given a collection of graphs and their attributes,  $\mathcal{D}$ , a set selection applied to it, based on a predicate  $\alpha$ , is written as  $\sigma_\alpha(\mathcal{D}) = \{G \in \mathcal{D} | \alpha(G) = TRUE\}$ . Observe that the selection predicate is evaluated independently for each element graph of the set, and the entire graph is either retained or discarded depending on the truth value of this predicate.

*Element Selection:* This is a basic filter based on a selection predicate that is applied to individual components of each element graph in the given collection of graphs. The cardinality of the collection remains unchanged, but each element in the collection is potentially reduced to a smaller graph. Recall that a graph may have a set of attributes  $X = [X_V, X_E, X_G]$ . An element selection  $\mathcal{S}$  takes as argument a predicate on either  $X_V$  or  $X_E$ , and accordingly selects either nodes (and incident edges) or edges, respectively, in each graph, if it satisfies the specific predicate. Notice that the predicate is evaluated on the entire data structure  $\mathcal{D}$ . Formally, we have  $\mathcal{D}' = \mathcal{S}(\mathcal{D} | X_i = \tau)$ , where  $\mathcal{D}$  denotes the input data structure on which the selection operator is applied to,  $X_i = \tau$  the generic predicate and the value that it is evaluated at, and finally  $\mathcal{D}'$  the output data structure.

An example of an application of the selection operator is using the computed node degree for the Catalano phone call network as the underlying predicate, setting a high threshold in order to get the subgraph of most active members of the movement.

*Set Aggregation:* We can union the sets of nodes and aggregate the sets of edges in each partition  $\mathcal{D}_i$ , of  $\mathcal{D}$  after we have partitioned it using a grouping function of your choice. Given a collection of graphs,  $\mathcal{D}_i$ , a set aggregation applied to it, based on a predicate  $\beta$ , is written as  $\varphi_\beta(\mathcal{D}_i) = \{\bigcup_i \mathcal{D}_i | \beta(\mathcal{D}_i) = TRUE\}$  for some  $i \in \{1, \dots, n\}$ . The aggregation predicate is evaluated independently for each set, and the entire set  $\mathcal{D}_i$  is either contained or discarded in the aggregated set depending on the truth value of this predicate.

An example related to the Catalano network is aggregating the daily data structures  $\mathcal{D}_i$ ,  $i = 1, 2, \dots, 10$  to a couple of them covering the periods of days 1-7 and 8-10, respectively (for a justification see Section 4). Another example would be to cluster similar nodes according to some of their characteristics.

*Element Aggregation:* This operator includes summations, counts and averages. In addition, we allow sampling as a form of aggregation that returns a subset of elements sampled according to the specified mechanism. Formally, we have  $\mathcal{D}' = \mathcal{A}(\mathcal{D} | Z)$ , where  $\mathcal{D}$ ,  $\mathcal{D}'$  are defined as above and  $Z$  is either an attribute or the graph itself that is being aggregated. The following is the element aggregation operator:  $\mathcal{D}' = \mathcal{A}(\mathcal{D} | Z) = \bigcup_i \{X_i \in \mathcal{D} | X_i = Z\}$ , where  $X_i = Z$  is a generic predicate.

*Graph Partitioning:* This operator is the "inverse" of aggregation with disjoint subsets. Each graph element of  $\mathcal{D}$  is partitioned into multiple subgraphs based on the value of appropriate predicates,  $\gamma$ , defined or computed on its nodes and edges.  $\mathcal{P}_\gamma : \mathcal{D} \rightarrow \{\mathcal{D}_1, \dots, \mathcal{D}_n\}$ , where  $\mathcal{D}_i \cup \mathcal{D}_j = \emptyset$  for all  $i \neq j$ .  $\gamma$  is the partitioning predicate. It evaluates independently for each element of the set and assign it to its own subclass.

An example related to the Catalano network is using the cell phone tower locations to split the network into three disjoint sets for tracking the geographical movements.

In addition, for accomplishing the visual analytic task we introduce a visualization operator  $\mathcal{V}$  whose role is to provide a visual

representation of the underlying data structure of interest. The visualization operator can take various forms, including different ways of laying-out graphs [8], e.g. force-directed, hierarchical, hyperbolic, and also presenting the attribute data (e.g. histograms for numerical attributes, bar-graphs for categorical ones, etc...). In addition, it is assumed that the visual operators can be composed and thus produce multiple and possibly linked displays. Note that  $\mathcal{V}$  is not an operator in the algebra, in that it does not have the closure property – it is a special operator, applied last, and used to create visual presentations. Its output is not a collection of graphs.

In order to accomplish the required visual analytic task, we need to apply multiple operators in sequence. In the presentation of the case studies, several such sequences are introduced and analyzed. Specifically, the final finding will usually be the results of a sequence of selection and aggregation operators; formally, the data structure  $\mathcal{D}^*$  from which the finding is obtained is given by  $\mathcal{D}^* = \mathcal{S}(\mathcal{S}(\mathcal{A}(\dots \mathcal{A}(\mathcal{D} | Z))))$ .

## 4. ANALYSIS CASE STUDY

We analyze the workflows of the Cell Phone Mini-Challenge from VAST 2008 [4]. This challenge requires analysis on a set of 400 unique cell phone call records over a ten-day period to learn the Catalano social network structure. The data set includes 9834 phone records with the following fields: calling phone identifier, receiving phone identifier, date and time, duration, and the cell tower of the call origin. A map is also provided to show the rough locations of the cell towers throughout the island region. The purpose of the challenge is to identify the Catalano/Video social network at day ten and to characterize the social structure changes throughout the time period. The first part of the challenge requires identifying Ferdinando Catalano, Estaban Catalano, David Video, Juan Video, and Jorge Video. Along with the data, the challenge provides a lead that Ferdinando Catalano is identifier 200. Also Ferdinando calls his brother, Estaban, most frequently. Finally, we know that David Video coordinates high-level activities and communications within the network.

Most competition submissions interpreted the challenge information as a static graph where nodes represent people and directed edges represent a call transaction. Competition entries also translated the challenge clues into a graph interpretation. For example, to find Estaban, a common method is to search within identifier 200's neighborhood for the node  $X$  with the most number of edges between 200 and  $X$ .

If we preprocess the data set to convert it from a multi-edge to a simple-edge graph by merging common directed edges between nodes and use a force directed layout on Cytoscape [1], we produce the following hairball network shown Figure 1. This graph displays the entire ten day period data set with node 200 and its neighborhood nodes magnified and colored black. We also computed common metrics of the data set in Table 1. The entire time period data set forms one connected component.

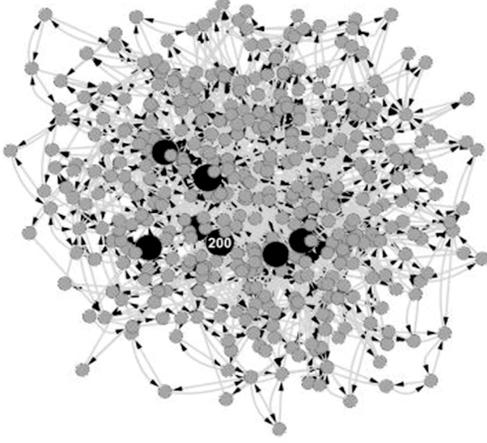
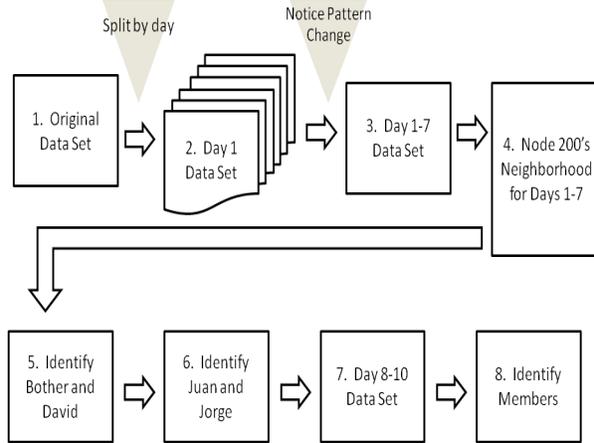
From this presentation of the network, it is impossible to complete the challenge tasks. There are too many nodes and edges to even grasp any of the attribute properties such as time, cell tower location, or duration of the call. Of the 22 competition entries, we analyze and interpret into our framework the workflows of two winners: MobiVis and GeoTemporalNet.

### 4.1 Case Study 1: MobiVis Entry

The first workflow we analyze through our framework is from the MobiVis entry [11]. MobiVis is a visual analytics system for social and spatial information. For social networks, MobiVis uses graphs where nodes represent entities and edges represent a rela-

**Table 1: Data Set Network Properties**

Measurement	Value
Number of Nodes	400
Number of Edges	9834
Clustering Coefficient	0.02
Connected Components	1
Network Diameter	12
Characteristic Path Length	4.832

**Figure 1: Hairball of entire dataset with node 200 and its neighbors colored black.****Figure 2: Workflow.**

tion or association. MobiVis filters on attributes and time to help understand large data sets. We trace the data manipulation steps used in MobiVis' solution to the challenge. At stages which involve human intuition and decisions based on visual interpretation of a display, we formalize the method into our framework. Figure 2 shows the high level workflow. For each stage we provide the analysis and algebraic representation of the data set and operations. The algorithm overview below shows the algebraic expressions that correspond to the workflow in Figure 2. In the following section we describe each of these steps more thoroughly by incorporating the MobiVis analysis.

**Algorithm 4.1: WORKFLOW(*Dataset*)****1: Read in the original dataset.**

$$D^0 = \{G = (V, E), X_E = (X^{date}, X^{duration}, X^{tower})\}$$

**2: Compute node count.**

$$X_G^{Vcount} : \sum_i 1_{\{v_i \in V\}}$$

$$D^0 = \{G = (V, E), X_E, X_G^{Vcount}\}$$

**3: Split dataset into ten days.**

$$\mathcal{P}_\gamma : D^0 \rightarrow D^1$$

$$\gamma = date$$

$$D^0, D^1 = \{D_1^1, \dots, D_{10}^1\}$$

**4: Element select node 200 subgraph for each day.**

$$D^2 = S(D^1 | X_i = \tau_{200})$$

$$\tau_{200} = \text{One of the neighbors of 200}$$

$$D^0, D^1, D^2 = \{D_1^2, \dots, D_{10}^2\}$$

**5: Set aggregation on days 1 - 7.**

$$D^3 = \varphi_\beta(D_i^1) = \{\bigcup_{i=1}^7 D_i^1 | \beta(D_i^1) = TRUE\}$$

$$\beta = \text{days 1-7}$$

or element selection based on days 1-7

$$D^3 = S(D^0 | X_i = \tau_{1-7})$$

$$\tau_{1-7} = \text{days 1-7}$$

$$D^0, D^1, D^2, D^3$$

**6: Element select node 200 subgraph for days 1-7**

$$D^4 = S(D^3 | X_i = \tau_{200})$$

$$D^0, D^1, D^2, D^3, D^4$$

**7: Identify members**

$$Estaban = \max_i X_{symmfreq}[i]$$

$$David = \max_i X_{freq}[i] = \arg \max_i \{X_{in} + X_{out}\}[i]$$

...

**8: Repeat identification for days 8-10 data set.**

At the first stage, the original data set is given by  $D^0 = \{G = (V, E), X_E = (X^{date}, X^{duration}, X^{tower})\}$ , where the nodes represent the identifiers in the call records and edges represent call records between identifiers.  $X^{date}$ ,  $X^{duration}$ , and  $X^{tower}$  are intrinsic edge attributes, directly inserted from the original data set. The choice between defining attributes as node versus edge attributes depends on the application. For example, the duration of a call is associated with a phone call transaction; therefore, duration is more appropriate as an edge attribute.

As we discussed earlier, it is impossible to answer the challenge questions when we view the whole time period data set at once. The MobiVis entry also sees this difficulty when they display the original data set and provide a node count. To perform these actions in our framework we call  $\text{Display}(G)$  using a suitable layout. We can also compute a network attribute,  $X_G$ , to represent the node count. We compute this by first aggregating the number of nodes:  $\sum_i 1_{\{v_i \in V\}}$ . Next we store this as a network attribute  $X_G^{Vcount}$ .

Based on the challenge hint, the MobiVis entry examines node 200's properties and interactions separately for each day. Since the original data set represents a ten day time period, a per day analysis is a logical choice for further examination. To perform a split on the original data set, we use partition operator  $\mathcal{P}_\gamma : D^0 \rightarrow D^1 = \{D_1^1, \dots, D_{10}^1\}$  based on the selection predicate  $\gamma = \text{date}$ .

Now we have the following collection of new data sets  $\{D_1^1, \dots, D_{10}^1\}$ . After creating new data sets, we must decide whether to re-compute attributes. Intrinsic attributes carry over; for example, the duration of a call record does not change if the call record is placed in a different data set. However, the number of nodes in each data set changes. So we recompute the network node count attribute,  $X_{G_i^1}^{Vcount}$  for each new data set created.

MobiVis filters all nodes except for the neighborhood of node 200 in their per day examination. For a similar effect, we select the subgraph for node 200's neighborhood on each day's data set. Again we use the element selection operator with predicates  $D^2 = S(D^1 | X_i = \tau_{200})$ , where  $\tau_{200} = \{\text{One of the neighbors of } 200\}$ , and produce the following set of graphs:  $D^2 = \{D_1^2, \dots, D_{10}^2\}$ . In Figure 3 we display each of these neighborhood subgraphs. We see that node 200 is active until day 8. On day 8, there is no communication and afterwards its call pattern changes. The MobiVis entry realizes this conclusion by observing the total duration of calls per day for node 200. In our framework, we produce this set of total duration values by aggregating the duration of calls on the adjacent edges to node 200 for each day data set and displaying the results. This leads to the next stage in the workflow: consider the first seven days and last three days separately.

At stage 3 of the flowchart, the MobiVis entry analyzes the first seven days. Storing all previous data sets allows us now to create a data set of call records between days 1 and 7. In our algebra we can either merge days 1 through 7 data sets by set aggregation:  $D^3 = \varphi_\beta(D_i^1) = \{\bigcup_{i=1}^7 D_i^1 \mid \beta(D_i^1) = \text{TRUE}\}$ . Or we can select the first 7 days from the original data set by element selection:  $D^3 = S(D^0 | X_i = \tau_{1-7})$  where  $\tau_{1-7} = \text{days } 1-7$ . Both methods result in the following data set of days 1 through 7,  $D_{1-7}$ . Again, MobiVis zooms into the neighborhood of node 200. We select the subgraph of node 200 in the days 1 through 7 data set,  $D^4$ .

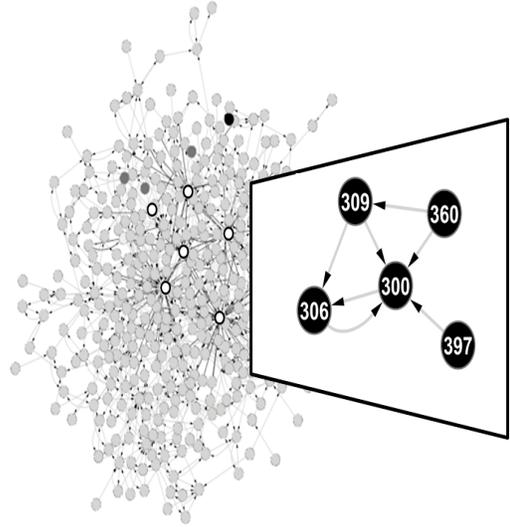
In stage 5 of the workflow, MobiVis reaches a human readable display of the graph and begins identifying the members of the Catalano/Vidro network. Their identification at this point is done by visually inspecting the graph. Assuming that 200 is Ferdinando, they identify the brother by selecting the node in 200's neighborhood with maximum symmetric frequency. The symmetric frequency for vertex  $i$  is defined as follows:

$$X_{1-7}^{symmfreq} = |(X_{1-7})_{in}[i] - (X_{1-7})_{out}[i]| \text{ where,}$$

$$X_{in}[i] = \sum_{j, j \neq i} (A_{1-7})_{ji}$$

$$X_{out}[i] = \sum_{j, j \neq i} (A_{1-7})_{ij}$$

The maximum symmetric frequency is defined as a function on



**Figure 4: Days 8-10 network and Catalano/Vidro subnetwork. The days 1-7 social network identified are the dark grey nodes. The white nodes represent the new social network with their adjacent edges darkened.**

$X^{\text{symm freq}}$ , where  $\mathcal{F}_{X^{\text{symm freq}}} : \mathbb{R}^{400} \rightarrow \mathbb{R}^1$ , for calculating the max of the attribute. This results in node 5 identified as Estaban, Ferdinando's brother. The challenge clue states that David is a highly active member of the network. Therefore, MobiVis identifies David by selecting the node with the maximum frequency:

$$\arg \max_i X_{freq}[i] = \arg \max_i \{X_{in} + X_{out}\}[i].$$

Hence, node 1 is David. Once David and Estaban are identified, the remaining two neighbors of node 200 are identified as Juan and Jorge. There is not enough information to distinguish between the two. In our framework, we find these nodes by selecting the next two maximum frequency nodes in 200's neighborhood subgraph:  $D^4$ . Juan and Jorge are identified as nodes 2 and 3.

#### 4.1.1 MobiVis Days 8-10 Analysis

To determine the social structure after day 8, the MobiVis entry examines the data set of days 8 through 10 merged. This data set is created similar to days 1 through 7 data set computed earlier. The MobiVis entry hypothesizes that the elite members switch phones after day 8. They support this hypothesis by searching for a subgraph in the days 8 through 10 network which resembles the 200 neighborhood found in days 1 through 7. The difficulty here is we cannot rely on the lead that node 200 is Ferdinando. To identify this subgraph, MobiVis visually inspects the days 8-10 data set to remove nodes with low frequency of communication until they identify a neighborhood similar to the subgraph of node 200 during the first seven days. In our framework, we rank edges by frequency of communication, and display the graph. Given this display, selections to remove lower frequency nodes can be iteratively performed until we find a similar network. Figure 4 shows the subgraph MobiVis finds on days 8 through 10, which they guess is the new Catalano/Vidro network. The nodes in this subgraph are 309, 306, 360, 397, and 300. Visually they map the members of this subgraph to the ones from days 1-7. We translate their visual mapping to our algebra.

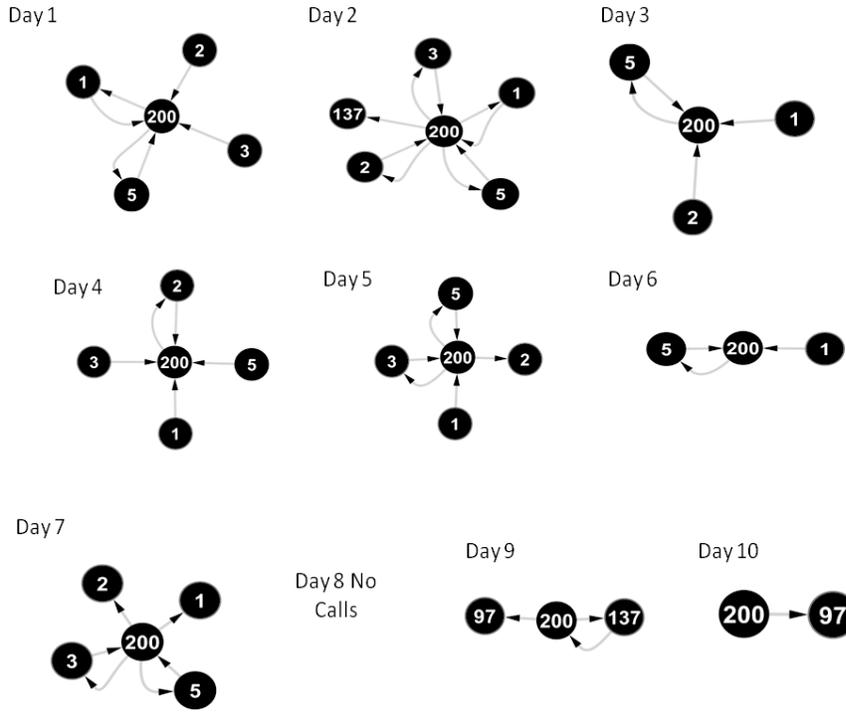


Figure 3: Node 200's neighborhood through the time period.

From this subgraph MobiVis identifies David by visually selecting the node with the highest frequency of communication. This is equivalent to a max frequency operation,  $\arg \max_i \{X_{1-7}\}_{freq}[i]$ , resulting in node 309. To find Ferdinando and his brother they find the two nodes in this network with high symmetric frequency. Ferdinando is identified as the node with connections to Juan and Jorge which are mapped to 397 and 301 from visual inspection. This method to determine the new subgraph in the days 8-10 data set might be time consuming if the data set is noisy or contains several possible subgraphs. We notice that David in days 1-7 has the highest degree in the network. If we simply select the node in the merged days 8-10 data set with the maximum degree, we can narrow the choices for possible subgraphs. After computing maximum degree,  $\arg \max_i \sum_{j, j \neq i} \{A_{1-7}\}_{ij} + \{A_{1-7}\}_{ji}$ , we find David is node 309. On the days 1-7 data set, a selection of node 309 results in null. This result might support the hypothesis that the phones were indeed replaced after day 7.

In the days 1-7 data set, Estaban and David have the most number of common neighbors. Again if we assume that the social structure during days 1-7 remains similar for the new set of phones, we can identify Estaban by selecting the node with the most number of common neighbors with David. This computation is done by creating a new node attribute: common neighbors with node 309. The common neighbor computation is the intersection of nodes in two rows of the adjacency matrix. For each node we store its common neighbors with the fixed node 309. We perform a maximum aggregation operator on this attribute  $\arg \max_i \{X_{8-10}\}_{common\ neigh. 309}[i]$ . Estaban is node 306.

Now we can find Ferdinando since we know Estaban is his highest interactivity neighbor. The final mapping produced is David: 309, Estaban: 306, Ferdinando: 300, and Juan and Jorge: 397 and 360. The algebra in coordination with the display helps support

our hypothesis and decisions at each stage. While the MobiVis entry does a visual inspection to support their identification mapping, providing the exact operation helps trace the stages in the workflow.

## 4.2 Case Study 2: GeoTemporalNet Entry

After the competition deadline, VAST never released correct answers for the challenges. The reason for not publishing the answers is to allow open interpretation of the data sets. In the above analysis, we see the interpretation of the data set given an analyst uses MobiVis. However, other winning entries delivered different conclusions. The differences in tools and interpretation of the challenge hints lead to unique results for the same data set.

The challenge hints must be translated from a word sentence to a network property. There was a different degree of open interpretation for these hints. For example, the first hint that 200 might be Ferdinando has a direct network interpretation: identify node 200 in the graph as Ferdinando. David's hint is that he coordinates high-level Paraiso activities and communications. This hint does not have a direct network interpretation. What are considered high-level activities? What does it look like in the graph to coordinate these activities?

We present the GeoTemporalNet entry [14], also a winner, for the cell phone mini-challenge. Instead of analyzing their workflow from the start, we describe the differences in analysis and results between GeoTemporalNet and MobiVis. As we will see, our algebraic framework provides a superset of operations used by MobiVis and GeoTemporalNet. We can use our framework as a linking language between the two tools. Therefore, again we analyze the steps, this time for GeoTemporalNet, within our framework.

The GeoTemporalNet entry used a combination of tools: JS-NVA (Java Straight-line drawing Network Visual Analysis framework) and TemporalNet. JSNVA is a software framework for net-

**Table 2: Identification of Catalano/Vidro Network**

Member	GeoTemporalNet	MobiVis (1-7)	MobiVis (8-10)
Ferdinando	200	200	300
Estaban	5	5	306
David	0	1	309
Jorge	1/2	2/3	397/360
Juan	1/2	2/3	397/360

work visual analysis in different applications. The GeoTemporalNet group developed TemporalNet within JSNVA to show communication patterns in call graphs. They use a static graph with nodes representing people and edges representing calls for the social network. Like MobiVis, they use force-directed graphs in their layout.

The first notable analysis difference between MobiVis and GeoTemporalNet occurs at stage 5 in the workflow: the identification of David and Estaban. GeoTemporalNet finds David through a most common neighbor element selection operation with node 200. They apply this operator on the original data set, which includes the entire time period.

Common neighbors( $i, j$ ) =  $N(i) \cap N(j)$  where,

$N(i)$  = list of neighbors for vertex  $i$

$N(j)$  = list of neighbors for vertex  $j$

The common neighbor operation is performed between a pair of nodes in a graph. As described earlier, common neighbor finds the intersection of two rows in the adjacency matrix. In this case, the node pair is 200 and each of the other nodes in the graph. We store a new node attribute for the number of common neighbors with node 200. Then we apply a maximum function to return the node with the most number of common neighbors. This function leads GeoTemporalNet to believe David is node 0.

$$\text{David} = \arg \max_i \text{Common neighbors}(i, 0) \text{ for all } i \in V$$

$$\text{where } V \in \mathcal{D}^0$$

The challenge instructions never explicitly state that David and Ferdinando communicate directly. During the ten day period, node 0 and 200 never call each other. MobiVis does not compute common neighbors between nodes. Also, they filter nodes to observe only node 200's neighborhood subgraph. Therefore, their tool cannot identify node 0 as a possible result for David.

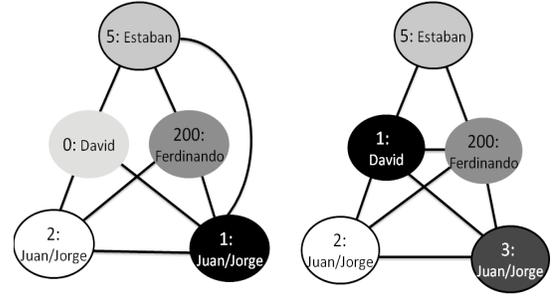
On the other hand, GeoTemporalNet does not filter the graph to zoom into only node 200's subgraph. In addition, they interpret the challenge hint differently. As a result, MobiVis and GeoTemporalNet provide different answers. However, as we have shown, our framework captures both methods and can arrive at both answers. The limitations of the tool on the analyst's interpretation are removed.

GeoTemporalNet identifies the other members, Estaban, Juan, and Jorge, similar to MobiVis' method. These are a series of selections for the nodes with most frequent communication with node 200. Again they perform this operation on the original data set.

The final mapping GeoTemporalNet produces is: Ferdinando: 200, David: 0, Brother: 5, Juan and Jorge: 1 and 2. Figure 5 shows the identification differences between GeoTemporalNet and MobiVis for the Catalano/Vidro network.

#### 4.2.1 GeoTemporalNet Days 8-10 Analysis

GeoTemporalNet does similar per day analysis as MobiVis to



**Figure 5: Mapping between MobiVis and GeoTemporalNet networks.**

discover node 200's calling pattern changes after day 7. However, GeoTemporalNet does not guess that the members replaced phones to explain this pattern change. In their work, they assume that a person is never assigned a new phone. Instead they hypothesize that new members entered the Pareto movement. These new members have equivalent roles as some of the high-level members identified in the problem: Estaban, Jorge, and Juan. They support this hypothesis by computing the Jaccard coefficient,  $J(i, j) = \frac{|N(i) \cap N(j)|}{|N(i) \cup N(j)|}$  for any vertices  $i$  and  $j$ , of these new members with their member of equivalent societal role. The Jaccard coefficient is a vertex similarity metric to measure the structural equivalence between two nodes. This metric is simply the number of common neighbors normalized. The GeoTemporalNet entry does not state how they identify the new set of members: 309, 306, 360, and 397; or their equivalent role pair: (1,309), (5,306), (3,360), and (2, 397). However, they compute and display the Jaccard coefficient for these pairs of nodes. They state that the high Jaccard coefficient leads them to believe these pairs have equivalent roles; therefore, the later appearing nodes may be replacements for the previous nodes.

In our framework to produce the same support we do the following operations: First we create a Jaccard coefficient attribute for the following node pairs: (1,309), (5,306), (3,360), and (2, 397). Since we are not interested in computing the Jaccard coefficient for all node pairs in the graph, we can create a set of network attributes for these node pairs,  $X_G^{\text{common neighbors}(i,j)}$  for all  $i$  and  $j$  in set  $\Omega = \{(1, 309), (5, 306), (3, 360), (2, 397)\}$  where  $G \in \mathcal{D}^0$ .

After analyzing the MobiVis and GeoTemporalNet entries, we see two different workflows. Our algebraic framework captures both methods and is also capable of filling in the intuition gaps with algebraic operations.

## 5. NEW FINDINGS

Of course, the algebraic model we propose is capable of representing analyses beyond the specific examples studied in the preceding section. In this section, we present one such analysis instance, showing a new analytic finding on the Catalano data set, one that was not reported by any of the teams participating in the VAST08 challenge.

### 5.1 Social Structure

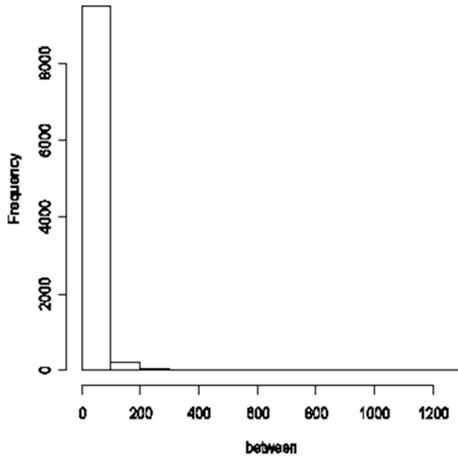


Figure 6: Edge betweenness for the original network .

On each step of the algebraical model, we compute graph related attributes for the set of graphs and denoted as  $X_G$ . Among all the computed metrics, the most important attributes for identifying the social structure are edge betweenness and clustering coefficient [2].

The edge betweenness is defined as  $B_u = \frac{g(i,u,j)}{g(i,j)}$ , where  $g(i,u,j)$  is the number of shortest paths between vertices  $i$  and  $j$  that pass through edge  $u$ ,  $g(i,j)$  is the total number of shortest paths between  $i$  and  $j$ , and the sum is over all pairs  $i, j$  of distinct vertices [2, 9].

The clustering coefficient is also known as transitivity [2, 9], which is based on the following definition for an undirected unweighted network:  $C = \frac{N_\Delta}{N_3}$ , where  $N_\Delta$  is the number of triangles in the network and  $N_3$  is the number of connected triples. Therefore we have

$$N_\Delta = \sum_{k>j>i} A_{ij}A_{ik}A_{jk}$$

$$N_3 = \sum_{k>j>i} (A_{ij}A_{ik} + A_{ji}A_{jk} + A_{ki}A_{kj})$$

are the elements of the adjacency matrix  $A$  and the sum is taken over all triples of distinct vertices  $i, j$  and  $k$ . Since it assigns the same weight for each triangle in the network, it can be related to the clustering coefficient for each vertex, which captures the hierarchical structure in the network.

According to the distribution of the edge betweenness [3] shown in Figure 6, there is only a small number of edges suggesting important relationships in all the graphs. Further, almost all clustering coefficients are small and low in transitivity, for all graphs during the ten day period. The combination of these findings strongly supports the existence of a hierarchical structure within the Catalano network.

## 5.2 Geographical location and movement

After the discovery of a change in the social structure after day 7, we examine the geographical location of the main actors in the network, as well as their movement in the 1-7 day and 8-10 day periods. The proposed framework can easily address such issues as shown next.

Based on the map of Isla del Sueno, we decided to partition the thirty cell phone towers on the island into three groups, represent-

ing the Upper, Middle and Lower sections of the island. This can be accomplished by applying an element aggregation operator to the towers' location. Since one of the available attributes corresponds to the cell tower used by the phone call's originator, it is possible to track the caller's movement throughout the ten day period. However, due to the finding of a hierarchical network structure, we focus on the leadership group formed around Ferdinando.

We start by selecting the nodes corresponding to the leadership group for each day  $i$ . Specifically,  $\tilde{D}_i^l = S(D_i^l | X_i = \tau)$ , where  $\tau =$  subset of  $\{1, 2, 3, 5, 200, 300, 306, 309, 360, 397\}$  for day  $i = 1, \dots, 10$ . Notice that for not all members of the leadership group made phone calls on every day. However, some broad patterns emerge from our analysis, as shown in Figure 7. It exhibits the geographic location of the leadership group at different days. The plots are constructed based on the bipartite graph defined by the caller's ID and the grouped (Upper, Middle, Lower) location of the cell tower employed. Further, the length of the edges is weighted by the call frequency of each node for that day; hence, shorter edges indicate a more active call pattern with regards to the tower under consideration, and longer ones a less active pattern.

It can be seen that on day 2, Ferdinando is located in the Middle section of the island, while his brother Esteban and Juan in the Upper section. This pattern holds for days 1-7. On the other hand, a more mobile pattern emerges for days 8-10 (operating under the assumption that node 300 is Ferdinando, 306 is Esteban, 309 is David, etc. For example, it can be seen that although Ferdinando remains stationary in the Middle of the island, Esteban moves from the Upper section in day 8 to the Middle section in subsequent days, while David shares his time between the Middle and Lower sections on days 8-9, but stays in the Lower one on day 10.

Figure 8 summarizes the location changes of selected members of the leadership group throughout the ten day period. The following conclusions can be reached.

- The person with IDs 200 and 300 is Ferdinando Catalano. He spends the entire ten day period in the Middle section of the island.
- Ferdinando's brother Esteban Catalano, corresponds to IDs 5 and 306. He spends most of his time in the Upper section of the island for the first eight days, while for the last two days he moves to Middle section and is co-located with his brother.
- David, a rather active member of the Catalano network, has IDs 1 and 309. He spends the first seven days in the Middle section, but starts visiting the Lower section on days 8 and 9 and stays there the entire length of day 10.
- A similar pattern to David's emerges for Juan and Jorge that exhibit more movement after day 8.

Our analysis provides additional insight on the location and movements of the leadership group of the Catalano network. It is worth noting the increased activity of several members (but not Ferdinando) over the last three days. Nevertheless, without additional information, it is hard to assess the significance of these movements regarding the activities of the network.

## 6. RELATED WORK

As the VAST Challenge demonstrates, there are several visual analytic tools with different capabilities for geospatial activity and behavior, text processing, and social network analyses. We focus on just a few references that particularly deal with the visual

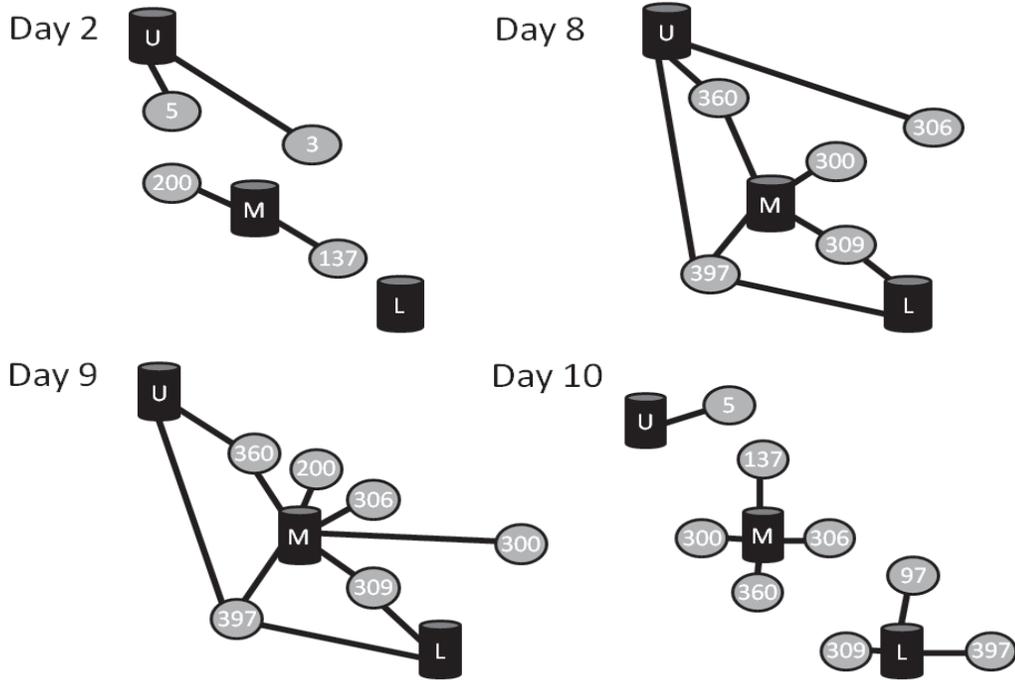


Figure 7: The location changes for elite members in Catalano family.

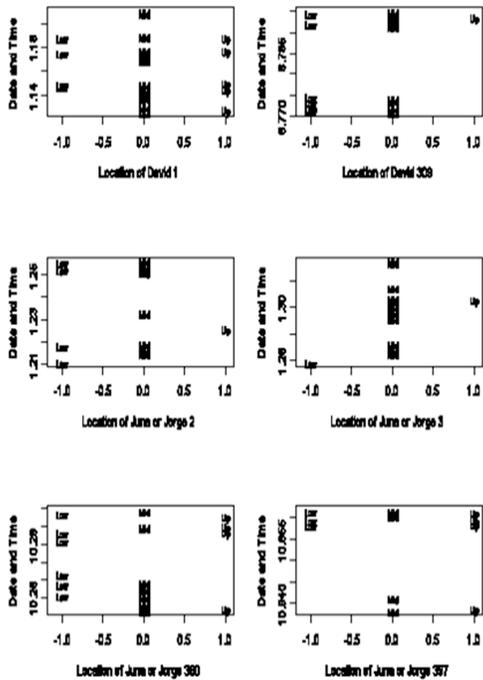


Figure 8: The individual geographical movement.

analytics of graphs. In [13], a human-centric design approach is adopted to create a tool for descriptive graphs, while in [12] information about the local topology of a graph is captured in a signature that aids exploration of graphs. In [6], interactive exploration of networks is undertaken through enhanced layouts, while in [10] semantic and structural abstractions are used for analyzing social networks. Data traffic is explored through network maps in [7] and Internet routing changes in [5]. A key point to observe is that while there are several systems that have been very effective in providing better support for visual analytics of network data in a particular application context, no one has attempted to develop a formal foundation on which to construct such systems. This is what we aim to do. Thereby, we hope to be able to support a broad range of applications rather than just one.

## 7. CONCLUSIONS

With our proposed algebraic model we can represent a large class of visual analytic operations on graphs, as we demonstrated through analysis of the VAST 2008 cell phone mini challenge. For future work, we plan to consider the computational issues involved in efficiently implementing our model and issues involved in incorporating this framework into a tool.

## 8. ACKNOWLEDGEMENTS

The research is supported in part by NSF grant numbers 0438909 and 0808824 and NIH 1-U54-DA021519.

## 9. REFERENCES

- [1] The Cytoscape Collaboration. *Cytoscape Users Manual*. The Cytoscape Collaboration, Institute for Systems Biology and University of California San Diego, 2006.

- [2] Luciano Costa, Francisco Rodrigues, Gonzalo Travieso, and Villas Boas. Characterization of complex networks. *Advances in Physics*, 56(1):167 – 242, May 2007.
- [3] Gabor Csardi. *igraph*. R User Manual, CRAN, 2009.
- [4] G Grinstein, C. Plaisant, S. Laskowski, T. O'SConnell, J. Scholtz, and M Whiting. Vast 2008 challenge: Introducing mini challenges. *Proceedings of IEEE Symposium*, 1(1):195 – 196, October 2008.
- [5] Mohit Lad, Dan Massey, and Lixia Zhang. Visualizing internet routing changes. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1450 – 1460, November 2006.
- [6] Bongshin Lee, Cynthia S. Parr, Catherine Plaisant, Benjamin B. Bederson, Vladislav D. Veksler, Wayne D. Gray, and Christopher Kotfila. Treeplus: Interactive exploration of networks with enhanced tree layouts. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1414 – 1426, November 2006.
- [7] Florian Mansmann and Svetlana Vinnik. Interactive exploration of data traffic with hierarchical network maps. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1440 – 1449, November 2006.
- [8] George Michailidis. *Data Visualization Through Their Graph Representations*. Springer, Berlin, 2006.
- [9] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(1):167 – 256, March 2003.
- [10] Zeqian Shen, Kwan-Liu Ma, and Tina Eliassi-Rad. Visual analysis of large heterogeneous social networks by semantic and structural abstraction. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1427 – 1439, November 2006.
- [11] Visualization and Interface Design Innovation (VIDI) Group. Intuitive social network graphs visual analytics of cell phone data using mobivis and ontovis. *IEEE Symposium on Visual Analytics Science and Technology*, 1(1):19–24, October 2008.
- [12] Pak Chung Wong, Harlan Foote, George Chin Jr, Patrick Mackey, and Ken Perrine. Graph signatures for visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1399 – 1413, November 2006.
- [13] Pak Chung Wong, Harlan Foote, Patrick Mackey, Ken Perrine, and George Chin Jr. Generating graphs for visual analytics through interactive sketching. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1386 – 1398, November 2006.
- [14] Qi Ye, Tian Zhu, Deyong Hu, Bin Wu, Nan Du, and Bai Wang. Exploring temporal communication in mobile call graphs. *IEEE Symposium*, 1(1):16 – 19, October 2008.