

ChurnVis: Visualizing Mobile Telecommunications Churn on a Social Network with Attributes

Daniel Archambault, Neil Hurley, and Cuong To Tu

Clique Strategic Research Cluster

University College Dublin

Email: d.w.archambault@swansea.ac.uk, neil.hurley@ucd.ie, totucuong@gmail.com

Abstract—In this paper, we present ChurnVis, a system for visualizing components affected by mobile telecommunications churn and subscriber actions over time. We describe our experience of deploying this system in a network analytics company for use in data analysis and presentation tasks. As social influence seems to be a factor in mobile telecommunications churn (the decision of a subscriber to leave a particular service provider), the visualization is based on a social network inferred from calling data between subscribers. Using this network, churn components, or groups of churners who are connected in the social network, are segmented out and trends in their static and dynamic attributes are visualized. ChurnVis helps analysts understand trends in these components in a way that respects the data privacy constraints of the service provider. Through this two pipeline approach, we are able to visualize thousands of churn components filtered from a social network of hundreds of millions of edges.

Keywords—Telecommunications Churn, Attributed Graphs, Visualization, Social Networks

I. INTRODUCTION

In the mobile telecommunications consulting industry, the analysis and prediction of subscriber churn is an important and valuable activity. A customer of a mobile telecommunications provider **churns** if they decide to leave that provider. There is some evidence [24], [10] that a factor in deciding to leave a given service provider is social (ie. the decision is influenced by the fact that friends of the subscriber in the social network have previously left the network). However, social influence is not the exclusive factor in deciding if a customer churns. Quite frequently, extrinsic attributes associated with the nodes/edges and intrinsic network attributes can influence a subscriber's decision to churn. Thus, the visualization of the evolution of churn phenomena in the context of a social network with attributes is of interest to our industrial collaborators. Attributes associated with the nodes and edges of the graph in this data can be both static and dynamic. **Static attributes** do not evolve over time (gender, birth date etc.) or can be represented as an average/most frequent value (most frequent calling location). **Dynamic attributes** are attributes that evolve over time and cannot be averaged (call activity or adoption of a particular model of phone).

Two of the principal challenges of this problem are that of data set size and variance in analyst expertise. The graphs that are dealt with on a daily basis by telecommunications analytics companies are very large. In our initial interviews with our collaborators, a data set of four million nodes was considered small. In this work, the largest input graph we consider has

close to one billion edges. Secondly, both telecommunication analysts and CEOs must be able to understand the summaries produced by the visualization system. These users have varied expertise in visualization literacy and it must be possible to quickly understand the system in exploratory and explanatory contexts.

The contribution of this work is ChurnVis, a system developed to visualize subscriber churn and associated attributes, in a large social network over time. Our design relies heavily on pixel oriented displays [14], [15] to illustrate a large number of attributes and their values simultaneously, allowing the analyst to discover trends in the data. We use the design study methodology [21] in order to implement and refine the visualization system design for our tasks. The system was deployed for a week at the offices of our collaborators, and analysts were able to use it to discover expected trends and features in their data as well as a few anomalous behaviours.

II. RELATED WORK

A. Graph Visualization Systems

The field of graph visualization has a long history with many techniques for understanding the structure of graphs [13], [11], [23]. As our approach is required to handle large, attributed graphs, we focus our discussion of previous work on visualization techniques for this type of data.

A promising set of methods for visualizing attributed graphs are aggregation methods [1], [2], [3], [6]. These techniques could be used to simplify parts of the graph with specific attribute values in order to understand the overarching connectivity between different segments of the graph. Although such techniques can be quite effective, in our application, we cannot use these approaches for scalability and data sensitivity reasons.

Our work, in many ways, is closely related to work on grouping subgraphs via structural similarity. Brandes *et al.* [8] described methods to classify subgraphs based on the spectrum of their adjacency matrices. As graphs derived from the same underlying process have similar spectra, these can be used to differentiate classes of graphs. Harrigan *et al.* [12] describes a system to cluster groups of egocentric networks using motif enumeration. In this approach, for every subgraph, all the motifs up to size five are enumerated and interpreted as points in a high dimensional space that can be visualized with dimensionality reduction techniques. Both of these techniques can be used to visualize the structural similarity of many subgraphs. Although either system could be extended to handle attributed

graphs, their focus is the grouping of graphs by structural similarity. In our problem, similarity based on attributes is more interesting for our intended users.

A number of other approaches exist to visualize the structural and attribute similarity of groups of subgraphs. Brandes *et al.* [7] designed a system to investigate trends in egocentric, social networks to study acculturation of migrant workers to the USA and Spain. The work is able to visualize classes of friends of these workers in order to illustrate patterns in relationships the migrant has with their host culture. The visualization was subsequently applied to help sociologists understand trends in this population [18]. Although this work uses both structural and attribute data on many networks, we focus mainly on attribute data in our work.

Our work, in spirit, mostly resembles that of von Landesberger *et al.* [22]. In her work, several properties of a network, associated both with the structural properties of the network and the attributes associated with the nodes and edges, are used to create a feature vector describing the subgraph. These feature vectors can be interpreted as points in a high dimensional space that can be clustered using self-organizing maps. In many ways, our system is similar to this technique but focuses mostly on the attribute values associated with each of these subgraphs.

B. Pixel Oriented Displays

Our visualization technique heavily relies on pixel oriented displays [14], [15] in order to visualize the attributes common to particular components. Pixel oriented displays encode each attribute value as a small rectangle. As each data value takes a small amount of area, many attribute values of each component can be visualized simultaneously.

Our pixel oriented display is very similar to the pixel bar charts of Keim *et al.* [16]. In this work, rather than presenting the summaries of all customers in a bar chart, the authors present the attribute values of each customer as a pixel oriented display. By avoiding aggregation, the authors are able to present trends in the attribute values of individual customers. In our approach, one could view our components of subscribers as the customers in Keim *et al.* [16]. By using this technique, we have the advantage that network privacy can be maintained and the visualization can be focused on the attribute values associated with the components present in the graph.

C. Churn and Mobile Telecommunications Analytics

One of the basic assumptions of our work is that social influence is a factor in mobile communications churn [24], [10]. Essentially, this implies that a decision of a subscriber to leave a mobile telecommunications provider, is likely to encourage other friends of the subscriber to also leave the network. In this context, friends are subscribers that are nearby, in a graph distance sense, on a social network described by calling behaviour. When this assumption is true, we can expect that churn will tend to propagate through the network, from friend to friend, similar to a diffusion process. It is important to note that our work is not aimed at the problem of predicting churn. Rather, the assumption of social churn motivates our data processing and visualization strategy, in which components of connected churners form natural graph segments around which to visualize the attribute space.

III. USERS, TASKS, AND DATA

In this work, our users are employees of a mobile telecommunications consulting company. These users are interested in a visualization system in order to explain and understand how churn propagates in customer bases with respect to attributes and social network structure.

In this domain, mobile telephone calls are used to derive an underlying social network whereby nodes are mobile telephones and edges link two telephones if a sufficient number of calls were made between them. This network is the underlying **social network**. **Subscribers** are actors in the social network or nodes in this graph. A **component** is a group of subscribers that share some relationship on the social network. This component could simply be a connected set of subscribers or a set of subscribers grouped together by a community finding algorithm [17]. Subscribers can have both static and dynamic attributes. Edges between subscribers can also have attributes and can be summarized per component.

The system should be able to address the following task:

- *The visualization should depict trends in components of subscriber churn in the context of the static and dynamic attributes*

Through several discussions with our users, we discovered several constraints to the design of the visualization in support of this task:

- 1) *The visualization should be based on components in terms of the underlying social network*
- 2) *The visualization approach should be able to represent social network metrics and subscriber attributes*
- 3) *If the attribute is dynamic, the visualization should be able to display changes in subscriber actions over time (e.g., the progression of churn).*
- 4) *It should be possible to grasp this visualization quickly, within a two to three minute presentation window.*

A. Challenges of this Problem

There are several challenges with respect to this task and the data as described below.

1) *User Expertise:* The system is intended for both analysis and presentation. Also, the technical expertise of the users/audience will vary greatly, and may include analysts and technically knowledgeable CEOs.

2) *Data Set Size:* The social network derived from the mobile telecommunications data is very large. In our early discussions, a social network of four million nodes was considered small. In this work, we deal with social networks with hundreds of millions of edges.

3) *Data Set Privacy:* The underlying mobile telecommunications data used in this study is real consumer information. In the data sets presented in this paper, subscriber ids are anonymized through a hash function and therefore cannot be reverse engineered. However, in certain applications they could be real. Any potential leak of information would damage the trust, credibility, and image of our industrial collaborators and potentially their customers. Attribute values for specific

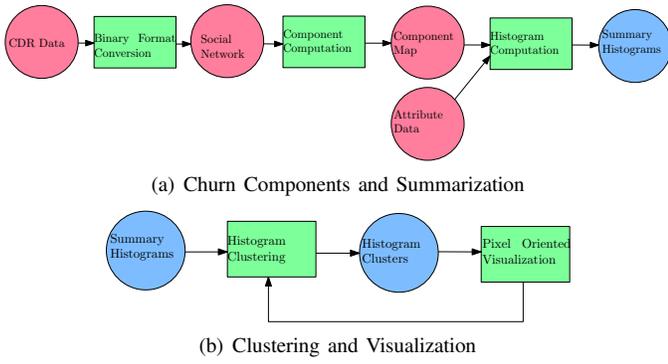


Figure 1. ChurnVis data processing and visualization architecture. (a) Data processing pipeline. This stage requires several hours to complete. (b) Visualization pipeline. As it is disconnected from the data processing pipeline, this process can be executed locally on any machine. In these diagrams, circles represent data and rectangles processing. Red circles could potentially contain private information and should remain on company servers. All blue circles can be copied locally to any machine.

subscribers, call record data, and the structure of the social network can not be copied from the machines of our industrial collaborators, including subsets of this information. However, summaries can be downloaded locally. Our collaborators wish to have an interactive system that can be used both for analysis and presentation. Thus, the system would have to run on machines other than their servers. Data must be copied locally and this constraint greatly influences our design.

IV. CHURNVIS SYSTEM

Based on the tasks and requirements set out by our collaborators as described above, we designed our system for visualizing churn based on the pipeline depicted in Figure 1. The pipeline architecture is designed in such a way to deal with the challenges described above. The **churn components and summarization** pipeline is executed on company servers. This phase takes graphs that are usually hundreds of millions of edges in size and groups subsets of nodes into churn components. From these churn components, summary histograms can be computed.

The **clustering and visualization** pipeline executes locally on a machine, usually a laptop, for discussions about the data. The input to this phase is the summary histograms for each churn component. In this phase histograms are clustered together based on the similarity of their attribute values. Representatives for each cluster are displayed, in a way similar to the work of von Landesberger [22], and the details can be displayed by clicking on each representative.

A. Churn Components and Summarization

Below, we describe the process for converting our data into the summary histograms used for visualization.

1) *CDR Data and the Social Network*: In order to compute the churn components, we must first derive the social network in order to satisfy the first constraint. **Call detail records** or **CDR data** is used to infer the structure of a social network that exists between subscribers. CDR data usually consists of an edge list containing the calling party and the called party along with the date, time, and duration of the call. The data

is usually supplied in a comma-separated, plain text format in several zipped files, describing the activity on a given day. Quite frequently, different types of communication can be listed such as SMS or Internet access (which usually does not have a called party). In our work, we use the voice CDR data in order to infer a social network.

As Figure 1(a) illustrates, the process of inferring a social network begins by converting each daily edge list of CDR data into a binary format for data compression reasons. Then, these edge lists are superimposed to infer a social network of callers which itself is stored in a binary format. This **social network** is undirected and each edge is weighted by the number of calls made between subscribers. Weak links are filtered out (communications that happened less than four times over several months), because they are poor indicators of a true social link between the calling parties. Nodes of exceptionally high degree are filtered out as well (nodes that place/receive a call on average once every ten minutes during several months of calling data). These nodes of exceptionally high degree are often calling centres and can introduce noise into the social network.

While creating this social network, it is often the case that the graph will not fit into main memory. Often, this network and its many intermediates has hundreds of millions of unique, weighted edges. In order to allow for the processing of this data, we map the binary representation of the graph into a file on the machine that is treated like virtual memory. This mapping procedure is used when creating the social network and the churn components as described below.

2) *Churn Components*: Once a clean social network has been curated and stored in binary format, this data is used to construct churn components. **Churn components** are loosely defined as sets of interesting nodes in the network. These components can be defined using standard community finding algorithms [17], overlapping community finding algorithms [19], or other methods for generating interesting groups of nodes. In the case of overlapping components, nodes are duplicated and placed in the multiple corresponding components. For all the results presented in this paper, churn components are connected components of churners in the social network computed through a breadth first search of the graph. Churn components defined in this way are easy to understand for our user community (Section III-A1), but more sophisticated definitions can be substituted for this stage.

A key output of this stage of the algorithm is a **component map**. A component map lists, for each component, the nodes the component contains. It has a unique identifier for the component and a list of node identifiers.

3) *Summary Histograms*: **Summary histograms** specify the demographics of each churn component in an aggregate and anonymous form. They are histograms showing the number of nodes in the component with a particular attribute value. The summary histograms can encode both static and dynamic data and can be based on structural properties of the social network or demographics data. Summary histograms are computed by custom programs that take attribute data and the component map as input in order to produce the demographics for the component.

In most cases, a program that generates one or more

summary histograms takes in as input the component map along with several text files encoding the attribute of the subscriber as node or link level data. Using the subscriber id associated with the node(s), we find the component(s) in which it participates and the appropriate histogram is incremented. If the subscriber has missing information for this field a value of *unknown* is entered instead. For dynamic attributes, this procedure is replicated over all time periods in the data set.

B. Clustering and Visualization

Once the churn component and summarization pipeline has finished processing the CDR and attribute data, visualization of attribute-based similarity of churn components can begin locally for presentation or analysis. The input to this phase consists of the summary histograms. The pipeline for this phase is shown in Figure 1(b).

1) *Histogram Clustering*: Before visualization begins, all churn components are clustered based on the similarity of their summary histograms. In order to perform this clustering, each component is transformed into a feature vector of high dimension. The dimensionality of this vector corresponds to all possible values for all of the attributes in the data set. The counts present in the histogram are placed in the fields of the vector and all of the vectors in the data set are clustered using k-means. This approach was used because the results are easily understood: clusters correspond to components with similar attributes. Other clustering algorithms could be substituted at this stage for analysis.

2) *Pixel Oriented Representation*: For each cluster of components, the closest component to the k-means centroid is selected as the representative for this cluster. These representatives are ordered from the cluster with largest to smallest number of components. These representatives are then depicted in the pixel oriented display shown in Figure 2.

Figure 2(a) shows the legend for the pixel oriented display used for this data set in both the representatives and details views. This legend appears at the top of both screens. Static attributes are on the left hand side of the display while dynamic attributes are on the right hand side of the display. Example cluster representatives are shown in Figure 2(b). The values that each static attribute can take on are ordered alphabetically, in the pixel oriented display, following the design of Oelke *et al.* [20] for visualizing consumer data. The values for the dynamic attributes, one per line, are ordered left to right chronologically. Mousing over a value gives the proportion of subscribers within that component with the value. If the attribute is dynamic, the date range the pixel represents is written as well.

Figure 3 shows the details view for one of these clusters of components. Three members of this cluster are shown. The pixel oriented display conveys that components in this cluster share a high propensity of *Nokia* mobile telephones (blue) with similar behaviour (grey). The component id and number of subscribers is indicated above each component.

3) *Filtering and Clustering*: Initially, all of the attribute data is used together in order to cluster the components with each dimension treated with the same weight. However, in many circumstances, our users would like to focus on one

or two attributes for clustering. Also, sometimes our users are only interested in one or two specific values for these attributes, for example, only components that are predominantly in specific cities.

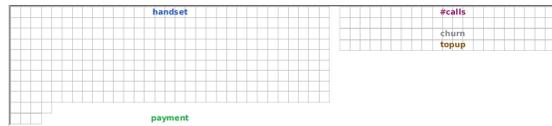
In order to support these usage scenarios, we provide the panel situated on the left hand side of the clustering view as shown in Figure 2(b). In the top left, we have a number of sliders which control the weight given to each of the attributes in the clustering. In this case, we only consider the handset and the churn attributes in the histogram clustering. Below this panel, we have a list box that controls the filtering of a given clustering. In this list box, all of the static attributes are listed and children of these static attributes in the list box are values that exist in the data set. When a number of values are selected in this widget, only those churn components that have a majority of subscribers with this value will be displayed. Through this widget, our users can adjust the clustering of churn components by attribute value and filter the display.

V. ITERATIVE REFINEMENT

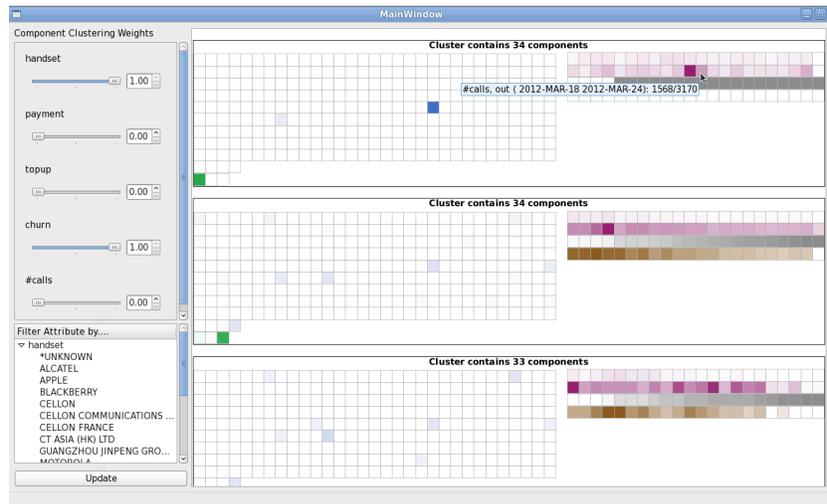
In this section, we describe our collaboration with a mobile telephone consulting company, leading to the design of ChurnVis. We began working with the group about a year and a half ago on this specific problem and developed various versions of the software to tackle the problem of visualizing churn and other attributes on mobile telephone communication networks. For most of the project, we worked with a technical-savvy member of the business side of the company and two of the engineers. In later stages of the project, we worked with two additional engineers in the company. In earlier versions of the tool we presented ideas and prototypes in meetings that occurred about once every two months. In later stages of the prototype, we met about once every two weeks in order to refine the tool in order to work specifically on the types of tasks our collaborators wanted to undertake.

In our initial meetings, we spent a fair bit of time discussing the types of problems that engineers and business-minded members face on a daily basis. The primary business of this company is the prediction of churn on social networks derived from mobile telecommunications data. As often technical and business staff at the company would like to explore the factors behind churn, we decided to investigate the problem of visualizing churn in the context of attribute values. Unanimously, they required that the developed tool be able to handle extremely large amounts of data with hundreds of millions of edges.

Our initial prototypes were heavily inspired by the work of von Landesberger [22]. We applied self organizing maps to attributes and structural properties of the social network to find subscribers and groups of subscribers with similar attributes and call behaviour. We applied both the churn components and egocentric methods to the data in order to visualize how churn was behaving on the large network. Although, in many cases, the visualization worked well, the presentation of SOMs and the clustering method itself were sometimes difficult to understand and explain to our audiences. Thus, we decided to settle on simpler clustering approaches. Additionally, we generalized the notion of churn component as it became clear that users would be interested in various types of structural



(a) Legend for Display



(b) Component Clustering Interface

Figure 2. Pixel oriented display for encoding representatives and component clusters. (a) Legend for pixel oriented display indicating the attributes and their colours. (b) Representatives for clusters in the pixel oriented display. The labels of each representative includes the number of components in the cluster.

groups as well as community finding methods that could be applied to the network.

Also, during these meetings, we discovered that attribute data associated with the nodes and edges of the social network should be emphasized. This attribute data was often more intuitive and important. For example, understanding the concentration of iPhones in particular groups of nodes in the graph seemed more important and intuitive than metrics derived directly from the social network structure. Slowly, we moved to a more generic interface which would be able to deal with generic static and dynamic attributes but still based on social network structure. During this time, methods that involved direct visualization of the social network became de-emphasized and the pixel oriented display became central.

One of the desired properties of the tool was to be able to visualize in a coherent manner as many of the attributes of the components simultaneously. As a result, we decided to use pixel oriented display techniques [14], [16] for this data. Many of the more technical-oriented members of the company found these displays intuitive, including some members of the business side of the company and one of the CEOs. Sales, however, found that the pixel oriented approach may be too difficult for new customers to understand. However, the member of the sales staff found that the tool could be used to isolate areas of the graph which would be then re-factored into bar charts for more intuitive presentation to customers.

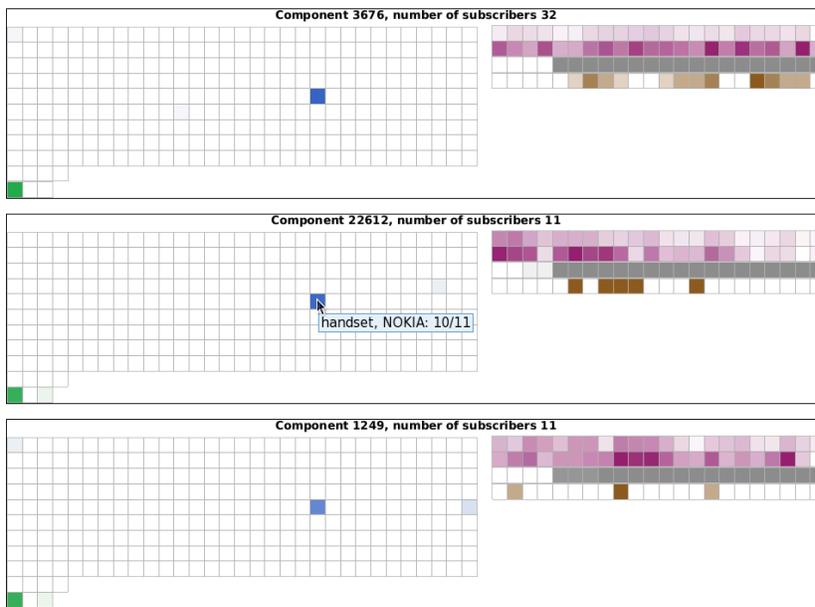
We moved away from showing the graph structure of the social network directly for scalability and privacy reasons. Thus, we focused on the churn component ids that are generated by the ChurnVis system and made them available in the interface. These ids can then be used to retrieve the structure

of the graph behind the components. Standard graph drawing tools [4], [5], [9] can be used as these components are usually no more than a few hundred nodes.

VI. CASE STUDIES AND USE BY END USERS

In order to test the design of our visualization with our users, we processed two mobile telecommunication data sets using the above-described pipeline. Initially, we presented some interesting features found by us, the designers, during meetings with the CEO and two analysts. ChurnVis was then installed locally on company machines and the analysts were able to further investigate the data over the course of a week without the designers of the system present. The two data sets, which we call *Location* and *Topup*, are described below.

Location was derived from CDR data collected from a large mobile telecommunications provider over the course of April 2011. The attributes associated with the nodes of the graph include an anonymized geolocation (name of actual city replaced a different city name) churn values, and the number of calls within and exterior to the component. The original social network contained 839,955,502 edges, reduced to 190,733,854 edges after filtering out weak edges and high degree nodes. In total, 114,322 churn components were found in the graph. During the visualization phase, all components less than two nodes were filtered out, leaving a total of 1347 churn components that were clustered by trends in their attribute values. To convert this graph to binary format took on the order of days. Summary histograms took on the order of several hours. Clustering the remaining churn components by attribute values took about thirty seconds on a laptop.



(a) Details View

Figure 3. Pixel oriented display for encoding the details of a cluster of components with similar attribute values. Each churn component is on its separate line in its own pixel oriented display. As the cluster of components was determined on handset (blue) and churn (grey) we notice a high similarity between component behaviours. The labels of each representative includes the number of subscribers in each component and the id number of the churn component that it represents. This id number can be used to retrieve the subscriber ids involved in the component and its graph structure locally on the servers of our collaborators.

Topup was derived from CDR data collected over the course of about five months from January through May 2012. The attributes associated with this graph include handset, method of payment, number of calls within and exterior to the component, churn, and topup information. The original social network contained 48,692,028 edges, reduced to 13,729,574 edges after filtering out weak edges and high degree nodes. In total, 165,952 churn components were found in the graph. During the visualization phase, all churn components of size four or less were filtered out leaving a total of 1,202 churn components that were clustered by trends in their attribute values. To convert this graph to binary format took about five hours. Summary histograms took on the order of an hour. Clustering the remaining churn components by their attribute values took about thirty seconds on a laptop.

A. Location

For this data set, our findings were made with the analysts during meetings and not independently, giving them some experience using ChurnVis. As we, the designers of ChurnVis, were present during these findings, we explain them but do not show screen shots for space reasons.

Immediately, it was apparent that good portions of the location information is unknown. This fact stood out through many of the saturated boxes on the far left of the display. Secondly, call activity drops with increased amounts of churn occur. This effect is also unsurprising as with increased churn calling activity should drop off as more subscribers choose to leave the network. These two behaviours were expected to be found in the data. Finally, no subscriber churns before midway through the month (April 14th). When the analysts were able to see this fact through the visualization, they believe that it was due to the way that this data set in particular was collected.

ChurnVis was then used to identify trends in churn in the context of anonymized regions by clustering on location. Once again, the feature that many of the locations of the subscribers is unknown is revealed by the visualization. However, filtering out only those churn components containing only subscribers from two large cities in the data reveals a number of trends. In both cases, it appears that churners tend to churn very late in the month.

By clustering on call activity and churn, we notice a strange anomaly. There are a few situations where call activity is high when a number of subscribers had already churned. This usually happened when the majority of the churn happens on the last day for the component and warrants further investigation.

B. Topup

For this data set, all of the findings reported below were made by one of the analysts, while he used ChurnVis over the course of a week without the designers of the system present. The analyst tried to explain his findings with the tool as described below and the provided all of the figures presented in this section.

While using the tool, one of our analysts noticed an anomaly with respect to churn and topups (Figure 4). The analyst noticed a cluster of churn components where nearly all of the subscribers in the cluster had churned but many of them were still topping up their mobile phones. He suggested that probable causes for this strange behaviour could be due to the churn flag associated with the algorithms used by the operator to predict churn. In effect, the churn flag is being set prematurely when the operator should wait for a longer period of inactivity before flagging the subscriber as churned.

This analyst also noted that there was a high correlation between number of topups and high call activity. This confirms something that would be expected of any mobile telecommunications data set: the more a subscriber calls the more that they would need to topup. Similarly, as components become saturated with churn, the number of calls made within the components falls off. This correlation is also expected, but the analyst believes that the correlation could be slightly weaker due to the churn flag problem described above.

Our analyst found that the subscribers that use a particular type of *Nokia* handset tend to use a *hybrid* payment plan with a combined prepay/postpay option (Figure 5). He thinks that this particular phone could be widely available on this particular price plan. Although the tool does not answer this question directly, it opens it up for further investigation. An interesting corollary is that there doesn't seem to be such a trend for prepaid users as no particular handset, or mobile telephone make, dominates this market.

The analyst found that when clustering by payment method a high proportion of customers that are postpay use *Research in Motion* handsets. He believes that these users are mostly business users as it is known that many companies provide its employees with this particular handset.

C. Qualitative Feedback

Qualitatively, both the CEO and analysts enjoyed using the tool. They believe the tool has potential and is able to easily illustrate changes in subscriber activities over time. The analysts particularly found the pixel oriented display useful as it was able to display large amounts of data in a succinct way.

Members of the sales part of the organization found the tool a bit complicated for presentation to customers. Our user in sales suggested that ChurnVis could be used by analysts to find bits of information of interest to their customers and create custom bar charts and pie charts for presentation.

VII. CONCLUSION AND FUTURE WORK

In this paper, we presented ChurnVis, a visualization tool for displaying the evolution of mobile telecommunications churn and subscriber actions over time. The visualization illustrates the evolution of churn components in the context of static and dynamic subscriber attributes. It does so in a way that the privacy of subscriber data and social network structure is protected. As the intended user community of this tool is diverse, our visualizations needed to be simple and easy to understand. Our users, employees of a mobile telecommunications company, were able to use the tool to find expected trends and anomalies in social networks of hundreds of millions of edges derived from CDR data.

In future work, we would like to consider other definitions for churn component. More specifically, churn components defined by community finding algorithms would be interesting. Visualization of components found in the graph in this way would help verify the consistency of communities and is very much of interest to our industrial collaborators.

ChurnVis is not designed to help with the prediction of churn. Rather, it provides a way of visualizing how churn progresses through a large graph in the context of subscriber

attributes. In future work, it may be interesting to see if visualization could help in finding ways of predicting churn, as social influence seems to be a factor in mobile telecommunications churn.

ACKNOWLEDGMENTS

The authors would like to acknowledge Ildiro Technologies and the support of the Clique Strategic Research Cluster funded by Science Foundation Ireland (SFI) Grant No. 08/SRC/I1407.

REFERENCES

- [1] J. Abello, F. van Ham, and N. Krishnan. ASK-GraphView: A large scale graph visualization system. *IEEE Trans. on Visualization and Computer Graphics (Proc. Vis/InfoVis '06)*, 12(5):669–676, 2006.
- [2] D. Archambault, T. Munzner, and D. Auber. GrouseFlocks: Steerable exploration of graph hierarchy space. *IEEE Trans. on Visualization and Computer Graphics*, 14(4):900–913, 2008.
- [3] D. Archambault, T. Munzner, and D. Auber. Tugging graphs faster: Efficiently modifying path-preserving hierarchies for browsing paths. *IEEE Trans. on Visualization and Computer Graphics*, 17(3):276–289, 2011.
- [4] D. Auber. Tulip : A huge graph visualization framework. In P. Mutzel and M. Jünger, editors, *Graph Drawing Software*, Mathematics and Visualization, pages 105–126. Springer-Verlag, 2003.
- [5] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An open source software for exploring and manipulating networks. In *Proc. of 3rd AAAI ICWSM Conference*, pages 361–362, 2009.
- [6] V. Batagelj, F. J. Brandenburg, W. Didimo, G. Liotta, P. Palladino, and M. Patrignani. Visual analysis of large graphs using (x,y)-clustering and hybrid visualizations. *IEEE Trans. on Visualization and Computer Graphics*, 17(11):1587–1598, 2011.
- [7] U. Brandes, J. Lerner, M. Lubbers, C. McCarty, and J. Molina. Visual statistics for collections of clustered graphs. In *Proc. of the IEEE Pacific Visualization Symposium (PacificVis'08)*, pages 47–54, 2008.
- [8] U. Brandes, J. Lerner, U. Nagel, and B. Nick. Structural trends in network ensembles. In *Proceedings of the 1st International Workshop on Complex Networks (CompleNet'09)*, pages 83–97. Springer, 2009.
- [9] U. Brandes and D. Wagner. Visone - analysis and visualization of social networks. In *Graph Drawing Software*, pages 321–340, 2003.
- [10] K. Dasgupta, R. Singh, B. Viswanathan, D. Chakraborty, S. Mukherjee, and A. A. Nanavati. Social ties and their relevance to churn in mobile telecom networks. In *Proc. of the 11th International Conference on Extending Database Technology: Advances in Database technology*, pages 668–677, 2008.
- [11] N. Elmqvist and J.-D. Fekete. Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *IEEE Trans. on Visualization and Computer Graphics*, 16(3):439–454, 2010.
- [12] M. Harrigan, D. Archambault, P. Cunningham, and N. Hurley. Egonav: Exploring networks through egocentric spatializations. In *Proc. of the ACM International Working Conference on Advanced Visual Interfaces (AVI '12)*, pages 563–570, 2012.
- [13] I. Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Trans. on Visualization and Computer Graphics*, 6(1):24–43, 2000.
- [14] D. A. Keim. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Trans. on Visualization and Computer Graphics*, 6(1):59–78, 2000.
- [15] D. A. Keim. Information visualization and visual data mining. *IEEE Trans. on Visualization and Computer Graphics*, 8(1):1–8, 2002.
- [16] D. A. Keim, M. C. Hao, U. Dayal, and M. Hsu. Pixel bar charts: A visualization technique for very large multi-attribute data sets. *Information Visualization*, 2(1):20–34, 2002.
- [17] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80:056117, 2009.

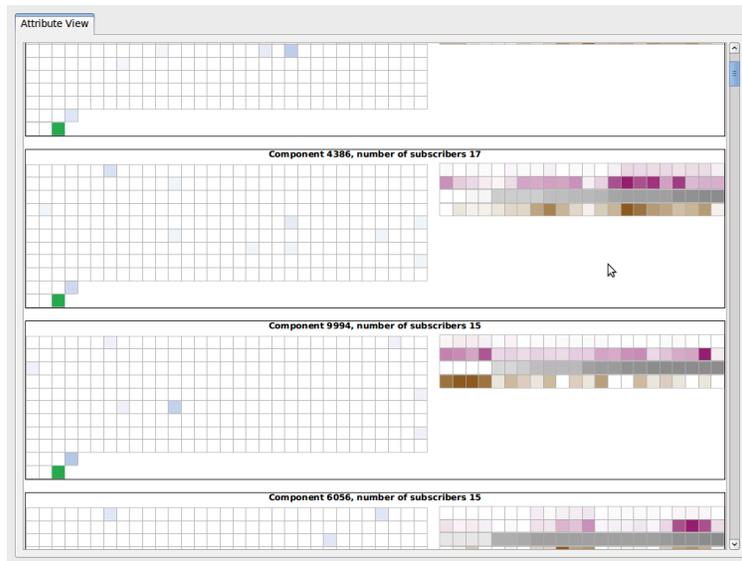


Figure 4. Anomaly found by one of our analysts. This screen shot was taken by the analyst during data exploration. In this screen shot of the details view, grey is churn and tan is topup. Time progresses from left to right in weekly intervals. In component 4386, underneath the cursor, notice a sharp spike of topups (saturated tan) when churn is high (saturated grey). The analysts believes that this could be due to a churn flag that is set too early after a period of subscriber inactivity.

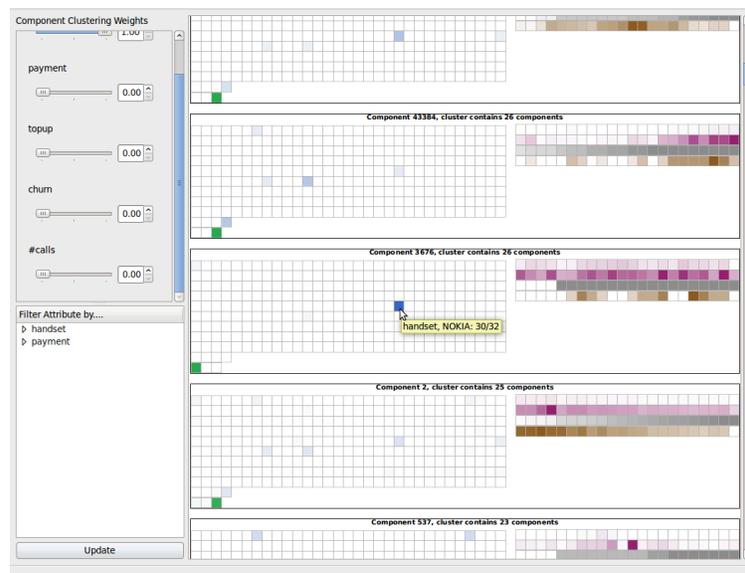


Figure 5. Correlation between *Nokia* subscribers and hybrid payment plans. This screen shot was taken by the analyst during data exploration. In the pixel oriented display, churn is grey and payment plan is green. Notice that in the cluster centroid below the cursor, there is a concentration of *Nokia* phones (Component 3676). The saturated green box in the bottom left indicates that these phones are mostly on hybrid plans. It could be the case that this mobile telephone is mostly available on a hybrid price plan. Hybrid price plans are generally less common than prepaid and postpaid plans.

- [18] M. Lubbers, J. Molina, J. Lerner, U. Brandes, J. Ávila, and C. McCarty. Longitudinal analysis of personal networks: The case of argentinean migrants in spain. *Social Networks*, 32(1):91–104, 2010.
- [19] A. McDaid and N. Hurley. Detecting highly overlapping communities with model-based overlapping seed expansion. In *International Conference on Advances in Social Networks Analysis and Mining (ASONAM'10)*, pages 112–119, 2010.
- [20] D. Oelke, M. C. Hao, C. Rohrdantz, D. A. Keim, U. Dayal, L.-E. Haug, and H. Janetzko. Visual opinion analysis of customer feedback data. In *Proc. of the 2009 IEEE Symp. on Visual Analytics Science and Technology (VAST '09)*, pages 187–194, 2009.
- [21] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Trans. on Visualization and Computer Graphics (InfoVis 2012)*, 18(12), 2012.
- [22] T. von Landesberger, M. Gerner, and T. Schreck. Visual analysis of graphs with multiple connected components. In *IEEE Symp. on Visual Analytics Science and Technology (VAST 2009)*, pages 155–162, 2009.
- [23] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner. Visual analysis of large graphs. *Computer Graphics Forum*, 30(6):1719–1749, 2011.
- [24] C.-P. Wei and I.-T. Chiu. Turning telecommunications call details to churn prediction: A data mining approach. *Expert Systems with Applications*, 23(2):103–112, 2002.