Noname manuscript No.
(will be inserted by the editor)

Visualization of Trends in Subscriber Attributes of Communities on Mobile Telecommunications Networks

Daniel Archambault · **Neil Hurley**

Received: December 15th, 2013 / Accepted:

Abstract Churn, the decision for a subscriber to leave a provider, is frequently of interest in the telecommunications industry. Previous research provides evidence that social influence can be a factor in mobile telecommunications churn. In our work, presented at ASONAM, we presented a system, called ChurnVis, to visualize the evolution of mobile telecommunications churn and subscriber actions over time. First, we infer a social network from call detail records. Then, we compute components based on an overlay of this social network and churn activity. We compute summaries of the attributes associated with the subscribers and finally, we visualize the components in a privacy preserving way. The system is able to present summaries of thousands of churn components in graphs of hundreds of millions of edges. One of the drawbacks of the original approach was that churn components were sometimes very large, leading to over-aggregation in the summary data. In this extension of the ASONAM paper, we adapt the ChurnVis approach to operate on the output of a community finding algorithm and present new results based on this adaptation.

Keywords Telecommunications Churn · Attributed Graphs · Graph Visualization · Social Networks · Community Finding Visualization

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

D. Archambault
Swansea University, Faraday Tower, Singleton Park, SA2 8PP, United Kingdom
Tel.: +44(0)1792 295371
Fax: +44(0)1792 295708
E-mail: d.w.archambault@swansea.ac.uk

N. Hurley
University College Dublin, School of Computer Science & Informatics, Belfield, Dublin 4, Ireland
neil.hurley@ucd.ie

1 Introduction

The ability to predict and analyse subscriber churn is important to the mobile telecommunications industry. When a customer **churns**, they decide to leave a particular service provider. There is evidence [40, 11] that social influence is a factor in the propagation of churn. Considering the social network of subscribers defined by their calling behaviour, the idea is that friends of subscribers who churn are more likely themselves to churn. However, social influence is not be the only factor involved in churn. There are many extrinsic and intrinsic factors that can influence a decision to leave a provider. Perhaps, for example, a different operator has a better deal for a particular handset which has become popular; or perhaps the subscriber has reached the end of a contract. In addition to the call detail records (CDRs) of the phone calls over their network, service providers maintain records associated with each subscriber, including **static** attributes that do not evolve over time, such as gender or birth date and **dynamic** attributes such as call activity or adoption of a particular model of a phone. Such records can be used to identify patterns in churn behaviour.

An up-to-date knowledge of the factors influencing churn, both social and non-social, allows providers to take steps to discourage their most valuable customers from churning. There is a business advantage to be gained in being able to quickly identify these patterns. The ChurnVis system presented in this paper is a visualization system focused on helping data analysts identify such patterns. Working with industrial collaborators, we developed a system that could identify the extent of social churn in the network and also present static and dynamic attributes of the churners in a way that would allow trends in these attributes to be discerned.

Analyst expertise and data set size are two significant challenges to this visualization problem. CDRs may be represented as a graph, where the nodes of the graph are the subscribers and the edges represent a phone call between a pair of subscribers. Telecommunications analytics graphs are very large with data set sizes of four million nodes considered as small. In this work, we have considered graphs of close to a billion edges. Secondly, ChurnVis needed to be developed to cater to analysts with significant technical expertise, as well as to CEOs, sales staff, and clients who may not have such expertise. With this wide range of visualization literacy, the system must be simple and must be usable for both exploratory and explanatory contexts.

In our previous work [2], we developed a system using the design study methodology [37] that was able to visualize churn in the context of a social network inferred from CDRs. In a design study approach, a visualization approach is developed via close collaboration with real users in a domain and is validated through evidence that the system does solve the intended target problem and is useful to experts. Design studies can have three primary contributions and in this paper we touch on all three.

In the first stage, *problem characterization and abstraction* defines the appropriate domain problem, abstracts it into tasks, and establishes the requirements against which the solution is judged. We have presented aspects of our problem characterization in section 3. In the second stage, a *validated visualization design* is presented, namely a tool that is the outcome of this project. The tool is validated with evidence that it is useful to experts and solves the intended target problem. The tool should be deployed to the intended users without the researchers present and evidence is collected that the end design actually helps the intended user community. Our tool is described in section 4 with evidence of rapid prototyping and iterative refinement presented in section 5. The tool is deployed in an environment with real users and we present screenshots of findings produced by one of the company engineers in section 6.2. In the third and final phase, *reflection* on the design study is presented in comparison to previous solutions which is present in many sections of the paper. We have made contributions to all three of these areas and have considered many of the pitfalls [37] when developing our design study.

In this paper, we extend ChurnVis so that it is able to handle the output of overlapping community finding algorithms. One of the drawbacks of our original ChurnVis system is that the components were sometimes quite large, consisting of hundreds of thousands of subscribers. Attribute aggregation over such large components can cause interesting low-level detail to be lost. Thus, it would be interesting to break these components down into smaller more intelligible *communities*. In this extended version of our paper, we apply community finding approaches to these large components and adapt our ChurnVis system to visualize the results of this process.

In this paper, sections 3 through 6 appear mostly as they did in the original ASONAM conference paper [2]. We have extended the discussion of related work (section 2) to include other relevant papers. This article further contributes to the literature by adapting ChurnVis to handle the output of an overlapping community finding approach [29] as presented in section 7.

2 Related Work

Related work in this paper is associated with several areas of visualization and some work in the area of churn prediction. In section 2.1, we present previous work in graph visualization. Section 2.2 looks at previous work in pixel oriented displays. Finally, section 2.3 briefly describes some work in mobile telecommunications churn prediction which has influenced the design of our system.

2.1 Graph Visualization Systems

The field of graph visualization has a long history with many techniques for understanding the structure of graphs [18, 14, 27]. 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, 3, 4, 7]. Scalable dynamic graph visualization approaches have also been developed based on grouping nodes of a large graph together [35]. 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.

Some approaches focus on the visualization of a graph via its static and dynamic attributes with considerations for graph structure. Semantic substrates [38] consider dynamic graph structures and their attribute values over time. In this work, graph nodes are plotted on the x-axis and node attribute values are plotted on the y-axis to understand how the graph and its attributes evolve over time. NetVisia [15] provides a large scale method for visualizing a graph and its dynamic attributes. The approach uses a heat map visualization of the evolution of node or edge attributes over time. The approach of von Landesberger *et al.* [25], provides a way for visualizing cascade information over time in a single integrated visualization. A cascade is a phenomenon that propagates from node to node via the edges of the network. The approach places nodes spatially according to when they participate in the cascade allowing for visualization of the process. These approaches are good methods for visualizing attribute information and graph structure, but they primarily focus on individual nodes and edges in the graph. The focus of ChurnVis is on components or communities located within the graph structure.

Our work, in many ways, is closely related to work on grouping subgraphs via structural similarity. Brandes *et al.* [9] 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.* [17] 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.* [8] 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 [28]. 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.* [26]. 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.

2.2 Pixel Oriented Displays

Our visualization technique heavily relies on pixel oriented displays [19,20] 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.* [21]. 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.* [21]. By using this technique, we have the advantage that network privacy can be maintained and the visualization can focus on the attribute values associated with each component.

2.3 Churn and Mobile Telecommunications Analytics

Much of the previous research applied to mobile telecommunications data has tackled two problems, namely churn prediction [40,11,13,16,12] and community finding or detection of social relationships in the subscriber network [31, 36,30,39]. The use cases provided in this paper focus mainly on the issue of understanding patterns that influence churn detection, although the system could be used for other data exploration tasks.

Many machine learning and statistical approaches to churn prediction have been investigated over the last number of years; an overview is provided in [22]. In summary, many of the standard classification techniques such as logistic regression, decision trees, and support vector machines have been applied to predict churn using features based on demographic and billing information. Other work, notably [40], has incorporated features based on CDR data into the churn prediction model, such as features pertaining to a subscriber's call patterns gathered over a fixed time period, namely, minutes of use, frequency of use and sphere of influence. While this work posits that a customer's phone usage can be a predictor of likelihood to churn, it does not explicitly model churn through *social influence*. One of the basic assumptions of our work is that social influence is a factor in mobile communications churn. 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.

Research on the prediction of churn through social influence is less extensive than churn using demographic data. In [11,12] a *spreading activation* method for churn prediction on telecommunication networks is studied. A churner's influence on his social network, is explicitly modelled as the spreading of a 'churn activation' from the churners to their neighbours through the call network. A churn prediction is formed by spreading an activation energy from the initial set of known churners to the rest of the network through their neighbours. A different approach is taken in [34], where a method for churn prediction in mobile networks is proposed that aims to identify closely-knit groups of subscribers and to analyse these groups to identify social leaders.

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 provide a natural segmentation of the graph around which we visualize the attribute space. Our visualization system assumes that churn operates based on social influence and visualizes this churn along with the attribute values associated with each component enriched with churn. In this way, we expect that it will be possible, not only to check the extent to which social churn occurs in the network under study, but also to discover which attributes are correlated with a tendency to churn.

3 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 a structural 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 [24]. Subscribers can have both static and dynamic attributes. Edges between subscribers can also have attributes and can be summarized per component.

Our users were interested in a system that would 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 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.*

3.1 Challenges of this Problem

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

3.1.1 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.

3.1.2 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.

3.1.3 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 subscribers, call record data, and the structure of the social network, or subsets of this information cannot be copied from the machines of our industrial collaborators, but 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.

4 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 Fig. 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 [26], and the details can be displayed by clicking on each representative.

4.1 Churn Components and Summarization

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

4.1.1 CDR Data and the Social Network

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 Fig. 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. Following previous social network analyses of mobile communications [11,23,33], we carry out some filtering of the data and retain only those nodes and links that are likely to correspond to true social connections between subscribers. In particular, we discard weak links on the basis that they correspond to incidental calls, rather than calls between friends. We also discard nodes of exceptionally high degree (nodes that place/receive a call on average once every ten minutes during several months of calling data), which may correspond to business call centres, rather than individual subscribers.

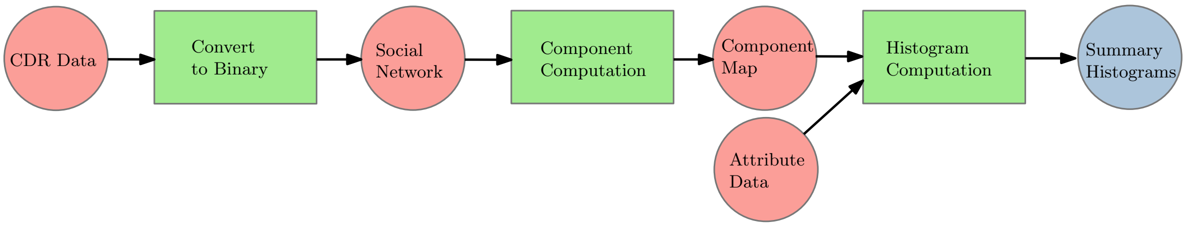
The precise rule to determine which weak links and nodes to discard can be somewhat arbitrary and tends to be guided by intuition. For example, in Dasgupta *et al.* [11] and Onnela *et al.* [33], only reciprocated links are retained. An even stronger criterion is used in Lambiotte *et al.* [23]: social links are identified as edges where at least six calls have taken place in each direction over the six months of the analysis, and it is argued that the analysis is not affected if the threshold is four or eight. In our analysis, we filter out links over which less than four communications have occurred over several months of data available to us. While the original call data is directed, following [33,23], we ignore the direction, resulting in an undirected **social network** where each edge is weighted by the number of calls made between the pair of subscribers. In ignoring direction, our contention is that the propensity to churn is affected by a subscriber's community of friends and that the volume of calls, rather than their direction, is more important for determining this community. These undirected graphs can be processed with a variety of clustering and learning approaches that do not operate on directed graphs. Nevertheless, as information about the direction of the call is still available through the CDR data, this information can still be leveraged by the system when computing summarization statistics.

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.

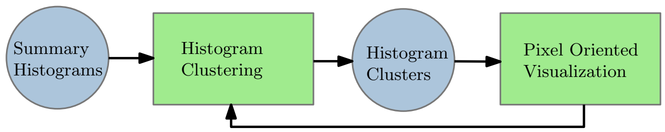
4.1.2 Churn Components

Once a clean social network has been curated and stored in binary format, this data is used to construct churn components. **Churn components** are the basic unit present in the visualization system. As the constructed social networks visualized by ChurnVis are large, direct visualization of all the nodes and edges individually would suffer from extensive visual clutter and would not be useful. Churn components group subsets of the nodes present in the social network into meaningful units for the visualization of macroscopic trends of the subscribers in the group while at the same time maintaining the privacy of the network structure.

A churn component is simply a set of nodes in the graph and can be derived in many different ways ways. Churn components could be defined using standard community finding algorithms [24], overlapping community finding algorithms [29], or other methods for generating interesting sets of nodes. For the results presented in section 6 churn components are induced subgraphs of churners on the social network computed through a breadth first search of the graph.



(a) Churn Components and Summarization



(b) Clustering and Visualization

Fig. 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.

Churn components defined in this way are easy to understand for our user community (Section 3.1.1).

Subsequently in section 7, we use overlapping community finding algorithms to define churn components for visualization. The MOSES [29] overlapping community finding algorithm is applied to break down large induced subgraphs into smaller components for a more fine-grained analysis. In overlapping community finding algorithms, nodes can belong to multiple communities. In order to accommodate this data, nodes that appear in multiple communities are duplicated and placed in the multiple churn components for summarization.

A key output of this stage of the algorithm is a **component map**. A component map lists, for each component, the identifiers of the subscribers that this component contains. This component map is used extensively when computing summary histograms as described in the next section.

4.1.3 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 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.

4.2 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 Fig. 1(b).

4.2.1 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 by our user community: clusters correspond to components with similar attributes. Other clustering algorithms could be substituted at this stage for analysis.

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. The representatives are visualized through pixel oriented displays as described in the next section.

4.2.2 Pixel Oriented Representation

Pixel oriented displays are a compact way of representing a large quantity of numeric data that can be ordered in some way. In the case of ChurnVis, we use this technique in order to represent a very large histogram as shown in Fig. 2. In the case of dynamic data the order is chronological, as in days of the calendar year from left to right. For static demographic data, it is ordered alphabetically, left to right, via attribute value. The number of subscribers in the churn component exhibiting a particular attribute is represented through saturation of the pixel. Highly saturated pixels indicate that almost all subscribers in this churn component have this particular value while pixels that are close to white in colour indicate that very few subscribers in this churn component have this value.

Fig. 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 Fig. 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.* [32] 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.

Fig. 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.

4.2.3 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.

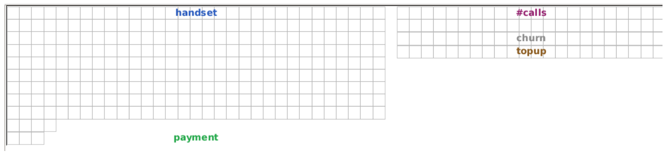
In order to support these usage scenarios, we provide the panel situated on the left hand side of the clustering view as shown in Fig. 2(b). In the top left, we have a number of sliders which control the weight given to each of the attributes in the clustering. In this case, we only consider the handset 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.

5 Iterative Refinement

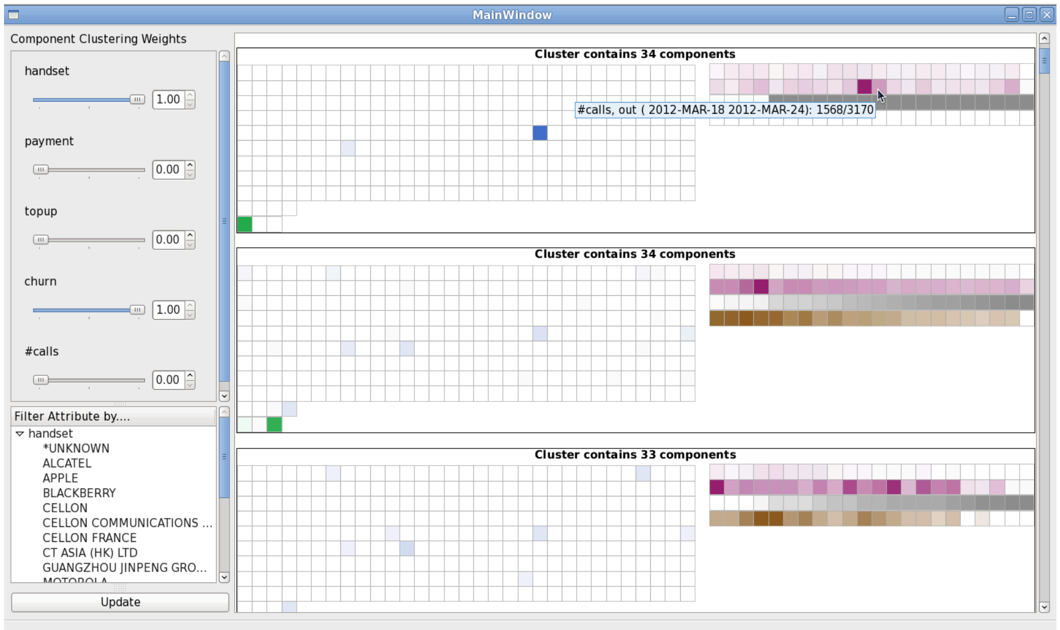
In this section, we describe our collaboration with a mobile telephone consulting company, leading to the design of ChurnVis. We began working with the group about in 2012 on this specific problem and developed various versions of the software to tackle the problem of visualizing churn and other attributes 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 [26]. 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 a churn component as it became clear that users would be interested in various types of structural groups as well as community finding methods that could be applied to the network.



(a) Legend for Display



(b) Component Clustering Interface

Fig. 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.

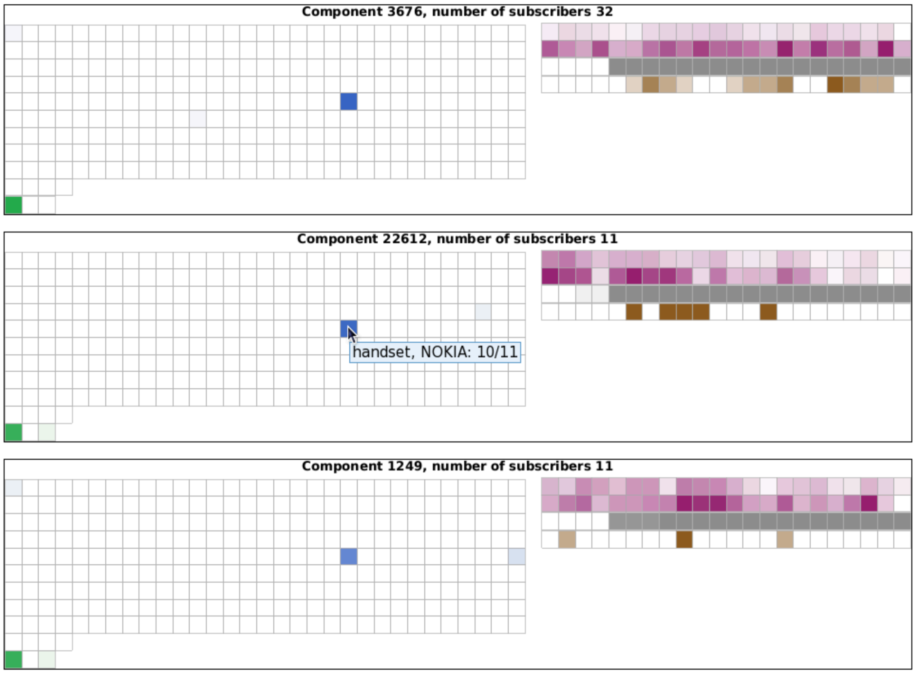
Also, during these meetings, we discovered that attribute data associated with the nodes and edges of the social network should be emphasized. This attribute data was often more intuitive and important. For example, understanding the concentration of iPhones in particular groups of nodes in the graph 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 [19,21] 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 [5,6,10] can be used as these components are usually no more than a few hundred nodes.

6 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



(a) Details View

Fig. 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.

without the designers of the system present. The two data sets, which we call `Location` and `Topup`, are described below. In all cases the visualization stage was executed on a 2.53 GHz laptop with 4GB of memory.

`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.

`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.

6.1 `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 do not show screenshots. When the tool was deployed on company machines, we show findings made by the analysts in section 6.2.

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.

6.2 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 (Fig. 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 top up. 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 (Fig. 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.

6.3 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. 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.

7 Analysis of Community Finding Approaches

In the original ChurnVis approach, churn components were the basic unit of analysis for the data. A churn component was defined as an induced subgraph of subscribers that churn. This definition facilitates analysis and provides a simplification of the graph based on churn in the network. However, this approach can lead to components that are quite large – sometimes in the range of hundreds of thousands of nodes. Although these churn components do form a meaningful group of nodes on the social network that are related by churn, they can often be too coarse obscuring detail.

A solution to this problem is to apply community finding approaches to these large components in order to break them down. These community finding approaches could be used to generate new churn components that can be visually analyzed using the ChurnVis system interface. We would just need to adapt our analysis and visualization pipelines in order to take into account of community structures instead of churn components.

In the following extension, we adapt the ChurnVis system to work with the output of overlapping community finding algorithms. More specifically, we apply the MOSES [29] approach to large churn components and subsequently apply ChurnVis to understand the trends in terms of dynamic and static attributes on the graph. Instead of comparing the

Fig. 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 top up. Time progresses from left to right in weekly intervals. In component 4386, notice a sharp spike of top ups (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.

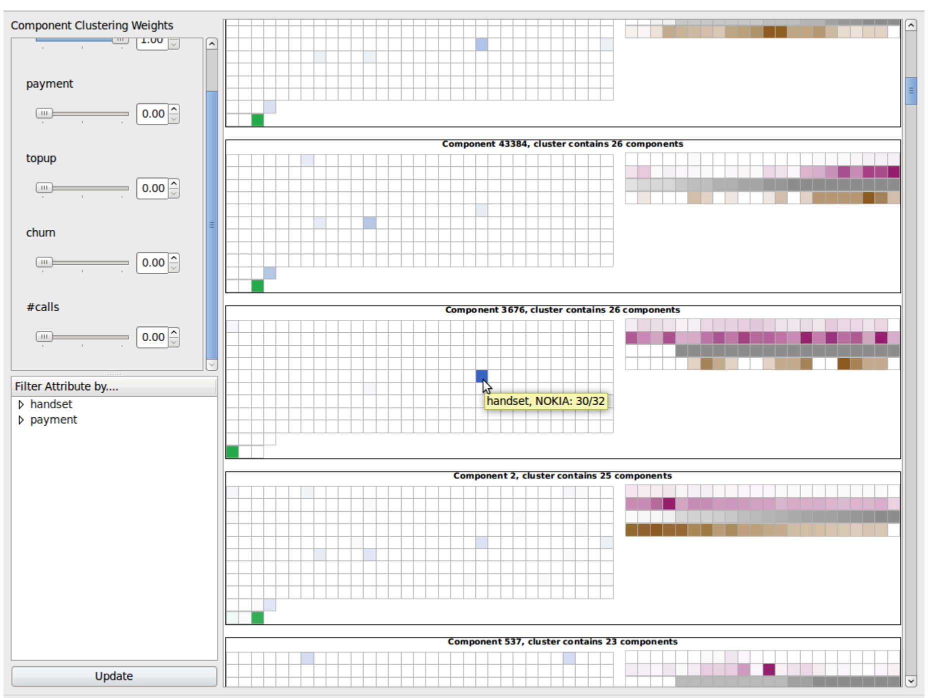


Fig. 5 Correlation between *Nokia* subscribers and hybrid payment plans. This screen shot was taken by the analyst during data exploration. In the pixel oriented display, churn is grey and payment plan is green. Notice that in the cluster centroid below the cursor, there is a concentration of *Nokia* 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 prepay and postpay plans.

summary histograms of churn components, we compare the summary histograms of overlapping communities.

In this section, we begin with a description of MOSES, the overlapping community finding algorithm that we use in this extension. Then, we present an outline of the new approach and describe the adjustments we had to make to the ChurnVis [2] pipelines in order to make them compatible with overlapping community finding algorithms. Subsequently, we present new results on the `Topup` data set. Finally, we discuss these results.

7.1 MOSES

MOSES is an overlapping community finding algorithm, proposed in [29]. It is based on a probabilistic generative model for the observed network. In this generative model, graph nodes may be assigned to multiple communities. Given a community assignment, an edge may exist between a pair of nodes if they share a community. In particular, if there are $m \geq 1$ communities in common between the two nodes, then each community has a chance to independently generate an edge between the nodes with a fixed probability p_{in}. The edge does not exist, only if each of the m trials fails. Furthermore, there is a small probability p_0 that an edge exists, regardless of community structure. Given this model, a likelihood for the observed network can be written down which depends on the parameters p_{in}, p_0 and the community assignment. A heuristic greedy optimisation algorithm finds the community assignment and parameters that maximise the likelihood. MOSES has been shown to be particularly effective on social network graphs with a high degree of overlap, that is, on graphs where nodes belong to two or more communities on average. Such highly overlapping structure has been shown to be prevalent in networks extracted from social media data, such as Facebook data. As these are typical friendship networks, such overlapping structure is likely to also occur in the calling circles of mobile subscribers.

MOSES has been evaluated and compared with the state-of-the-art overlapping community finding algorithms that exist in the literature. For further details of this evaluation, we refer the reader to Xie *et al.* [41].

7.2 ChurnVis and Overlapping Communities

Our approach to allow ChurnVis to summarize the results of overlapping community algorithms is as follows:

1. *Construct a social network and perform churn analysis*
2. *Select large churn components and apply MOSES [29]*
3. *Using the attribute data and overlapping community results, construct summary histograms.*
4. *Summary histograms can be visualized using the visualization pipeline*

The principal changes to allow ChurnVis to process the results of overlapping community finding algorithms is the introduction of steps (2) and (3) as specified above. The challenges of introducing these two steps are:

1. Scalable community detection with MOSES
2. Adapting the creation of summary histograms to process components where nodes can be in multiple components

The scalable community detection pass is executed on the social network in binary format. In order to perform this community detection, for each churn component we map its nodes back to the graph in binary format. The induced subgraph is taken and this subgraph is processed by MOSES. The resulting communities are then mapped back to a new component map with node ids for the contents of each community structure.

As MOSES is an overlapping community detection algorithm, a particular node id can be contained in multiple lines of the component map. As the creation of summary histograms processes each line of the component map separately, these new overlapping community maps can be processed as if they were component maps. The result of this process is a set of summary histograms for each overlapping community.

The final summary histograms can be visualized with ChurnVis unaltered. Thus, instead of grouping churn components by common attributes, we group overlapping community structures by common attributes. This grouping of overlapping communities allow us to see patterns in the dynamic and static attributes of these communities. Large churn components are broken down allowing for more detailed visual exploration of the phenomena within the community structures.

7.3 MOSES on `Topup`

The largest component of `Topup` is shown in Fig. 6. This very large component, 136,563 subscribers, exhibits the average behaviour of the data set: as churn increases both top ups and number of calls decrease. There is no handset that dominates the prepay market with prepay seeming to be the most popular payment plan for the data set. This result for such a large component is expected, but the analyst would probably want to drill down further into the data to see if there are smaller components with more localized behaviour.

We broke this component down using the above methodology and visualized the resulting communities with ChurnVis. MOSES produced 9,518 communities. We applied the visualization approach, filtering out all components with fewer than ten nodes, giving a total of 2,886 communities remaining. The k-means algorithm was applied with 100 centroids to classify the communities into groups with similar behaviour.

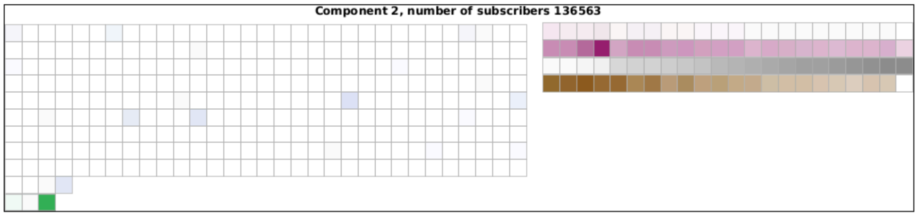


Fig. 6 The largest churn component in `Topup`. This component consists of 136,563 subscribers. Notice that number of top ups (tan) and number of calls (purple) steadily decrease as churn steadily increases. Also, the handset attribute (blue) is mostly desaturated. Payment plan (green) has a saturated element which is postpay. These attributes indicate an average behaviour over the large nodes which is expected: no handset dominates the market and the overall trend of volume and top up decreasing with increasing churn. Further decomposition of this component may yield better results.

For community analysis, we concentrate on all attributes except churn in this large component. As call volume is indicative of churn and some of the flags were misleading in the data set, we use a decrease in call volume to understand churn in the components of the data set.

Fig. 7 shows the classification of these communities by static and dynamic attribute value. At a high level, we immediately see that the attributes within these communities do not all vary in the same way. Thus, this decomposition of the large churn component into communities is beneficial to help illustrate the variation in both static and dynamic attributes within communities of this large churn component.

We click on a class of communities where the dominant handset is *Nokia* as shown in Fig. 8. This collection of communities seems to exhibit a common trend in the data set: call activity and top up activity steadily decrease together. This decrease occurs as members of these communities most likely churn. As seen in the previous study, the second component in this data set consists mostly of hybrid payment plans. This trend was noticed by our collaborators in the mobile telecommunications industry at a macroscopic level in this data set.

Using the interface, we filter out many collections of communities that do not have a large proportion of *Research in Motion* handsets in Fig. 9. When looking at the patterns of these clusters, we can see that many of these clusters have the same behaviour as the *Nokia* handsets mentioned above. However, the third family of clusters has an interesting pattern. For this family of communities, call volume and the number of top ups suddenly increase at the end of the time series where there was little activity before. This phenomena could be indicative of the early triggered churn flag; nodes would not have appeared in this data set if they did not churn, but we clearly see an increase in activity rather than a decrease in activity. It could be the case that the low activity earlier on in the time series caused churn flags to be triggered, but normal activity resumed later in the time series. Further investigation with telecommunications analysts would be required to confirm that this is the case.

8 Conclusion

In this paper, we presented ChurnVis as system for visualizing mobile telecommunications churn and subscriber actions over time. Our visualization process is simple to support a diverse community of members of the mobile telecommunications industry. ChurnVis was originally described in our ASONAM paper [2], but we have extended this visualization pipeline to process the results of overlapping community finding algorithms such as MOSES [29]. In the original ChurnVis system, churn components could become large, causing some components to be summarized a too high a level. This modified pipeline breaks down these large components by using a community finding algorithm. The results are still visualized in a privacy preserving way.

We have extended ChurnVis to work with the output of overlapping community finding algorithms. In future work, the system could be tried with a more diverse set of community finding approaches for comparison and applicability to the problem of visualizing subscriber attributes and churn over time.

Visualization methods for the problem of churn prediction continues to be an important area of future work. ChurnVis was not designed with the goal of predicting churn in mind; rather it is a visualization system that can be used to try and understand the factors influencing churn. Possible systems could exploit computational steering methods to reduce the time required in order to visualize the large volumes of data in this space.

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

References

1. Abello, J., van Ham, F., Krishnan, N.: 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. Archambault, D., Hurley, N., Tu, C.T.: ChurnVis: Visualizing mobile telecommunications churn on a social network with attributes.

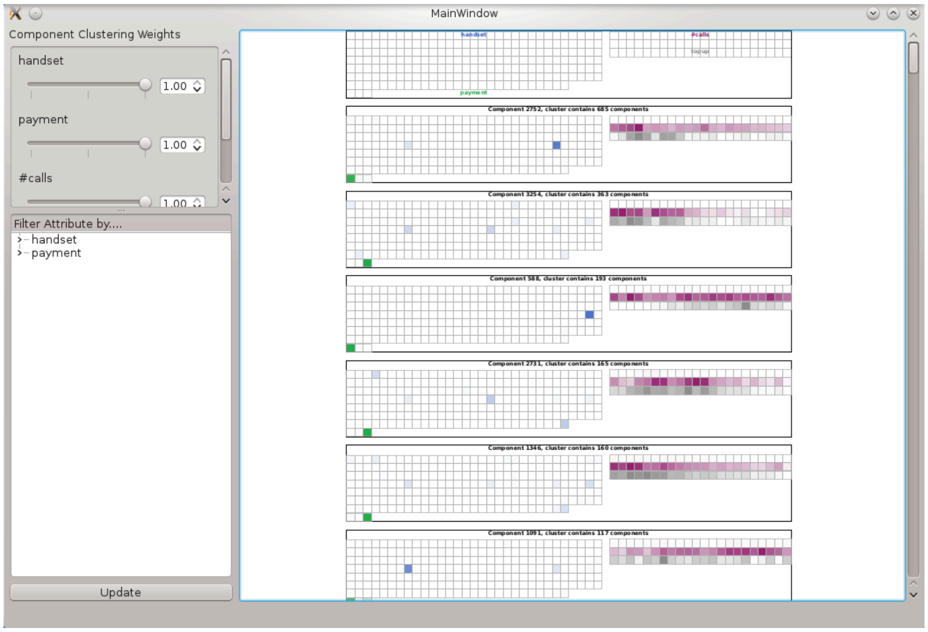


Fig. 7 Classification of the communities detected by MOSES [29] in the largest component of `Topup`. In this figure, the colour scheme is exactly the same as the rest of the paper except top up is grey and churn is not shown. Many different types of communities which vary in common ways are illustrated by breaking this large churn component down.

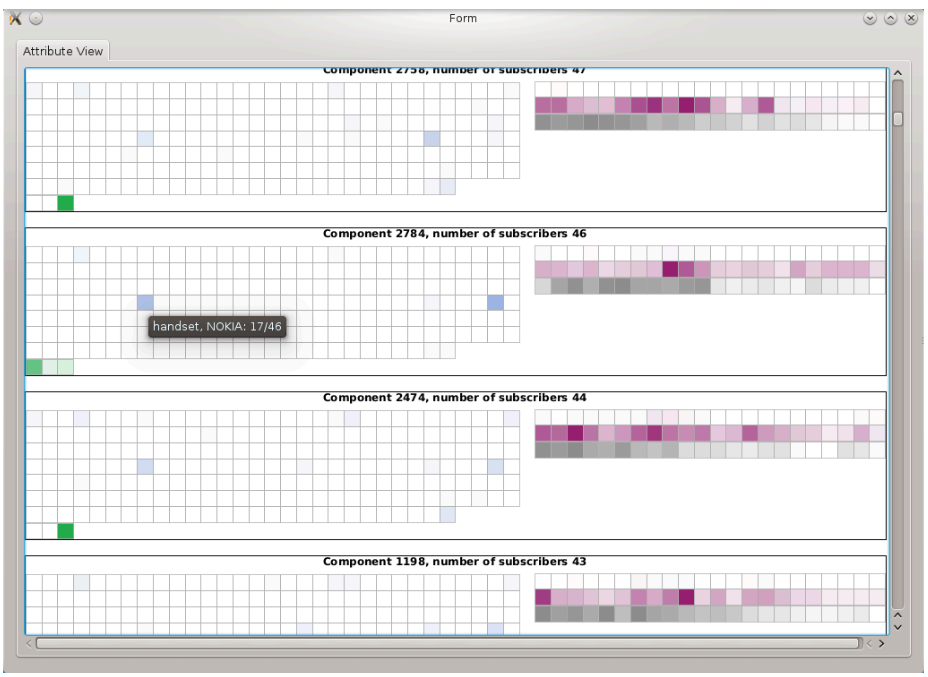


Fig. 8 The details of a collection of communities where a number of *Nokia* handsets are available. These communities exhibit a common trend: as call volume decreases so does the number of top ups. This phenomenon is indicative of churn activity and is common in `Topup`.

In: International Conference on Advances in Social Networks Analysis and Mining (ASONAM'13), pp. 894–901 (2013)

3. Archambault, D., Munzner, T., Auber, D.: GrouseFlocks: Steerable exploration of graph hierarchy space. IEEE Trans. on Visualization and Computer Graphics **14**(4), 900–913 (2008)
4. Archambault, D., Munzner, T., Auber, D.: Tugging graphs faster: Efficiently modifying path-preserving hierarchies for browsing paths. IEEE Trans. on Visualization and Computer Graphics **17**(3), 276–289 (2011)
5. Auber, D.: Tulip : A huge graph visualization framework. In: P. Mutzel, M. Jünger (eds.) Graph Drawing Software, Mathematics and Visualization, pp. 105–126. Springer-Verlag (2003)
6. Bastian, M., Heymann, S., Jacomy, M.: Gephi: An open source software for exploring and manipulating networks. In: Proc. of 3rd AAAI ICWSM Conference, pp. 361–362 (2009)
7. Batagelj, V., Brandenburg, F.J., Didimo, W., Liotta, G., Palladino, P., Patrignani, M.: Visual analysis of large graphs using (x,y)-clustering and hybrid visualizations. IEEE Trans. on Visualization

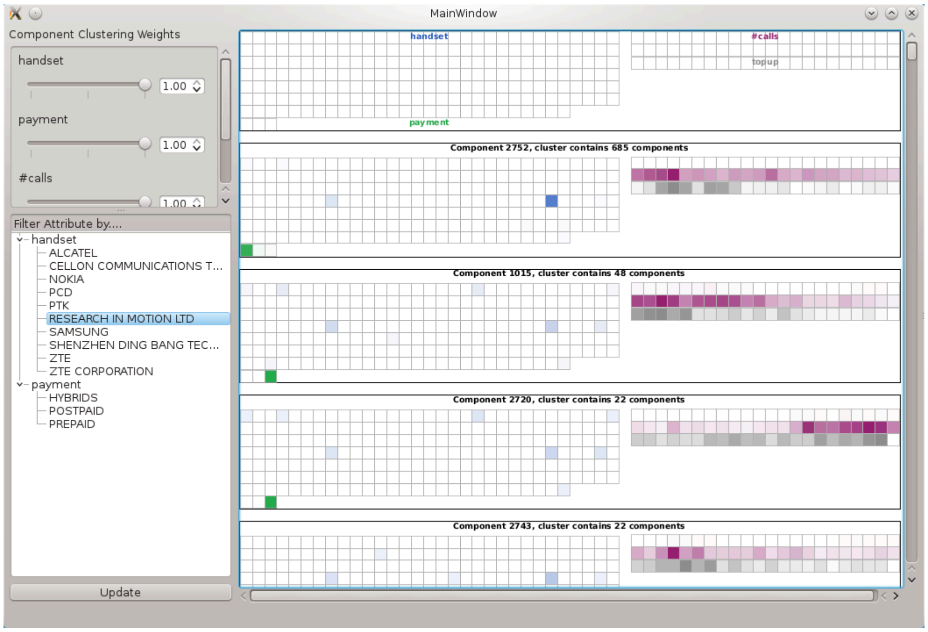


Fig. 9 Groups of communities that have a number of subscribers with *Research in Motion* handsets. A representative of these communities, clustered by attribute value is shown. The first two groups of communities exhibit typical behaviour: call volume falls off with number of top ups. The third group however (the one with 22 components) exhibits the opposite behaviour with both call volume and top ups increasing late in the time series. As all the nodes in these communities have been flagged as churning at some point, it could be the case that the churn flag was incorrectly set during the low activity in the component at the beginning of the time series.

and Computer Graphics **17**(11), 1587–1598 (2011)

8. Brandes, U., Lerner, J., Lubbers, M., McCarty, C., Molina, J.: Visual statistics for collections of clustered graphs. In: Proc. of the IEEE Pacific Visualization Symposium (PacificVis'08), pp. 47–54 (2008)
9. Brandes, U., Lerner, J., Nagel, U., Nick, B.: Structural trends in network ensembles. In: Proceedings of the 1st International Workshop on Complex Networks (CompleNet'09), pp. 83–97. Springer (2009)
10. Brandes, U., Wagner, D.: Visone - analysis and visualization of social networks. In: Graph Drawing Software, pp. 321–340 (2003)
11. Dasgupta, K., Singh, R., Viswanathan, B., Chakraborty, D., Mukherjea, S., Nanavati, A.A.: 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, pp. 668–677 (2008)
12. Droftina, U., Kosirb, A.: Customer churn prediction: Integration of sociometric theory of cliques into a diffusion model. In: Proceedings of Third International Conference on the Analysis of Mobile Phone Datasets (NetMob 2013) (2013)
13. Dyagilev, K., Mannor, S., Yom-Tov, E.: On information propagation in mobile call networks. Social Network Analysis and Mining **3**(3), 521–541 (2013)
14. Elmqvist, N., Fekete, J.D.: Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. IEEE Trans. on Visualization and Computer Graphics **16**(3), 439–454 (2010)
15. Gove, R., Gramsky, N., Kirby, R., Sefer, E., Sopan, A., Dunne, C., Shneiderman, B., Taieb-Maimon, M.: NetVisia: Heat map & matrix visualization of dynamic social network statistics & content. In: Proc. IEEE 3rd International Conference on Social Computing, pp. 19–26 (2011)
16. Han, Q., Ferreira, P.: Determinants of subscriber churn in wireless networks: The role of peer influence. In: Proc. of 26th European Conference on Operations Research (2013)
17. Harrigan, M., Archambault, D., Cunningham, P., Hurley, N.: Egonav: Exploring networks through egocentric spatializations. In: Proc. of the ACM International Working Conference on Advanced Visual Interfaces (AVI '12), pp. 563–570 (2012)
18. Herman, I., Melançon, G., Marshall, M.S.: Graph visualization and navigation in information visualization: A survey. IEEE Trans. on Visualization and Computer Graphics **6**(1), 24–43 (2000)
19. Keim, D.A.: Designing pixel-oriented visualization techniques: Theory and applications. IEEE Trans. on Visualization and Computer Graphics **6**(1), 59–78 (2000)
20. Keim, D.A.: Information visualization and visual data mining. IEEE Trans. on Visualization and Computer Graphics **8**(1), 1–8 (2002)
21. Keim, D.A., Hao, M.C., Dayal, U., Hsu, M.: Pixel bar charts: A visualization technique for very large multi-attribute data sets. Information Visualization **2**(1), 20–34 (2002)
22. KhakAbi, S., Gholamian, M., Namvar, M.: Data mining applications in customer churn management. In: Intelligent Systems, Modelling and Simulation (ISMS), 2010 International Conference on, pp. 220–225 (2010)
23. Lambiotte, R., Blondel, V.D., de Kerchove, C., Huens, E., Prieur, C., Smoreda, Z., Dooren, P.V.: Geographical dispersal of mobile communication networks. Physica A: Statistical Mechanics and its Applications **387**(21), 5317–5325 (2008)
24. Lancichinetti, A., Fortunato, S.: Community detection algorithms: A comparative analysis. Physical Review E **80**, 056,117 (2009)
25. von Landesberger, T., Diel, S., Bremm, S., Fellner, D.W.: Visual analysis of contagion in networks. Journal of Information Visualization (2013)
26. von Landesberger, T., Gorner, M., Schreck, T.: Visual analysis of graphs with multiple connected components. In: IEEE Symp. on Visual Analytics Science and Technology (VAST 2009), pp. 155–162 (2009)

27. von Landesberger, T., Kuijper, A., Schreck, T., Kohlhammer, J., van Wijk, J.J., Fekete, J.D., Fellner, D.W.: Visual analysis of large graphs. Computer Graphics Forum **30**(6), 1719–1749 (2011)
28. Lubbers, M., Molina, J., Lerner, J., Brandes, U., Ávila, J., McCarty, C.: Longitudinal analysis of personal networks: The case of argentinean migrants in spain. Social Networks **32**(1), 91–104 (2010)
29. McDaid, A., Hurley, N.: Detecting highly overlapping communities with model-based overlapping seed expansion. In: Proceedings of the 2010 International Conference on Advances in Social Networks Analysis and Mining, ASONAM '10, pp. 112–119 (2010)
30. Miritello, G., Moro, E., Lara, R., Martínez-López, R., Belchamber, J., Roberts, S.G., Dunbar, R.I.: Time as a limited resource: Communication strategy in mobile phone networks. Social Networks **35**(1), 89–95 (2013)
31. Muhammad, S.A., Laerhoven, K.V.: Quantitative analysis of community detection methods for longitudinal mobile data. International Conference on Social Intelligence and Technology (SOCIETY) pp. 47–56 (2013)
32. Oelke, D., Hao, M.C., Rohrdantz, C., Keim, D.A., Dayal, U., Haug, L.E., Janetzko, H.: Visual opinion analysis of customer feedback data. In: Proc. of the 2009 IEEE Symp. on Visual Analytics Science and Technology (VAST '09), pp. 187–194 (2009)
33. Onnela, J., Saramaki, J., Hyvonen, J., Szabo, G., Lazer, D., Kaski, K., Kertesz, J., Barabasi, A.L.: Structure and tie strengths in mobile communication networks. Proc. Natl. Acad. Sci. USA **104**(18), 7332–7336 (2007)
34. Richter, Y., Yom-Tov, E., Slonim, N.: Predicting customer churn in mobile networks through analysis of social groups. In: SDM, pp. 732–741 (2010)
35. Sallaberry, A., Muelder, C., Ma, K.L.: Clustering, visualizing, and navigating for large dynamic graphs. In: Proc. 20th Intl. Symp. on Graph Drawing (GD 2012), *LNCS*, vol. 7704, pp. 487–498 (2012)
36. Sarraute, C., Calderon, G.: The evolution of communities with focus on stability. In: Proceedings of Third International Conference on the Analysis of Mobile Phone Datasets (NetMob 2013) (2013)
37. Sedlmair, M., Meyer, M., Munzner, T.: Design study methodology: Reflections from the trenches and the stacks. IEEE Trans. on Visualization and Computer Graphics (InfoVis 2012) **18**(12), 2431–2440 (2012)
38. Shneiderman, B., Aris, A.: Network visualization by semantic substrates. IEEE Trans. on Visualization an Computer Graphics (Proc. Vis/InfoVis 2006) **12**(5), 733–740 (2006)
39. Walsh, F., Pozdnoukhov, A.: Spatial structure and dynamics of urban communities (2011). The First Workshop on Pervasive Urban Applications (PURBA)
40. Wei, C.P., Chiu, I.T.: Turning telecommunications call details to churn prediction: A data mining approach. Expert Systems with Applications **23**(2), 103–112 (2002)
41. Xie, J., Kelley, S., Szymanski, B.K.: Overlapping community detection in networks: The state-of-the-art and comparative study. ACM Computing Surveys **45**(4), 43:1–43:35 (2013)