AUDITING DATA STREAMS FOR CORRELATED GLITCHES

(Research Paper)

Ji Meng Loh & Tamraparni Dasu
AT&T Labs - Research
{loh,tamr}@research.att.com

Abstract: Cellular networks carry vast amounts of voice, text and data traffic every second. The networks are monitored constantly to measure network performance, detect traffic congestion, identify anomalies, and to serve other customer service and network support functions. The data collected from mobility networks is used to make many critical decisions. The quality of the information plays an important role in the effectiveness of these decisions. Therefore it is important to ensure that the data collected from cellular networks meets quality standards. In particular, identifying glitches that are correlated can help in identifying root causes and facilitate more efficient problem solving in the network as well as quicker data repairs.

In this paper, we present a methodology for automated auditing of massive, complex data streams with a focus on correlated glitches, and a case study that illustrates the application of this methodology. The methodology has two main components, a set of logical constraints that embody domain specific information, and statistical methods for identifying correlated glitches to enable automated quantitative cleaning of data. Together, the two components provide a comprehensive yet customizable set of criteria for evaluating information quality as a function of time and network topology. We demonstrate the use of the *cross g function* to identify correlations in glitches. In the case study, we focus on duplicate, missing, inconsistent and anomalous data, and correlations between glitches across time, space and topology.

Key Words: Data Quality, Correlated glitches, Automated detection

INTRODUCTION

The advent of new technologies that enable video chat, gaming, movies and other data-intensive applications on mobile devices such as cell phones and iPads, together with the ubiquity of these mobile devices caused by a radical shift in people's behavior in the consumption of entertainment and information, has led to a phenomenal growth in voice and data traffic in cellular networks. In order to provide reliable, high quality service to their customers, operators of cellular networks collect and maintain vast amounts of data to monitor their networks. This data is used for network management and optimization. Given the critical nature of the decisions that are supported by this data, it is important to ensure that the data quality is kept at a high level, with timely diagnosis and mitigation of data quality issues.

In this paper, we present a methodology for automatically auditing massive streams of network data, with an emphasis on identifying data glitches with spatial and temporal correlations. The methodology uses a combination of logical constraints based on specific network characteristics, and time series and spatial statistical methods. We use a combination of univariate and multivariate outlier detection methods. We measure the extent to which outlier sequences generated by different variables or different network components match. In addition, we also explore how the degree to which the outlier sequence match varies with time, space and network topology.

While we focus on mobile telecommunication data streams in our case study, the methodology is generally applicable. See Wang *et al* [10] for a general overview of maintaining, monitoring and measuring data quality in databases.

Challenges

Monitoring cellular network data streams is a challenging task for several reasons. (1) An extremely complex network with hundreds of thousands of components organized in a hierarchical structure; (2) massive numbers of data streams that accumulate at a rapid rate; (3) glitches, or data quality issues that exhibit intricate spatio-temporal dependence patterns due to complex underlying root causes. We describe these challenges in detail below.

(1) Cellular Network

A telecommunications network has a complicated hierarchical structure consisting of multiple layers with interconnected sub-networks. Each sub-network has numerous software and hardware components that are constantly being repaired, upgraded, removed or added. The network is a dynamic and evolving entity.

Figure 1(a) shows a highly simplified telecommunications network structure. The conventional Circuit Switched network (CS-Core Network), the Packet Switched network (PS–Core Network), and the Universal Terrestrial Radio Access Network (RAN) communicate with each other to transfer different types of telecommunications traffic – voice, data and internet traffic – from source to destination. These sub-networks are controlled by a signaling system (SS7) that routes traffic between conventional phone service (PSTN), the Internet (IP) and mobile phone networks (RAN). More detailed information is beyond the scope of this paper and the reader is asked to refer to Lin and Chlamtac [7].

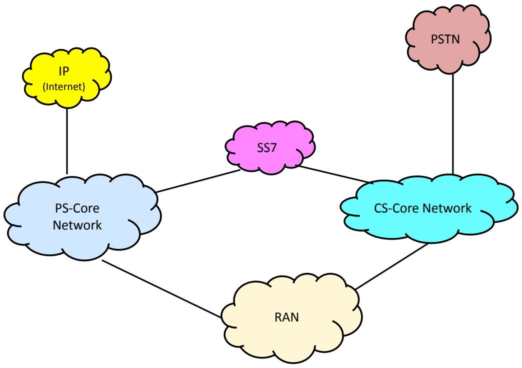

Figure 1 (a)

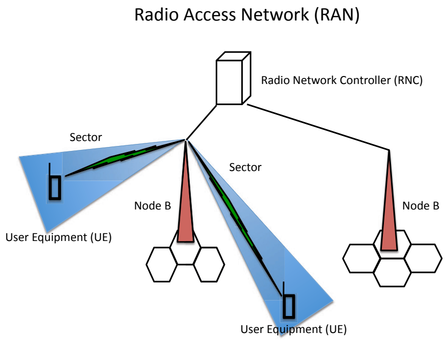

Figure 1 (b)

The data analyzed in the case study comes from the Radio Access Network (RAN), the components of which are shown in Figure 1(b). To convey the complex nature of our data, we briefly describe in a simplified manner the process that the RAN goes through to facilitate end-to-end completion of a mobile call. Note that terminology is specific to the technology underlying the RAN. We use 3G (broadband) network terminology here.

When a user initiates a call on a handset (labeled user equipment (UE) in Figure 1(b)), the signal is picked up by a directional antenna that covers the current location of the handset. This antenna is one of many, as many as nine, antennas mounted on a cell tower. A "Node B" associated with a cell tower passes the call signal onto its controller (RNC). The RNC manages signals from many Node Bs which are often located in the same geographical region. The RNC then contacts the appropriate sub-network, e.g. the CS-Core Network if the destination is a public switched telephone service (PTSN), or the PS-Core Network if the traffic is headed towards the Internet.

In order to properly address data quality issues, it is important to understand the processes that generate the data. Acquiring sufficient domain knowledge is necessary. A lack of accurate, complete and timely documentation increases the likelihood of misinterpretation of the data and of the incidence of glitches.

(2) Computing with Data Streams
A mobility network consists of hundred of thousands of components in a hierarchical structure. Counters measure various performance and traffic volume metrics on each of these components, with the data collected at very fine time intervals. Essentially, we have a set of *N* time series, where *N* is extremely large, each with a very high rate of data accumulation. It is not feasible to store or access the raw data. We need to summarize the data to make further analyses possible. The *choice of summarization and modeling techniques determines the usability of data*. For instance, aggregating data too coarsely across time and devices could mask data quality issues specific to time periods or devices.

(3) Complex Glitch Patterns
Given the hierarchical nature of the network elements and longitudinal propagation of errors, data glitches exhibit temporal and spatial correlations. Furthermore, different types of glitches have a tendency to co-occur. For example, missing values are often accompanied by outlying values of other attributes. A high load on the network might suppress pollers, resulting in a lagged or simultaneous co-occurrence of outliers and missing values.

Berti-Equille *et al* [2] propose exploiting dependence between glitches, and other glitch patterns to formulate effective data cleaning strategies. Berti-Equille *et al* [1] provide a comprehensive overview of advances in data quality mining, for using data mining techniques to identify, measure and treat complex data glitches in massive amounts of data.

TYPES OF GLITCHES

While the methodology is generally applicable, our case study uses cellular network data. We define here the specific glitches we focus on – duplicates, inconsistencies, missing values and anomalies.

Duplicates

We define duplicates as records that have the same supposedly unique identifier but different attribute values. In principle, there should be no duplicates in the data set. Data repair consists of eliminating the duplicates either by retaining a random record or fusing the duplicates into a single unique record. See Elmagarmid et al [4] for an overview of duplicate detection in databases.

Inconsistencies

Inconsistencies come in many flavors. Some can be inferred from the data description. For example, “duration should be non-negative”, or “if US ZIP code = 07932 then state=NJ”. Some constraints can be formulated based on heuristics or consultation with experts. For instance the purely fictional example, “if service=voice_only, then text_traffic=0”. Any record that violates this rule is deemed inconsistent. Data repair consists of imputing consistent values using either functional dependencies or other criteria. See Golab *et al* [5] for details.

Missing Data

When an attribute value is not populated, it is considered missing. Occasionally, a default value, such as

9999 or -10000 is used to denote missing values. Usually, these are easy to detect, unless a common value like 0 is used to denote missing values, or there is no standard representation resulting in multiple representations of missing values. A data browser can be used to identify such non-standard representations. See Dasu *et al* [3] for details. Data repair consists of imputing missing values or dropping records that have missing values. Different imputation techniques often result in different "clean" data sets and can lead to different results and conclusions.

Outliers & Anomalies

A significant portion of our paper is devoted to outlier detection. Outliers and anomalies are the most prevalent glitches, but hard to determine with certainty. Our approach entails using multiple methods to screen for outliers. Outlier detection through multiple methods presents some interesting theoretical questions that are described later in the paper.

OUR APPROACH

The methodology consists of two parts. (1) A set of constraints, either developed in consultation with experts or constructed from data properties and functional dependencies, that the data must satisfy, and (2) statistical spatial and temporal analyses on the output of anomaly detection techniques.

In our experience, an effective data quality auditing methodology must have both components. Data quality is highly domain and context dependent, and therefore it is important to incorporate domain knowledge gathered from experts into a set of rules or constraints that the data must satisfy. In addition, given the vast amounts of rapidly accumulating data, statistical anomaly detection techniques are essential to screen the data and identify smaller subsets of data for further analyses in order to identify patterns and co-occurrences that can then aid root cause identification and more effective data repair.

Statistical Methods

The statistical methods we use fall into two broad categories, anomaly detection methods and techniques for analyzing the spatial or temporal correlation of detected anomalies. We describe these methods in more detail below.

Univariate and Multivariate Outlier Detection Techniques

There are many outlier detection methods. Most are univariate, i.e. they detect outliers in each attribute separately. Common univariate methods include 3-σ limits of a Gaussian distribution and quantile-based methods such as the 5th and 95th percentiles of a distribution. See Kriegel *et al* [6] for a comprehensive overview of outlier detection methods. A common multivariate outlier detection method is the Hotelling's T^2 statistic. Details of this method can be found in Rao [8].

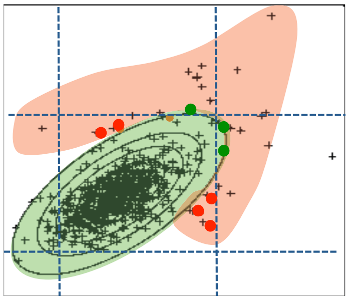

Figure 2*: Bivariate (green) and univariate (red) outliers

It is possible for univariate and multivariate outlier detection methods to identify different outliers. Such differences could be caused by fundamental conceptual differences in the detection methods rather than by random chance due to Type 1 (false positive) and Type 2 (false negative) error rates. Figure 2* (courtesy of Dr. Laure Berti-Equille) shows a bivariate Gaussian represented by the green ellipse and the outlying region of the joint distribution shaded in red. The dotted blue lines represent univariate bounds for outlier detection in the X or Y dimension. The bold green dots are univariate outliers that lie outside these dotted blue lines. These are however not multivariate outliers with respect to the bivariate Gaussian. On the other hand, the bold red dots are outliers based on the bivariate Gaussian but are not univariate outliers since they lie within the univariate bounds.

In general, multivariate outliers can be more informative because they take into account the dependence structure between the attributes, but they are more difficult to compute because finding outliers in multiple attributes simultaneously entails specifying and estimating a joint distribution.

In this paper, we do not develop new outlier detection techniques. Instead our methodology uses a combination of univariate and multivariate techniques to harness the strengths of each approach. Specifically, we report detections from the Hotelling's T^2 statistic, but use the detections from the 3-σ limits of the Gaussian distribution as an additional performance check.

Setting Thresholds

In a statistical hypothesis-testing framework, the level of significance, α, plays a crucial role as a threshold for determining whether a data point is an outlier. This quantity is also known as the Type I error and corresponds to the false positive rate under the null model. It is not straightforward to establish an equivalence between the significance levels for univariate and multivariate methods. If we fix the false positive rate to be 5% for each of the two attributes, what is the comparable level of significance for the joint distribution? This is an interesting theoretical question that needs further investigation. In this paper, we compare methods by choosing thresholds so that about the same number of outliers are detected by the individual methods.

Temporal and Spatial Correlations

Consider an outlier sequence as a sequence of points in time, say $x_1, x_2, ..., x_{N_x}$. With another outlier sequence $y_1, y_2, ..., y_{N_y}$, we are often interested in measuring the degree at which these outlier sequences match. The outlier sequences may be univariate outliers from different variables measured at the same network component, or multivariate outlier sequences measured at neighboring components. More specifically, we are interested in how much the sequences match temporally, and how the degree of match varies with spatial separation and network topology. For example, since a Node B manages multiple sectors in the same physical location, one might expect that technical issues at that location can manifest themselves in the data collected from these sectors as outliers that occur around the same time. If outlier sequences from nearby Node Bs match to a high degree, it may be indicative of more widespread network problems.

There are many ways to compare two outlier sequences. A comprehensive study is beyond the scope of this paper and we defer a detailed comparison to a later work. Here, we treat outlier sequences as one-dimensional point processes and focus on using the cross *g* function to compare pairs of outlier sequences. The cross *g* function is a variation of the usual *g* function (also called the pair correlation function) commonly used in the analysis of spatial point patterns. Given a lag *l*, the cross *g* function is estimated by counting the number of points separated by distance *l*,

$$\hat{g}(l) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} 1\{|x_i - y_j| \in (l - dl, l + dl)\} / (N_x N_y).$$

Thus, instead of a metric that yields a single distance between two outlier sequences, the cross *g* function is a function of lag *l* and measures the strength of the correlation between two outlier sequences at each lag *l*. It is related to the probability of finding a point pair separated by distance *l*. Large values of the cross *g* function at the very small lags, say, *l*=0 or 1, are indicative of a match between the two outlier sequences. Peaks at other values of *l* may suggest periodicity in the patterns or time lags between the sequences. See Stoyan *et al* [9] for details.

CASE STUDY: APPLICATION TO MOBILITY DATA

In this section, we present results from our case study of applying the techniques described above to network monitoring data collected by a United States telecommunications company. For proprietary reasons, we will describe the data in somewhat general terms, and present results at a high level on a limited number of attributes. The methodology however, can be applied in general at any level of granularity. All the data processing and statistical analysis is done using a combination of SAS routines and R code.

Motivation

Many important decisions related to the management and optimization of telecommunications network are based on monitoring data collected from switches that control voice, data and text traffic on the cellular network. Since such decisions cost the company millions of dollars in equipment and significantly impact customer experience, it is important that the data be of good quality. However, the sheer volume of the data makes it impossible to monitor it in any manual fashion. Furthermore, some glitches can only be detected at a fine granularity and can be lost when the data is aggregated to make it more manageable.

Data Description

The original data set consists of performance metrics of antennae on cell towers gathered from counters that are polled at regular time intervals. In this paper, we focus on three attributes that we denote USAGE, SAMPLES and PERFORMANCE, where USAGE refers to the load handled by the network element during the interval between consecutive polls, SAMPLES to the number of samples collected during the same interval, and PERFORMANCE to a performance metric measured over that same interval.

A typical data record contains the time stamp, the values of these three quantities and information about the network component and its hierarchy, i.e. the Sector ID, and the IDs of the Node B and RNC that it falls under. The components RNC, Node B and Sector were briefly described in the Introduction and illustrated in Figure 1. Thus the data consists of rows of the form

TIME|RNC|NODE B|SECTOR|USAGE|SAMPLES|PERFORMANCE

with more than 100 million such records collected over a period of several months. The data arrive as a sequence of flat files, each file containing multiple records. At any point in time, we have access to the most current collection of files along with the cumulative summaries that we maintain of all the data observed so far. The case study described below was carried out on the entire data, not in any experimental or simulation-based setup.

Data Quality Assessment

After initial data preprocessing, the first step in our data quality assessment methodology involves procedures to enforce compliance with domain specific constraints. We deal first with duplicates and missing records, whose constraints are relatively easy to define, then identify and address inconsistent or damaged values. After these fundamental issues are addressed, univariate and multivariate techniques are employed to detect outliers. Finally, we study correlations among different types of outliers and outlier sequences (next section).

Golab et al [5] describe the use of functional dependencies to formulate constraints for identifying inconsistencies and to isolate subsets that match or violate the constraints. In our case, there are many logical constraints particularly with respect to the network topology. For simplicity and to avoid having to describe the network in great detail, we list several of the simplest ones:

1. If two or more records have the same unique identifier, the remaining attributes should also be identical (duplicate records).
2. No attribute should be missing (missing values).
3. All possible occurrences of the unique identifier should be present (missing records).
4. Logical inconsistencies
   a. When not missing, attribute USAGE ≥ 0.
   b. When not missing, attribute PERFORMANCE should lie in the interval [C_1, C2], specified by experts.
   c. If attribute SAMPLES is missing, then attribute USAGE should be missing.

Table 1 summarizes the distribution of duplicates, missing data and inconsistencies across RNCs and the frequency at which they occur. We classify an RNC as affected at a given time if at least one sector of a Node B associated with the RNC has a data glitch. We next describe more detailed results from running the data stream through this set of constraints.

TABLE 1

		% of RNCs Affected	**% of TIME Affected**	**% RECORDS Affected**
Duplicates		3.6%	26%	0.05%
Missing Attributes	All	5%	12%	0.01%
	USAGE	22%	24%	0.08%
	SAMPLE	74%	43%	2%
	PERFORMANCE	99%	99%	8%
Inconsistencies	Negative Usage	1.5%	52.8%	0.006%
	Negative Performance	87%	99%	0.04%
	Usage Without Sample	15%	13%	1.7%

Duplicates

Under ideal conditions, there should be a single record for every combination of TIME, RNC, NODE B and SECTOR, the unique identifier for a record. We found that a little over 0.03% of the total records were duplicates with respect to the unique identifier (unique combination of TIME|RNC|NODE B|SECTOR). Further investigation showed that the duplicates were created under two circumstances.

In the first scenario, duplicates are generated when one of the fields in the unique identifier is overwritten by a common error code. For example, the two records

TIME1|RNC|NODE B|SECTOR|value1|value2|value3
TIME2| RNC|NODE B|SECTOR|value4|value5|value6

become mapped to

ERROR CODE| RNC|NODE B|SECTOR|value1|value2|value3
ERROR CODE| RNC|NODE B|SECTOR|value4|value5|value6

resulting in duplicates.

Data repair consists of using functional dependencies, interpolation or other domain knowledge to assign correct values, or to drop records that are duplicates AND contain an error code in any of the fields of the unique identifier.

In the second scenario, we found that records were re-transmitted when the system identifies them as damaged or incomplete. Thus, for example, the damaged record

TIME|RNC|NODEB|SECTOR|missing|inconsistent value|missing

may be re-transmitted (accurately) as

TIME|RNC|NODE B|SECTOR|correct value|correct value|correct value

resulting in two records with the same join key. Data repair involves the simple deletion of duplicate records that have missing or inconsistent attribute values.

Note that the co-occurrence of missing and inconsistent values along with duplicates is an example of multivariate, co-occurring glitches.

In Figure 3 we show the occurrence of duplicates across time and RNCs. The *X*-axis represents time while the *Y*-axis corresponds to RNCs organized approximately by hierarchy. Each row of dots correspond to duplicates occurring at a specific RNC. A majority of the duplicates occur at a handful of RNCs over a distinct period of time. The RNCs enclosed in the red boxes are "neighbors" on the network indicating localization of duplicates by network as well as in time. By isolating the duplicates in this manner, we can provide a succinct description of the occurrence of duplicates (e.g. "RNCs in city X during time period [time1,time2]") that might help in two ways: (a) identify root causes for physical repair and (2) propose efficient rules for data repair.

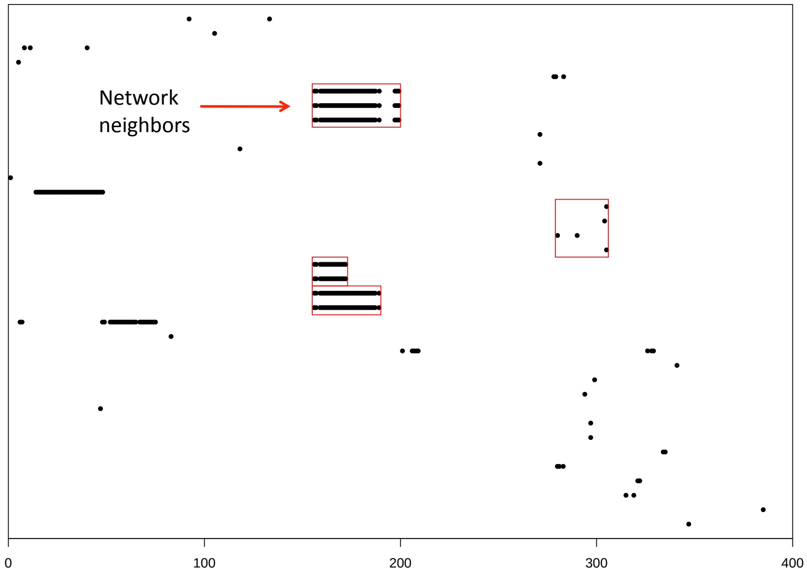

Figure 3: Duplicates grouped in time (*X*-axis) and network topology (*Y*-axis). The red boxes show RNCs that are close as defined by the network topology that often, but not always, translates to spatial proximity. The plot indicates that a majority of the duplicates were generated by a small set of RNCs over a contiguous period of time.

Missing Values

Missing data can occur in two ways: an entire record or any combination of individual attributes can be missing.

We can determine missing records because we expect data from every SECTOR of every NODE B of every RNC at each TIME interval. When entire records are missing, these records can be analyzed to identify any patterns, e.g. all the missing records may be associated with a particular RNC or time period. We found that such records constitute 0.01% of the total number of records. They occurred at the same contiguous time period, affecting only 5% of all the RNCs. Identifying such patterns facilitates data repair as well as their effects on subsequent analyses.

Records with missing USAGE attribute constituted around 0.08% of the data. Almost all these records had a low value (< predetermined constant K) of SAMPLES, indicating that the polling process was incomplete when these records were collected.

Note that an RNC has many Node Bs associated with it, and each Node B has several sectors associated with it. It is not unreasonable that over a duration of several months at least some sector associated with any given RNC would have missing values. Often the polling is stopped for maintenance or to upgrade software and hardware components.

Inconsistencies

For the purposes of this case study, we focused on three logical inconsistencies. (1) USAGE cannot be negative; (2) PERFORMANCE must lie within a fixed interval; (3) If SAMPLES is zero, then the attribute USAGE is meaningless since is not possible to report usage without reporting the number of samples. The distribution of inconsistencies is summarized in Table 1. Some inconsistencies (negative usage) are localized to a few RNCs at specific time periods while others (negative performance) are distributed more widely across time and RNCs.

Outliers and Anomalies

In this section, we study outliers detected in the data stream. We use a 14-point sliding window for statistical techniques based on exponentially weighted moving averages to identify univariate and multivariate outliers. While we use specific techniques described or referenced in the "Statistical Methods" section, note that any outlier detection technique can be used.

Univariate and Multivariate Outliers

It is instructive to look at the differences between outliers detected using univariate and multivariate methods. Figure 4 shows a plot of PERFORMANCE (red line at the bottom of the plot) and USAGE (blue line at the top of the plot) for a particular RNC. The *Y*-axis represents the values of these two attributes, transformed to fit in a single plot, while the *X*-axis represents time. The red and blue dots are respectively PERFORMANCE and USAGE outliers, detected at 0.05 level of significance based upon marginal distributions. Bivariate outliers represented by black dots are detected using Hotelling's T^2 method based on the joint distribution of the two variables.

We find that there are multivariate outliers without corresponding univariate outliers, and vice-versa, as discussed earlier. The exclusively detected multivariate outliers (shown within red boxes) appear to occur when there is a trend in one or more of the variables (in this example, in PERFORMANCE) even though marginally the values do not appear to be extreme. This suggests that the procedure is detecting a change in the correlation between the variables. On the other hand, the exclusively univariate outliers identify moderately extreme values (e.g. the blue dot near day 275). This example demonstrates the importance of using multiple outlier detection techniques.

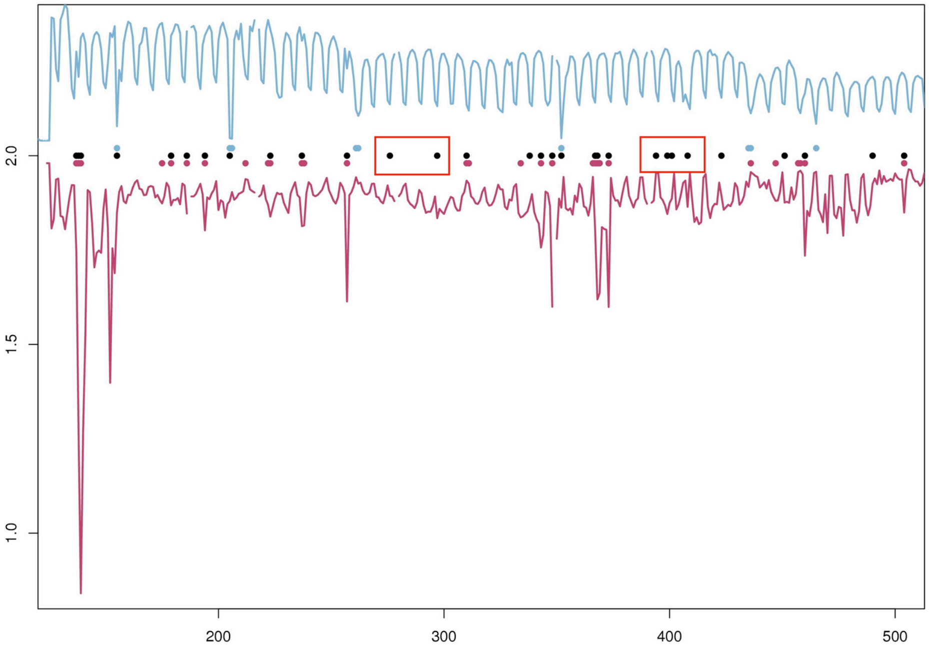

Figure 4: Outliers for two attributes for an RNC. Blue and red dots are univariate outliers, while black dots are multivariate outliers with respect to the joint distribution of the two attributes.

Correlations between Outlier Sequences

Here, we investigate the correlation between outliers across time, space and network topology. While we focus on outliers, this technique is applicable to sequences of any type of data glitches. In this particular case, we expect outliers to cluster together in the network and in time. This is because outliers are often caused by abnormalities that are propagated through the network over time.

Correlation by network topology

We find that outliers of network components in a hierarchical structure are indeed correlated. In Figure 5, we show a plot of the outliers of a given RNC, of 4 of its Node Bs, and of all the sectors in the selected Node Bs. The figure is divided into 4 horizontal bands, separated by gray dotted lines, corresponding to each of the 4 Node Bs. Within each band, the Node B outliers are plotted (blue dots) along with those of its sectors (red dots) and of its parent RNC (black dots). To facilitate comparison, a vertical line is drawn through each RNC and each Node B outlier across the band.

Since all 4 Node Bs are from the same RNC, the black dots and lines are the same across bands. However, there is variability in the outliers between Node Bs – not all the blue dots across bands match. The sector outliers do not necessarily match either. However, outliers in sectors from the same Node B (red dots within each band) appear to match more closely than outliers of sectors from different Node Bs (red dots across bands). See, for example, the lower band near time 380 in Node B 4.

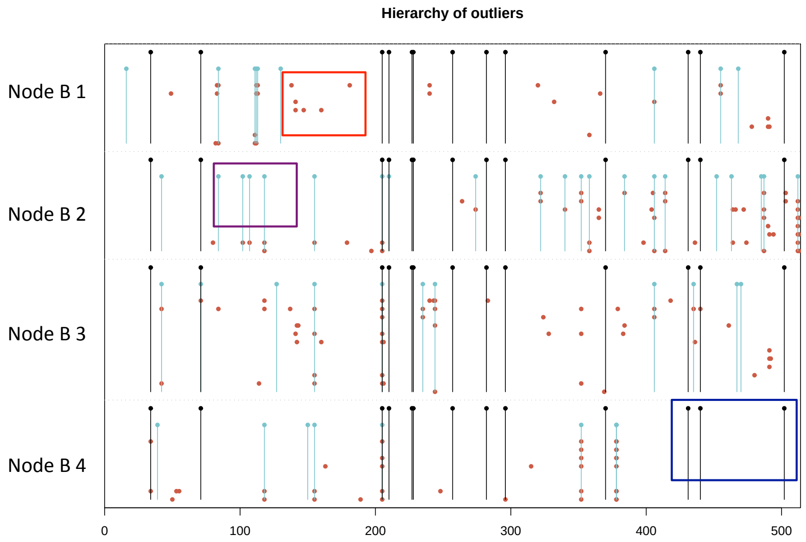

Figure 5: Outliers in sectors (red dots), associated Node B outliers (blue dots with bar) and outliers in the parent RNC (black dots with bar.) Red outliers without blue outliers (red box, Node B 1), and blue outliers without black (purple box, Node B 2), imply that outliers get washed out upon aggregation from sector to Node B, and Node B level to RNC level.

We note that there are sector outliers (red box in Node B 1) that do not have corresponding RNC outliers – there are no corresponding black dots in the figure. In such cases, the outliers at the sector level have been washed out by other sectors associated with the same RNC. This is an example where a signal at a granular level gets swamped upon aggregation to a coarser level. The same phenomenon occurs also for Node B outliers (see purple box in Figure 5). In addition, we find RNC outliers (blue box in Node B 4) that do not have corresponding outliers in the Node Bs or sectors shown in the plot. This is because these outliers are associated with Node Bs and sectors that belong to the RNC but are not shown in the plot.

Next, we investigated the time correlation between outlier sequences detected at the sector level. We used the cross *g* function to measure the degree of matching between outlier sequences of sectors within the same Node B, and of sectors in different Node Bs. Specifically, we computed the cross *g* function for all pairs of sector outlier sequences from the same Node B, and found its mean. We call this the "within Node B" mean cross *g* function.

We repeated the procedure for all pairs of sector outlier sequences where each sequence belonged to different Node B. We call this the "across Node B" mean cross *g* function. These are shown in Figure 6 as solid black and gray lines for the within and across Node B mean cross *g* functions respectively. Note that the mean cross *g* functions are much higher for smaller lags, indicating that outliers do bunch together in time. Furthermore, the mean cross *g* function is higher for sector outlier sequences within the same Node

Bs (solid black line) compared to across Node Bs (solid gray line), suggesting a greater degree of matching of outlier sequences for sectors within the same Node B.
Note that individual cross *g* functions for any pair of sector outlier sequences can be quite variable but are relatively stable when averaged over all possible pairs of sector outlier sequences.

We repeat the above procedure by replacing Node Bs with RNCs. Figure 6 also shows mean cross *g* functions for pairs of Node B outlier sequences within the same RNC (solid blue line) and from different RNCs (solid cyan line). We find that the same results hold for RNCs. The mean cross *g* function for Node B outlier sequences within the same RNC is higher than that for Node B outlier sequences across RNCs. Note also that the correlations are higher for sector sequences than Node B sequences (the black and gray lines are higher than the blue and cyan lines). This is not surprising because the sectors of a Node B are generally at the same physical cell tower. On the other hand, Node Bs in an RNC tend to be more widely scattered spatially and geographically and hence may experience different physical and environmental factors.

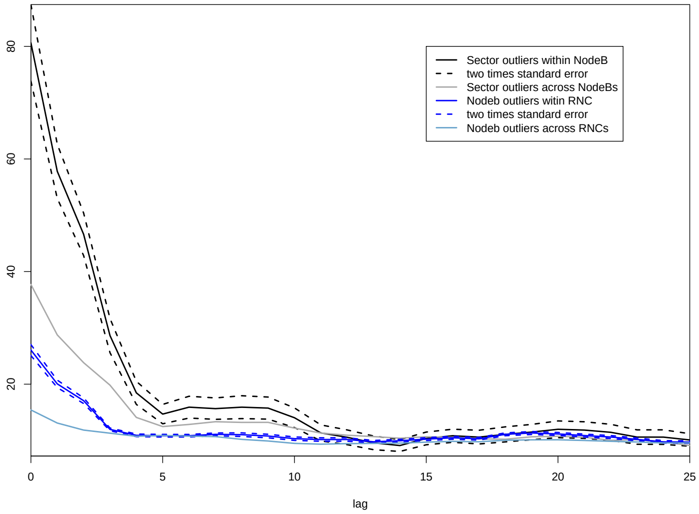

Figure 6: Mean cross *g* function for outliers averaged over all pairs of sectors within Node B (black line represents cross *g*, dashed lines represent two standard deviation bands), and across Node Bs (gray line). Blue lines correspond to mean cross *g* function for outliers for all pairs of Node Bs within RNC (blue line) and across RNC (cyan line).

Correlation by geography

Next, we studied how the time correlation between outlier sequences varied with spatial separation. We selected 4 physically disparate locations, indicated in Figure 7 by black triangles. We identified the 10 nearest Node Bs for each of these locations and extracted the multivariate outliers, so that we have 4 sets of 10 outlier sequences corresponding to the 4 geographical regions.

We computed the cross *g* function for each pair of Node B multivariate outlier sequences (a) within sets and (b) across sets, for time lags from 0 to 25, and computed the mean of the within set and of the across set cross *g* functions (see Figure 8). As expected, the Node B outlier sequences show greater correlation within regions (solid blue line) than across regions (solid cyan line).

When we used sector outlier sequences corresponding to the 4 geographical regions, instead of Node B outlier sequences, we reach a similar conclusion – sector outlier sequences from the same geographical region (solid black line) show higher degree of matching than sector outlier sequences from disparate geographical regions (solid gray line). Note in addition that the sector correlation is greater than the Node B correlation, suggesting that there is greater degree of matching between sector outlier sequences than Node B outlier sequences.

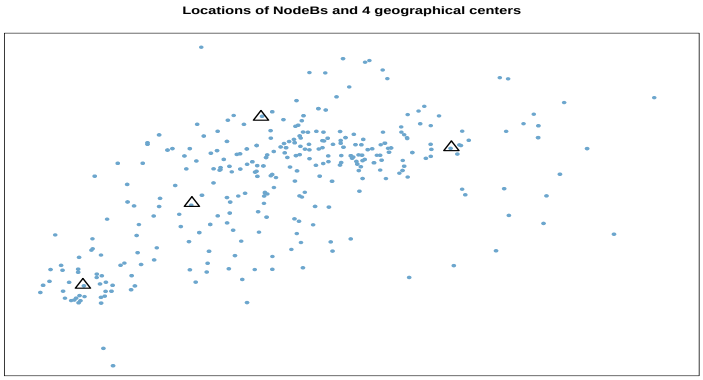

Figure 7: Four geographical reference points (triangles) and Node B's (blue dots) near them.

Finally, when we compare Figure 6 and Figure 8, we find that the mean cross *g* function is higher, especially at the smaller lag values, for pairs of outlier sequences that are within the network hierarchy than for sequence pairs within the same geographical region. This suggests that outliers are more dependent through the network topology than by geographic proximity. This is reasonable since the underlying causes for outliers are more often network phenomena propagated along the network graph rather than physical phenomena.

The bumps in the within-region mean cross *g* function at time points 12 and 24 in Figure 8 indicate possible periodicities in time, perhaps due to a maintenance schedule.

CONCLUSION & FUTURE WORK

In this paper, we highlighted the challenges of managing the data quality of mobility network data streams. The data, consisting of rapidly accumulating time series of hundreds of thousands of network components, is both massive and complex. Data cleanup requires expert domain knowledge. Only summaries of data can be stored. Statistical techniques need to be lightweight yet effective. An effective methodology for maintaining data quality needs to address each of these issues within a consistent framework that allows smooth progression from cleanup to analysis.

We presented our methodology in context of a specific case study. Our investigations show that mobility network data streams can be riddled with data quality issues. We developed a set of domain-specific constraints that allows automatic identification and repair of simple data glitches, such as duplicates and missing values. More importantly, we developed a general methodology for studying the temporal, geographical and topological correlations of detected outliers. This methodology is generally applicable to any data stream and any type of glitch sequence.

Our future work leads us in several directions. First, while we currently employ both univariate and multivariate outlier detection techniques, the results of the univariate procedure only serves as a quality check for the multivariate procedure. We believe that a more integrated use of univariate and multivariate outlier detection methods will be more effective in identifying different types of outliers. There needs to be proper calibration of thresholds when the methods are used together. A system needs to be in place to (a) adjust one procedure based on the output of the other procedure and (b) to combine the results of both procedures together with accurate measures of confidence.

Secondly, while the cross *g* function allows the examination of correlations at different lags, it is highly variable especially when the outlier sequences are sparse. Further investigation is needed to study other available measures of correlations or to develop new ones. For example, a cumulative version of the cross *g* function (i.e. integrated up to lags *l*) will be more stable. Distance metrics that yield a distance between two point sequences can serve as an overall measure of correlation instead of a correlation at different lags.

The problem of co-occurring and correlated data glitches offers many research opportunities that could potentially have a major impact on automated auditing of complex data streams.

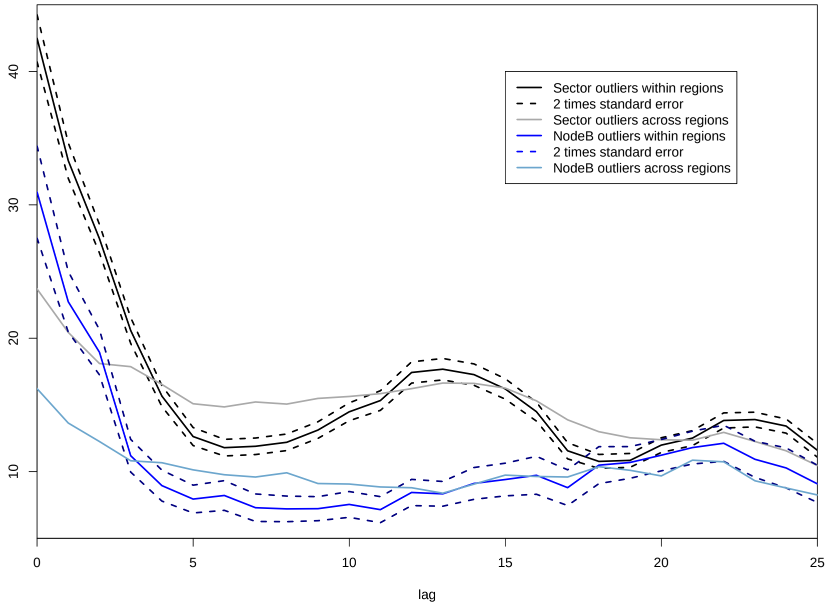

Figure 8: Within-region (solid black line) sector outlier sequences for geographically close Node Bs have a higher cross *g* (better match) than across-region (solid gray line). The cross *g* for sector level outlier sequences is higher than Node B level within-region (solid blue line) which in turn is higher than across-region (solid cyan line) Node B level outlier sequences.

ACKNOWLEDGEMENTS

We would like to thank Dr. Laure Berti-Equille for the use of Figure 2.

REFERENCES

[1] BERTI-EQUILLE, L., DASU, T. Advances in Data Quality Mining, Tutorials at KDD 2009, ICDM 2009.

[2] BERTI-EQUILLE, L., DASU, T., SRIVATAVA, D. Discovery of complex glitch patterns: A novel approach to quantitative data cleaning. *IEEE Conference on Data Engineering, 2011*.

[3] DASU, T., JOHNSON, T., MUTHUKRISHNAN, S., SHKAPENYUK, V. Mining Database Structure; Or, How to Build a Data Quality Browser, Proc. SIGMOD 2002.

[4] ELMAGARMID, A. K., IPEIROTIS, PANAGIOTIS G., VERYKIOS, V. S. Duplicate Record Detection A Survey, IEEE Transations on knowledge and Data Engineering (TKDE) Vol. 19 No. 1 January 2007, pp. 1-16.

[5] GOLAB, L., SAHA, A., KARLOFF, H., SRIVASTAVA, D., KORN, P. Sequential Dependencies. *VLDB 2009.*

[6] KRIEGEL, H., KROGER, P., ZIMEK, A. Outlier Detection Techniques. Tutorial, PAKDD 2009. http://www.dbs.informatik.uni-muenchen.de/Publikationen/Papers/tutorial_slides.pdf

[7] LIN, Y., CHLAMTAC, I. Wireless and Mobile Network Architectures. John Wiley & Sons, 2000.

[8] RAO, C. R. Linear statistical inference and its applications. John Wiley & Sons, 1973.

[9] STOYAN, D., KENDALL, W.S., MECKE, J., 1995. Stochastic Geometry and its Applications, second ed Wiley, New York.

[10] WANG, R. Y., ZIAD, M., LEE, Y. W. Data Quality. Advances in Database Systems, vol. 23. Kluwer Academic Publishers, 2002.