iCoDA: Interactive and Exploratory Data Completeness Analysis

Ruilin Liu [#1], Guan Wang [#1], Wendy Hui Wang [#1], Flip Korn [*2]

[1] {rliu3,hui.wang,gwang6}@stevens.edu
[2] flip@research.att.com[#] *Department of Computer Science, Stevens Institute of Technology*
Hoboken, NJ, USA, 07030

[*] *AT&T Shannon Labs, 33 Thomas Street*
New York, NY, USA, 10017

***Abstract*—The completeness of data is vital to data quality. In this demo, we present $iCoDA$, a system that supports interactive, exploratory data completeness analysis. $iCoDA$ provides algorithms and tools to generate *tableau patterns* that concisely summarize the incomplete data under various configuration settings. During the demo, the audience can use $iCoDA$ to interactively explore the tableau patterns generated from incomplete data, with the flexibility of filtering and navigating through different granularity of these patterns. $iCoDA$ supports various visualization methods to the audience for the display of tableau patterns. Overall, we will demonstrate that $iCoDA$ provides sophisticated analysis of data completeness.**

***Keywords*—Data completeness, graph tableau discovery, exploratory pattern analysis, pattern visualization**

I. Introduction

Data analysts are commonly faced with the problem of missing or spurious data as incompleteness of data has become a ubiquitous problem in practical data management. This problem arises in many domains including sensor networks [1], pharmaceutical testing [7] and social networks [3]. Analytics on incomplete data may lead to misleading results. Therefore, it is vital to fully understand the completeness of data.

A common notion of completeness, especially in network monitoring systems, is that, for a given set of attributes, the cross-product of values from their respective domains exists; this was the subject of our prior work in [6]. For instance, in a sensor network that collects real-time data, normally all sensors poll data at every time stamp. Enumerating each individual missing tuple is not only inefficient but also not informative. Presenting a concise representation by clustering data exhibiting similar loss can often yield insights towards identifying root causes. To achieve this goal, the notion of *tableau patterns* was defined in [6] to summarize the distribution of data loss. It has been demonstrated in [6] that tableau patterns can discover meaningful patterns of missing data.

Motivating Example: Consider a data set in Table I that records temperature readings by multiple sensors in a weather network. Normally the sensors report their temperature readings on an hourly basis. However, some sensors failed to report any data at some time points. For instance, Sensor 4 only reports a reading for 11AM and not for 9AM, 10AM, or 12PM. The following two tableau patterns, which are ordered pairs consisting of a time interval and object set, represent data of high loss (where at least 75% of desired measurements are missing):

ID	Time	Temperature reading
Sensor 1	9AM	60F
Sensor 2	9AM	40F
Sensor 3	10AM	50F
Sensor 1	10AM	62F
Sensor 2	10AM	40F
Sensor 3	11AM	50F
Sensor 4	11AM	60F
Sensor 1	11AM	62F
Sensor 2	11AM	43F
Sensor 3	12PM	55F

TABLE I. Example of a Dataset with Missing Data

Pattern 1: ([12PM, 12PM], {Sensor 1, Sensor 2, Sensor 3, Sensor 4}])
Pattern 2: ([9AM, 12PM], {Sensor 4}).
These two patterns indicate combinations of specific time intervals (e.g., 9-12PM) and specific sensors (e.g., Sensor 4) that should be paid special attention. Therefore, the patterns can provide useful insights of the missing data.

In this demonstration, we present our **interactive Completeness Data Analysis** (**$iCoDA$**) system that constructs spatio-temporal tableau patterns of data completeness; it is built on our previous work [6]. Some important issues from that work remain unaddressed. First and foremost, our previous work did not consider relationships between objects but instead grouped objects arbitrarily into flat sets. Sometimes there is an underlying hierarchy that may be useful for grouping objects, for example, by geographical location or by object taxonomy (e.g., routers contain interfaces which contain subinterfaces). Or there may be other relationships between data objects such as topological connectivity that can help to produce patterns that are useful. We note that many possible equivalent groupings of objects are often possible using the techniques from [6]; considering object relationships can find more judicious groupings from these. In addition, our previous work was rather limited in the notion of incompleteness used for finding patterns, providing only a simple density-based measure that may be more appropriate for some applications than others (e.g., those where missing values follow a uniform or Gaussian distribution over the objects). Finally, our previous work also did not fully exploit visualization of patterns to achieve a better understanding of semantics. To address these issues, we designed $iCoDA$.

The main features of $iCoDA$ are summarized as follows. First, we extend the pattern discovery algorithm from [6] by allowing various new functionality, including adding weights on data objects; using different support and confidence functions in the definition of tableau patterns; and allowing correlations between data objects. Second, we allow three operations, *filtering*, *drill-down*, and *aggregation* on tableau patterns to explore the relationships among patterns. In particular, the filtering operation allows the user to focus tableau patterns from the data on specific regions, while the drill-down and aggregation operations allow the user to navigate the tableau patterns at various granularity. Third, we visualize the patterns using two schemes: (1) *colored rectangles* that are used to display the patterns without explicitly considering spatial/temporal hierarchy, and (2) *sunburst partitions* [9] that uses a radial layout for the display of patterns that involve spatial/temporal hierarchy. In the rest of the paper, we briefly describe the graph tableau pattern problem. Then we present our $iCoDA$ system for exploratory data completeness analysis.

II. Tableau-based Data Completeness Analysis

Data quality research has received much attention recently [5], [4], [8], [2]. In our recent work [6], we formally define the notion of *tableau* to measure the data completeness as a function of time. In particular, let G be a bipartite graph on vertex sets $\mathcal{T}$ and $\mathcal{O}$, with $\mathcal{T}$ denoting time stamps and $\mathcal{O}$ denoting objects. Let $E \subseteq \mathcal{T} \times \mathcal{O}$ be the set of tuples ("edges") obtained from the monitoring system as measurements. Let $\bar{E} = (\mathcal{T} \times \mathcal{O}) - E$. Given thresholds for confidence $\hat{c}$ and support $\hat{s}$, the *Fail Tableau Discovery Problem* is to find a smallest size set T of pairs (I, S) with (interval) $I \subseteq \mathcal{T}$ and $S \subseteq \mathcal{O}$ such that $conf(I, S) \leq \hat{c}$ for each (I, S) in T and $\left|\bigcup_{(I,S)\in T}((I \times S) \cap \bar{E})\right| \geq \hat{s}|\bar{E}|$, where

$$conf(I, S) = \frac{|(I \times S) \cap \bar{E}|}{|I||S|}.$$

Consider the dataset in Table I. The fail tableau pattern ([9AM, 12PM], {Sensor 4}) is of confidence 75% (i.e., at least 75% of readings are missing).

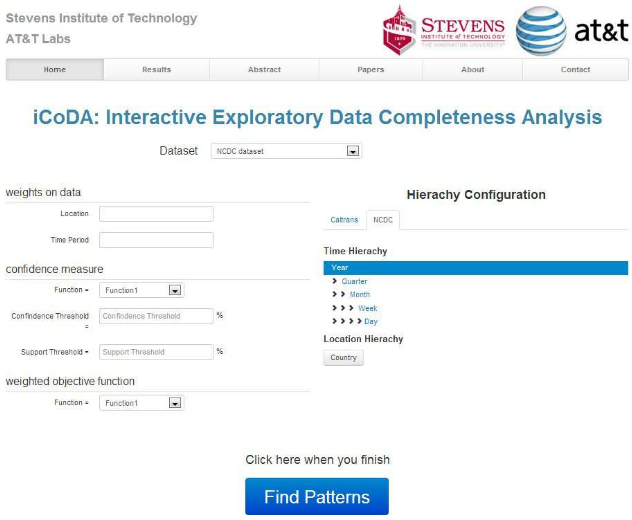
In [6], we showed that it is NP-hard to find an optimal tableau, and developed a polynomial-time approximation algorithm with quality guarantees compared to an optimal tableau.

III. System Description

$iCoDA$ consists of four main components: (1) system configurator, (2) pattern generator, (3) pattern explorator, and (4) pattern visualizer. In this section, we describe the system configurator, the pattern explorator and the pattern visualizer components. The pattern generator component will be built from the tableau discovery algorithm in [6]. We omit the details of the algorithm here.

A. System configurator

In this system, we provide a configuration tool to support tableau-based data completeness summary under various settings, aiming to bring insights to the existence of data completeness patterns, and why such patterns happen. The users can configure iCoDA via the interface shown in Figure 1. The system configuration supports the following settings.



iCoDA: Interactive Exploratory Data Completeness Analysis

Dataset: NCDC dataset

weights on data

Location

Time Period

confidence measure

Function = Function1

Confindence Threshold = Confindence Threshold %

Support Threshold = Support Threshold %

weighted objective function

Function = Function1

Hierachy Configuration

Caltrans | NCDC

Time Hierachy

Year
› Quarter
›› Month
››› Week
›››› Day

Location Hierachy

Country

Click here when you finish

Fig. 1. GUI interface of system configurator

Data with weights. In real world applications, specific data objects or timestamps may have higher importance priority than others. For example, the patterns of missing traffic data in urban areas are more interesting than those in rural areas. Therefore, the system allows the user to specify weights. For example, a GUI is provided for weight specification by regions of space and time (e.g., *urban locations during rush hour*).

Confidence measures. Besides the default density-based confidence measure from [6], we allow the user to choose more sophisticated confidence measures based on the distribution of missing values over a region. For example, a robust confidence measure that requires most objects to exhibit some loss, rather than allowing the loss of a few objects to dominate, may be more appropriate for some applications. Comparing patterns obtained from different confidence measures can help in the design a good confidence measure as well as yielding additional insight into data semantics.

Aspect ratio. Users may prefer to find patterns having distinct properties. For example, a user may be interested in finding the patterns that cover a large number of data objects (e.g., to capture the missing data due to a communication blackout on the whole network), or the patterns that involve large time intervals (e.g., to capture the missing data by detectors that stop working due to mechanical problems). These patterns may give more insights to users for the analysis of missing data. A possible solution to enable generating patterns of specific formats is to enhance the tableau discovery algorithm with *weighted objective functions* [6]. In the demo, we will enable the user to choose various weighted objective functions, e.g., the weighted objective function that prefers tableau patterns of large time intervals but small numbers of objects (i.e., the patterns showing bad time period), or a weighted objective function that prefers "burst" patterns containing a large number of objects over short time intervals.

Correlations of objects and time points. There may exist correlations among data objects. For example, in *CalTrans* road traffic dataset that we will use for the demonstration (more details of the dataset in Section IV), the station ID of data objects can be generalized into highways (based on the route information), or into city, county, state (based on the geographic information). A hierarchy may also exist on the temporal dimension. For instance, individual time points can be grouped into months, quarters, seasons, and years. A taxonomy tree can be used to represent the hierarchy information and *generalize* data objects and/or time points to various granularity to discover *generalized* tableau patterns. Such generalized tableau patterns will help users to analyze data completeness patterns at a coarser level (e.g., the patterns of missing traffic data in a specific region).

B. Pattern Explorator

Our demo aims to enable exploration of tableau patterns so that users can focus on patterns that are interesting to them. Users can utilize *filtering*, *drill-down* and *aggregation* operations on the tableau patterns to better understand the semantic meaning of these patterns.

Filtering. The filtering operation allows users to select time and/or objects that they are interested in. Users can select filtering conditions on data objects and time points; the patterns are generated only from the data that satisfy users' filtering conditions. For example, the GUI allows users to filter over given object attributes and time patterns (e.g., weekdays between 9am-5pm).

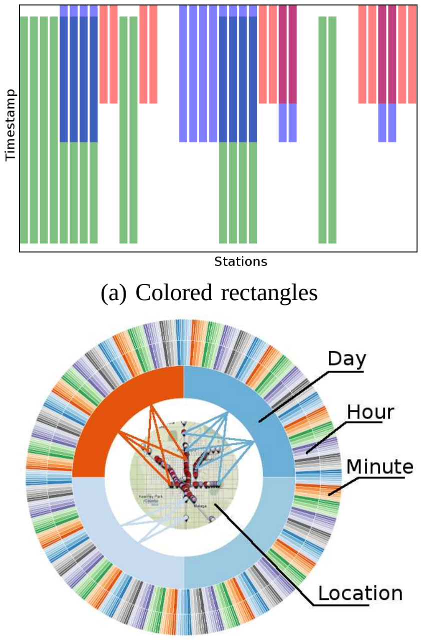
Drill-down. When there exists a hierarchy on data objects or time points, the generated patterns may involve time/objects at a generalized level. The drill-down operation allows the user to navigate these patterns by showing the details at a more detailed level. For instance, given a pattern that involves the time interval shown as January, the user may drill down at the time dimension to the level of day to find which days in January were included in the patterns.

Aggregation. We also provide an aggregation operator that first transforms objects into "superobjects" (e.g., from individual sensors to blocks containing sensors) or time points into time intervals (e.g., from individual polls every 30 minutes to hourly statuses) according to any given dimension hierarchies. The patterns obtained from different levels of granularity are computed on-the-fly.

C. Pattern Visualizer

A good visualization of discovered patterns can help users understand the patterns and missing data in general. In the demo, we will use two visualization methods, namely the *colored rectangles* and *sunburst partitions*.

Colored rectangle based visualization. We use colored "rectangles" to display the tableau patterns. Unlike time, the objects do not necessarily have a natural ordering and, therefore, may not result in contiguous rectangular patterns. Figure 2 (a) shows an example of tableau visualization by using colored rectangles, in which the y-axis denotes time and the x-axis is (lexicographically) sorted by object identifier. Each pattern (I, S) is represented by a set of thin vertical rectangles of the same color (some spread out across the x-axis); different patterns have different colors. The user can click on a color or any slice from a "rectangle", which will highlight the elements corresponding to a pattern. The user can also explore patterns with respect to different object orderings (say, sorted by identifiers or installation dates).



(a) Colored rectangles

(b) Sunburst partitions

Fig. 2. Tableaux Visualization

Sunburst partition based visualization. We will use the so-called "sunburst partitioning" scheme to display the tableau patterns that explicitly consider relationships among objects as well as temporal hierarchy. The sunburst partition scheme uses a radial layout to represent hierarchical data objects [9]. In general, the root node of the tree is at the center, with leaves on the circumference. The area (or angle, depending on implementation) of each arc corresponds to its value. We will adapt the sunburst partition visualization techniques to display tableau patterns for various types of hierarchy information. Below we show two examples of tableau pattern visualization via the sunburst partition. Figure 2 (b) shows an example of tableau visualization of tableau patterns generated from *CalTrans* traffic dataset (more details of the dataset in Section IV) by using sunburst partitions.

Example 1: patterns with hierarchy on time points. To visualize such patterns, a geographic map that displays the locations of the data objects in the tableau patterns will be located at the center of the Sunburst partitions, while time intervals in the patterns will be displayed on the circumference. The closer the circle to the center, the coarser the granularity of time dimension is. Figure 2 (b) shows an example of visualizing patterns with hierarchy on time points. The visualization consists of three circles, representing three levels at the time dimension. For each pair of time interval and data objects (aka detectors in *CalTrans* road traffic dataset) in the generated patterns, it is visualized as an edge connecting a point on the geographic map at center (representing the location of detectors) and a point on one of the circles (representing the time interval). The edges are assigned with different colors; edges of the same color belong to the same pattern.

Example 2: patterns with hierarchy on data objects. This

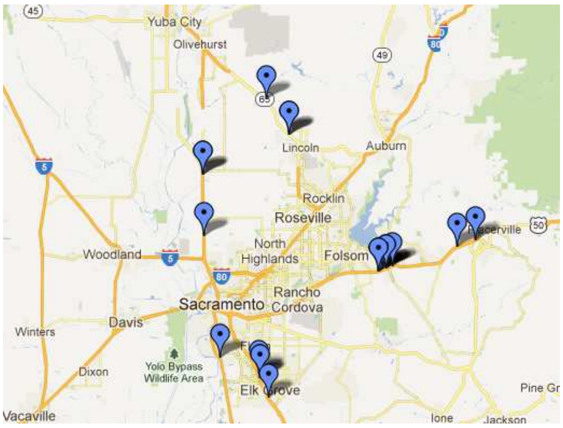


Fig. 3. Locations of stations in top-6 patterns on *CalTrans* dataset

case is similar to the display of hierarchical time points, with an inner circle at the center displaying the time points, while the circumference at various outer circles representing the data objects at different granularity. In particular, the inner circle of the sunburst partitions represents the time dimension, while the stations (i.e., data objects) are grouped on outer circles according to their locations at various granularity. All stations in the same group are (lexicographically) sorted by their object identifiers. Other details are similar to the above visualization with hierarchy on time points.

IV. iCoDA Demonstration

In the demo, we will use two real data sets to show the effectiveness of our system. The two data sets are the *CalTrans road traffic dataset* and the *NCDC Global Summary of Day (GSOD) dataset*. In general, $iCoDA$ can handle data that contains both spatial and temporal information. For instance, $iCoDA$ can be applied to a real-world physical activity monitoring dataset[1] which collects the physical activity data of objects from wearable inertial measurement units and heart rate monitors. This dataset has missing data due to hardware setup problems.

CalTrans road traffic dataset. *CalTrans* road traffic dataset[2] is generated by embedded sensors for cars passing over the detectors scattered in the state of California, US. The data quality analysis tool at the CalTrans website shows that only 71.3% of the detectors worked properly. In the demo, we will demonstrate how $iCoDA$ can provide sophisticated analysis of missing data, for example, finding specific sections of roads/routes along which data was missing (e.g., some detectors located at NB Sunrise Boulevard, Rancho Condova, CA did not work during most of the time interval that was examined). We will consider hierarchies on the dimensions of time and location of sensors. We will use Google Map to visualize the geographical locations of sensors in the tableau patterns discovered by $iCoDA$. An example of such visualization is shown in Figure 3.

NCDC dataset. *NCDC* Global Summary of Day (GSOD) dataset[3] records the climate measurements (temperature, sea level, etc.) collected from over 9000 worldwide stations. However, not all stations collect and report data at all time points due to mechanical problems. Our preliminary work does discover a few interesting patterns of missing data from NCDC dataset. For example, there is exactly one station in Afghanistan that appears in every fail tableau pattern. We suspect that this station may not be functioning due to volatility of that region from war. The remaining stations involved in the discovered patterns were mostly from polar regions, such as Northern Norway, with harsh climates; most of the missing data for these stations was summarized by patterns containing time interval Sept-Dec as well as Jan-March, for both 2010 and 2011. In the demo, we will show how to use $iCoDA$ to discover tableau patterns from the NCDC data, and how to explore and visualize these patterns to help users identify stations and time periods that are involved in large amounts of missing data.

V. Conclusion

This demo introduces a system for exploratory data completeness analysis. It has the ability to support user-friendly configuration, personalized tableau discovery and visualization of tableau patterns. The system shows its effectiveness on data completeness analysis of real world datasets.

References

[1] J. Biswas, F. Naumann, and Q. Qiu. Assessing the completeness of sensor data. In *Database Systems for Advanced Applications*, pages 717–732. Springer, 2006.

[2] P. Bohannon, W. Fan, F. Geerts, X. Jia, and A. Kementsietsidis. Conditional functional dependencies for data cleaning. In *Proceedings of the 23rd International Conference on Data Engineering (ICDE)*, pages 746–755, 2007.

[3] H.-H. Chen, L. Gou, X. L. Zhang, and C. L. Giles. Discovering missing links in networks using vertex similarity measures. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 138–143, 2012.

[4] F. Chiang and R. J. Miller. Discovering data quality rules. *Proceedings of VLDB Endowment*, 1(1):1166–1177, Aug. 2008.

[5] L. Golab, F. Korn, and D. Srivastava. Efficient and effective analysis of data quality using pattern tableaux. *The Bulletin of IEEE Data Engineering*, 34(3):26–33, 2011.

[6] F. Korn, R. Liu, and W. H. Wang. Understanding data completeness in network monitoring systems. In *Proceedings of 12th IEEE International Conference on Data Mining (ICDM)*, pages 359–368, 2012.

[7] J. Mestres, E. Gregori-Puigjané, S. Valverde, and R. V. Sole. Data completenessłthe achilles heel of drug-target networks. *Nature biotechnology*, 26(9):983–984, 2008.

[8] T. C. Redman. *Data quality: management and technology*. Bantam Books, Inc., 1992.

[9] J. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of the IEEE Symposium on Information Vizualization 2000*, pages 57–, 2000.

[1] PAMAP2 Physical Activity Monitoring Data Set UC Irvine Machine Learning Repository: http://archive.ics.uci.edu/ml/datasets/PAMAP2+Physical+Activity+Monitoring.

[2] http://pems.dot.ca.gov/

[3] ftp://ftp.ncdc.noaa.gov/pub/data/gsod/