Path Maps: Visualization of Trajectories in Large-Scale Temporal Data

David Borland*
RENCI, The University of North Carolina at Chapel Hill
Eugenia McPeek Hinz†
Duke Health Technology Solutions, Duke University
Leigh Ann Herhold‡
Duke Center for Health Informatics, Duke University
Vivian L. West§
Duke Center for Health Informatics, Duke University
W. Ed Hammond¶
Duke Center for Health Informatics, Duke University

Abstract

Understanding how a quantity changes over time for multiple entities is a common task when analyzing time-varying data sets. Various temporal visualization techniques exist, however many are ineffective for large data sets. We introduce the path map, a temporal visualization technique designed to effectively handle data sets with many entities. The path map is a rectangular space with columns representing temporal samples and rows representing individual data entities. Rectangular cells with a single color-mapped value are generated from adjacent rows based on their vertical order. An interactive sorting interface reveals patterns in the data by reordering rows vertically based on their values at user-selected columns. Additional contributions include missing data display and aggregation methods to handle larger data sets. We demonstrate path maps using lab data from over 500 and over 3500 diabetic patients.

Index Terms: H.5.2 [Information Systems]: Information Interfaces and Presentation—User Interfaces; I.3.8 [Computing Methodologies]: Computer Graphics—Applications; J.3 [Computer Applications]: Life and Medical Sciences—Health

1 Introduction

Visualization techniques for time-varying data, such as line graphs, stacked graphs, and horizon graphs, effectively show how a given quantity changes over time for multiple entities. However for large data sets with of hundreds or thousands of entities, the effectiveness of such techniques suffers due to problems such as over-plotting.

We are studying the temporal trajectories of measures, such as lab values, related to various diseases for large cohorts of patients in electronic health record (EHR) databases. Many temporal visualization techniques developed for health-related data, although effective, specifically target sequences of discrete events [3, 4], and are not directly applicable to our problem. In previous work we have explored visualization of trajectories in time-varying clinical data [2]. While effective for small numbers of temporal samples, the visual complexity increases with the number of samples, making interpretation difficult when there is a lot of variation in the data. Reducing the number of samples hides any variability between the remaining samples.

We have therefore developed the path map, a visualization technique designed to reveal patterns in large temporal data sets. Path maps are based on heat maps, similar to [1], which was applied only to relatively small data sets. The path map is a rectangular space with columns representing temporal samples and rows representing individual data entities. Rectangular cells with a single color-mapped value are generated from adjacent rows based on their vertical order. An interactive sorting interface enables the user to effectively organize and reveal patterns in the data by reordering rows vertically based on their values at user-selected columns, while still showing the variability between selected samples.

Additional contributions include missing data display and data aggregation methods to reveal patterns in larger data sets. We demonstrate path maps with lab data from over 500 and over 3500 diabetic patients, taken over a period of up to ten years before death.

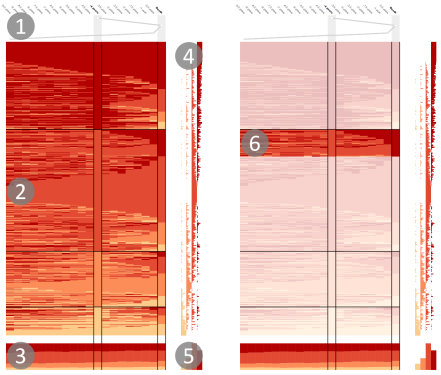

Figure 1: Path map of 546 patients with contiguous map method and custom sorting: 1) sorting interface, 2) path map, 3) column overview, 4) row overview, 5) value overview, 6) highlighting all patients who moved from controlled (orange) to uncontrolled (red) between -4 years and Death.

2 Methods

2.1 Data Description

The path map assumes ordinal data with regular temporal samples. The data shown here are hemoglobin A1c (HbA1c) levels from diabetic patients, aligned by date of death on the right. Visualization tasks of interest include showing general trends, e.g. whether HbA1c levels tend to increase or decrease over time, and identifying subpopulations of interest, e.g. groups of patients that maintain elevated HbA1c levels or whose levels normalize before death. We compute average HbA1c every six months, starting ten years from death. When there are no readings in a sample interval the previous value is carried forward, and the sample marked as missing. Samples prior to the first reading for a patient are colored gray. HbA1c values are categorized according to clinical guidelines as normal, borderline, controlled, or uncontrolled, and color-mapped from yellow to red.

*e-mail: borland@renci.org
†e-mail: eugenia.mcpeek.hinz@duke.edu
‡e-mail: leigh.herhold@duke.edu
§e-mail: vivian.west@duke.edu
¶e-mail: william.hammond@duke.edu

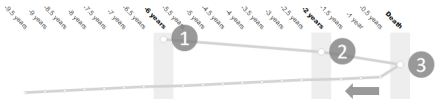

Figure 2: Sorting interface with custom sorting: 1) first by value at -6 years, 2) then -2 years, 3) then Death, and backwards from Death.

2.2 Basic Path Map

The basic path map represents each data entity as a row and each temporal sample as a column (Figure 1). Benefits of the path map representation include ensuring no line crossings, making each individual trajectory easier to follow than with standard line graphs in the presence of many data entities, and an information-dense display. A *contiguous* map method is used, which generates cells from rows with vertically contiguous values per column. An overlay highlights selected columns and outlines regions based on the primary column. Row, column, and value overview visualizations are also drawn to show general trends in the data. Section 2.4 describes other map methods to handle large data sets. For all map methods, highlighting any cell shows the distribution of that cell's rows in all other cells.

2.2.1 Sorting Interface

The vertical position of each row is crucial for finding patterns of interest. We provide five basic sorting methods: *weighted average*, *forward*, *backward*, *first*, and *last*. For each, the user selects a column of interest, and each row is sorted by its value at this column, resulting in a stacked bar. Within each group of rows with the same value, the *weighted average* method sorts by weighted average around the selected column, *forward* and *backward* sort by the value at each subsequent column in the indicated direction (reversing direction at the last column in that direction), and *first* and *last* sort by the value at the indicated column, then forward or backward from there. The *last* method is especially useful for our purposes, as it shows the breakdown of HbA1c values at death given the HbA1c values at the selected sample before death (Figure 1).

A *custom* sorting method enables the selection of any number of columns to sort by, in any order, and a sort direction (forward or backward) for ordering unselected columns. The interactive sorting interface provides a visualization of the column order for all sorting methods (Figure 2).

2.3 Data Status Display

Our data preprocessing carries forward the previous value when there is missing data for a sample interval. We therefore provide the capability to display data status, such as missing, via a striped pattern. Two sorting options are provided. Sorting by status *first* more effectively shows the total amount of missing data, whereas sorting *second* more effectively shows the breakdown of missing data per data value (Figure 3).

2.4 Data Aggregation Map Methods

The *contiguous* map method works well for moderately-sized data sets, however over-plotting becomes a problem when the number of rows exceeds the number of vertical pixels in the path map. We have therefore developed map methods that aggregate data to more effectively handle larger data sets (Figure 3). The same sorting interface is used, however between selected columns aggregation methods are used to generate cells. Rows can be grouped for aggregating based on common values at the column earlier in the sort order (larger groups), or later in the sort order (smaller groups).

2.4.1 Column Summary

The *column summary* map method rearranges row samples to display stacked bar charts for each row group at each intermediate column. This method is most effective for showing general trends between the selected columns.

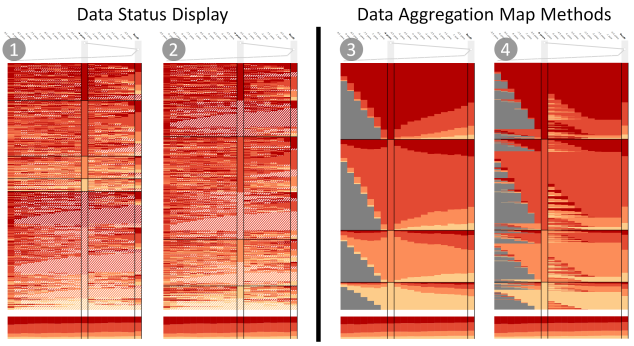

Figure 3: (Left) Missing data status display via striped pattern: 1) sorting by status first, 2) sorting by status second. (Right) Data aggregation map methods for 3638 patients: 3) column summary, 4) row compress.

2.4.2 Row Hierarchy and Row Compression

The *row hierarchy* map method generates a hierarchical layout between each pair of selected columns based on value counts per row. The per-group hierarchy order is determined by the total value counts for the group. Cell width represents the count (number of samples with that value), and cell height represents the number of rows with that count for that value. The *row compression* map method is similar, however rows are initially grouped by the value with the highest count per row, then by remaining values. These layouts are useful for selecting groups of rows with certain characteristics, such as mostly uncontrolled, between selected columns.

3 Conclusion

Path maps have been useful for the visualization of temporal trajectories of HbA1c values in thousands of diabetic patients. We are interested in exploring their use with other medical data, and incorporating additional patient data via coordinated views. We also plan to apply path maps to non-medical time-series data.

Acknowledgements

This work is supported by the US Army Medical Research and Materiel Command (USAMRMC) under Grant No. W81XWH-13-1-0061. The views, opinions and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy, or decision unless so designated by other documentation.

References

[1] L. Chittaro, C. Combi, and G. Trapasso. Data mining on temporal data: a visual approach and its clinical application to hemodialysis. *Journal of Visual Languages and Computing*, 14(6):591–620, Dec. 2003.

[2] E. M. Hinz, D. Borland, H. Shah, V. L. West, and W. E. Hammond. Temporal visualization of diabetes mellitus via hemoglobin A1c levels. In *Proceedings of the 2014 Workshop on Visual Analytics in Healthcare (VAHC 2014)*, 2014.

[3] M. Monroe, R. Lan, H. Lee, C. Plaisant, and B. Shneiderman. Temporal event sequence simplification. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2227–2236, 2013.

[4] K. Wongsuphasawat and D. Gotz. Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2659–2668, Dec. 2012.