Finding Comparable Temporal Categorical Records: A Similarity Measure with an Interactive Visualization

Krist Wongsuphasawat* Ben Shneiderman†

Department of Computer Science & Human-Computer Interaction Lab
University of Maryland, College Park, MD 20742

ABSTRACT

An increasing number of temporal categorical databases are being collected: Electronic Health Records in healthcare organizations, traffic incident logs in transportation systems, or student records in universities. Finding similar records within these large databases requires effective similarity measures that capture the searcher's intent. Many similarity measures exist for numerical time series, but temporal categorical records are different. We propose a temporal categorical similarity measure, the M&M (Match & Mismatch) measure, which is based on the concept of aligning records by sentinel events, then matching events between the target and the compared records. The M&M measure combines the time differences between pairs of events and the number of mismatches. To accommodate customization of parameters in the M&M measure and results interpretation, we implemented Similan, an interactive search and visualization tool for temporal categorical records. A usability study with 8 participants demonstrated that Similan was easy to learn and enabled them to find similar records, but users had difficulty understanding the M&M measure. The usability study feedback, led to an improved version with a continuous timeline, which was tested in a pilot study with 5 participants.

Keywords: Similan, M&M Measure, Similarity Search, Temporal Categorical Records

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces (GUI); I.5.3 [Pattern Recognition]: Clustering—Similarity measures; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Search process

1 INTRODUCTION

Various organizations are increasingly collecting temporal categorical data. Electronic Health Records (EHRs) are being collected by leading health organizations. These EHRs contain millions of records with patient histories. Transportation systems are being monitored at an unprecedented scope which is resulting in gigantic traffic incident logs. Academic institutes are also keeping track of educational advancement of their students. Challenges arise when there is a need to find similar records within these large-scale databases. For example, clinicians want to find patients with similar symptoms to a target patient in order to guide the treatment of the target patient. A major challenge of this problem is defining similarity measures for temporal categorical data.

Many methods for computing a similarity measure between time series have been proposed. However, modifying them to suit temporal categorical data remains an open problem. This paper presents a temporal categorical similarity measure called the *M&M* measure, which is based on aligning temporal data by sentinel events [26], then matching events between two records. If the events are identical between records, then the M&M measure is the sum of the distances (time difference) between the matched pairs. A lower distance represents higher similarity.

The problem becomes more complex when the set of events in the target record does not exactly match those in another record. To accommodate unmatched events, we convert this into an assignment problem and use the Hungarian Algorithm [12, 16] to match events that produce the minimum distance. Consequently, the M&M measure is redefined as a combination of the number of mismatches and the distance.

Furthermore, we believe that an interactive user interface will provide help in finding and understanding results. We developed an interactive interface, Similan, that allows users to adjust parameters of the M&M measure and see the results in real time. (See Figure 1.) Similan adopts the alignment concept from LifeLines2[26] and allows users to preprocess the dataset by aligning events by a sentinel event. Similan displays all events in a timeline for each record. Our extension to the rank-by-feature framework [23] allows users to select a target record and then adjust the ranking criteria to explore the impact of result order.

Records are simultaneously visualized on a coordinated scatterplot according to the number of mismatches and the distance function. The comparison panel provides more advanced exploration. When users select one record for a detailed comparison with the target record, they see links between events, enabling them to understand how close the relationship is.

This paper is organized as follows: Section 2 covers the relevant history of similarity searching, temporal data visualization and related areas. Section 3 provides a brief explanation of the M&M measure. Section 4 introduces Similan and describes the user interface. Section 5 explains the M&M measure in more details. Section 6 describes a usability study done to evaluate the interface. We follow by a brief discussion about the new version in Section 7, describe future work in Section 8, and conclude in Section 9.

2 RELATED WORK

A growing body of recent work is focused on similarity searching, mainly in the medical domain. For example, the national health insurance system in Australia records details on medical services and claims provided to its population. Tsoi et al. [25] proposed a method to classify patients from medical claims data into various groups. Their aim is to detect similar temporal behavioral patterns among patients in the dataset. *PatientsLikeMe* [10] is an online community where patients with life-altering diseases share and discuss personal health experiences. Users enter their structured data on symptoms, treatments, and health outcomes into the site. This information is rendered as data visualizations on both an individual and an aggregate level. Users can also search for similar patients by specifying demographic information. Unlike PatientsLikeMe, this work focuses on a sequence of events (symptoms, treatments, and outcomes) in patient records, which is a special type of time series called *temporal categorical data*.

*e-mail: kristw@cs.umd.edu
†e-mail: ben@cs.umd.edu

IEEE Symposium on Visual Analytics Science and Technology
October 12 - 13, Atlantic City, New Jersey, USA
978-1-4244-5283-5/09/$25.00 ©2009 IEEE

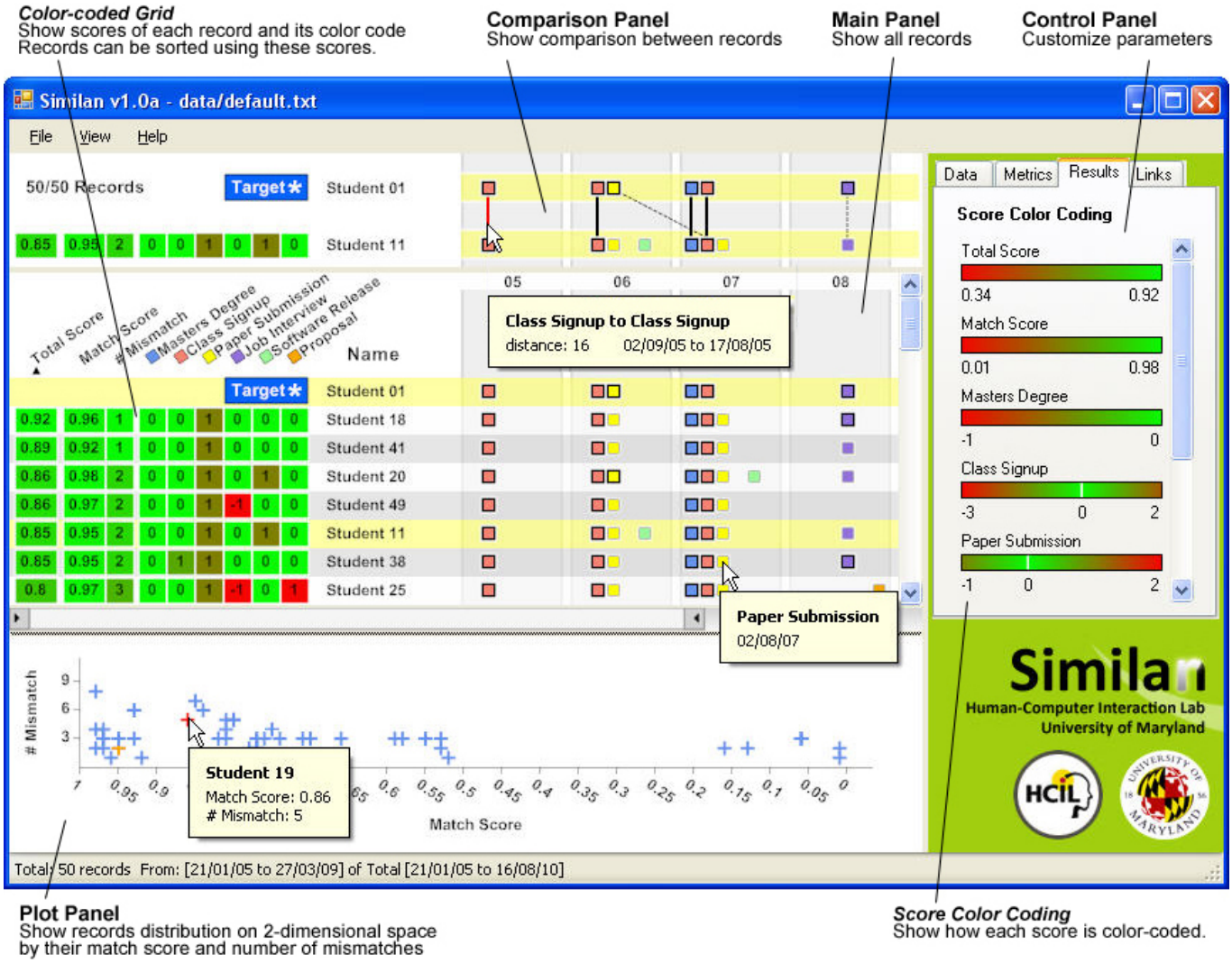

Figure 1: Users can start by double-clicking to select a target record from the main panel. Similan will calculate a score that indicates how similar to the target record each record is and show scores in the color-coded grid on the left. The score color-coding bars on the right show how the scores are color-coded. The users then can sort the records according to these scores. The main panel also allows users to visually compare a target with a set of records. In this early prototype, the timeline is binned (by year, in this screenshot). If the users want to make a more detailed comparison, they can click on a record to show the relationship between that record and the target record in the comparison panel on the top. The plot panel at the bottom shows the distribution of records. In this example, the user is searching for students who are similar to Student 01. The user sets Student 01 as the target and sorts all records by total score. Student 18 has the highest total score of 0.92 so this suggests that Student 18 is the most similar student. Student 41 and Student 18 both have one missing paper submission but Student 41 has a lower match score so Student 18 has higher total score.

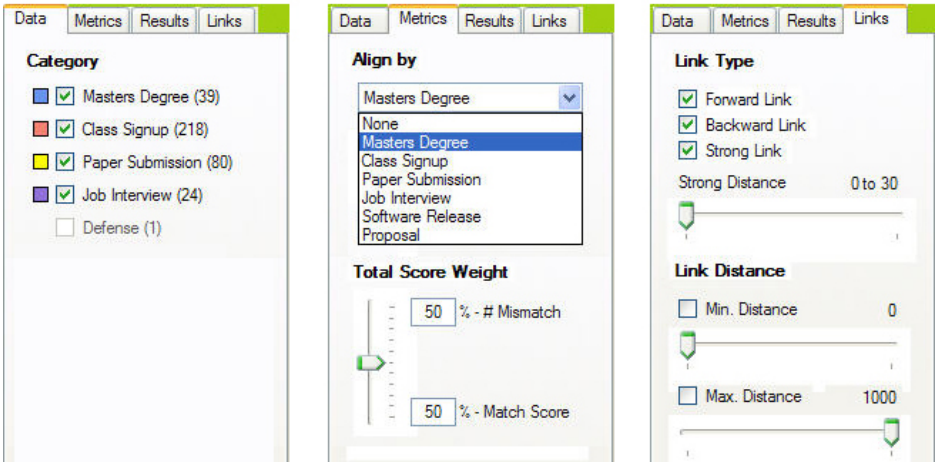

Figure 2: Control Panel: Users can select categories of interest (left). The numbers of events in each category are displayed in the label. By clicking on the colored squares users can customize color. Users can choose to align events by selecting a sentinel category (middle). Weight for calculating total score can be adjusted using sliders and textboxes (right). Links in the comparison panel can be filtered using these parameters.

A time series is a sequence of data values, measured at successive times, spaced at (often uniform) intervals. One notation is:

$$X = \{(t,v) \mid t \in Time\}$$

Temporal categorical data, e.g. athlete injuries, is one type of time series. However, unlike the *numerical time series*, e.g. stock indices, every v is not a numerical value (1, 2.9, 3.5, ...), but a category ("Jammed finger", "Broken leg", etc.). For temporal categorical data, we will call each (t,v) an *event*. The i-th event in the sequence is denoted by x_i.

Stock Indices *(Numerical)*	Injuries *(Categorical)*
(10/8/07, 540.35)	(10/18/07, "Jammed finger")
(10/9/07, 555.32)	(11/10/07, "Broken leg")
...	...
(12/1/07, 410.94)	(12/31/08, "torn ACL")

Many similarity measures between numerical time series have been proposed. According to the surveys of previous methods by Ding et al. [8] and Saeed and Mark [22] , similarity measures for time series can be grouped into various types.

The first type is *lock-step* measures, which compare the i-th point of one time series (x_i) to the i-th point of another (y_i). The most straightforward measure is the *Euclidean distance*. However, since the mapping between the points of two time series is fixed, these distances measures are sensitive to noise and misalignments in time.

Second, *elastic* measures are distance measures that allow comparison of one-to-many points (e.g., Dynamic time warping (DTW)) and one-to-many / one-to-none points (e.g., Longest Common Substring (LCSS)). *DTW* [5] is an algorithm for measuring similarity between two sequences which may vary in time or speed with certain restrictions. The sequences are "stretched" or "compressed" non-linearly in the time dimension to provide a better match with another time series. DTW is particularly suited to matching sequences with missing information. However, DTW requires monotonicity of the mapping in the time dimension.

Another group of similarity measures are developed based on the concept of the *edit distance* [21], the number of operations required to transform one string into another string. The lower the number is, the more similar the strings are. *Hamming distance* [11], *Levenshtein distance* [13] or *Jaro-Winkler distance* [27] are some examples. The best known such distance is the *LCSS* distance. [3]

Another related area is biological sequence searching. There exist many algorithms for comparing biological sequence information, such as the amino-acid sequences of different proteins or the nucleotides of DNA sequences. Some examples of these algorithms are BLAST [2], FASTA [17] and the TEIRESIAS algorithm [20].

The *transform-based* techniques project time series onto a set of functions such as sinusoids or principal components. The data transformation reduces the dimensionality of the original time series and facilitates the use of machine learning techniques [14] or other methods in matching time series.

However, most of the existing methods are designed for numerical time series and require v to be a numerical value. This motivates the need of a similarity measure for temporal categorical data (when v is a category). The M&M measure is then proposed to support this type of data. The M&M measure is also different from existing approaches in other aspects. It is different from lock-step measures because it does not fix the mapping of i-th events together. Unlike elastic measures, it does not allow one-to-many but allows one-to-none mapping. It is also not limited to monotonicity of the mapping as in DTW. It is different from edit distances and biological sequence searching because the data is sampled at non-uniform intervals and more than one event can occur at the same time while two characters or amino acids cannot occur at the same position in the string or biological sequence.

The first step of the M&M measure is to match every event (t,v) in the target record with an event from the compared record. New challenges arise since there are many possible ways to match events but the M&M measure requires matching which will yield the maximum similarity. This problem can be reduced to a problem called the *assignment problem* [12], which is described as follows [7]:

"There are a number of agents and a number of tasks. Any agent can be assigned to perform any task, incurring some cost that may vary depending on the agent-task assignment. It is required to perform all tasks by assigning exactly one agent to each task in such a way that the total cost of the assignment is minimized."

If the numbers of agents and tasks are equal and the total assignment cost for all tasks is equal to the sum of the costs for each agent, then the problem is called the *linear assignment problem*. When the assignment problem has no additional qualifications, the term linear assignment is used. If there are n tasks, the *Hungarian algorithm* [12, 16, 6] can solve the assignment problem in polynomial time ($O(n^3)$). The first version, known as the Hungarian method, was invented by Kuhn [12]. After Munkres [16] revised the algorithm, it has been known as the Hungarian algorithm, the Munkres assignment algorithm, or the Kuhn-Munkres algorithm. Later, Bertsekas [6] proposed a new and more efficient algorithm for the assignment problem.

Using an absolute time scale alone does not address all of the tasks users face when comparing temporal categorical data. In particular, tasks that involve temporal comparisons relative to important events such as a heart attack are not supported. Wang et al. [26] proposed a concept of aligning temporal data by sentinel (important) events, e.g. heart attack. The time in each record is then recomputed, referenced from the time that the sentinel event in each record occurs. Making time at which the sentinel event, the events before the sentinel event and the events after the sentinel event occur become zero, negative and positive, respectively. Before applying the similarity measure, Similan allows users to preprocess the data by aligning them by sentinel events.

Seo and Shneiderman [23] presented a conceptual framework for interactive feature detection named *rank-by-feature framework*. In the rank-by-feature framework, users can select an interesting ranking criterion, and then all possible axis-parallel projections of a multidimensional data set are ranked by the selected ranking criterion. Similan, inspired by this rank-by-feature idea, allows users to rank the dataset by many criteria derived from the M&M measure.

In order to facilitate the result interpretation, the data records should be visualized in a meaningful way. Ma et al. proposed *Event Miner*, a tool that integrates data mining and visualization for analysis of temporal categorical data. [15] However, Event Miner was designed for analyzing only one record. *Pattern Finder* [9] is an integrated interface for visual query and result-set visualization for search and discovery of temporal patterns. There has been a number of published visualization works on temporal categorical data on timelines. A design using timelines for medical records was proposed by Powsner and Tufte [19], who developed a graphical summary using a table of individual plots of test results and treatment data. *LifeLines* [18] presented personal history record data organized in expandable facets and allowed both point event and interval event representations. Alonso et al.[1]'s experiment to compare a tabular format and the LifeLines representation suggested that overall LifeLines led to much faster response times and can reduce some of the biases of the tabular format. However, their design does not assist comparison between records.

3 Introduction to the M&M Measure

We define a new similarity measure for temporal categorical records called *M&M (Match & Mismatch)*.

The first step of the M&M measure is to match the events in the target record with events in the compared records. Since there can

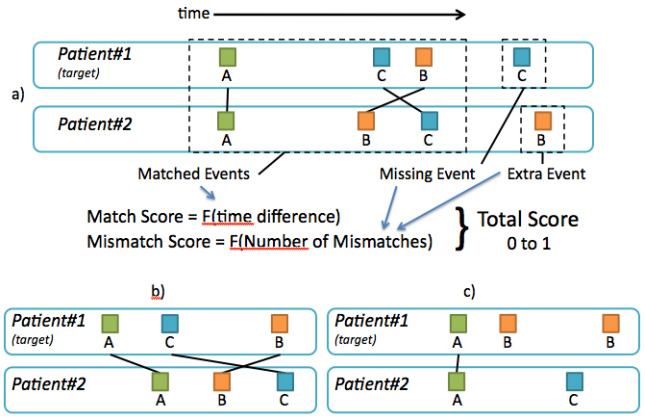

Figure 3: *(top)* The M&M measure *(bottom-left)* High time difference (low match score) but no mismatch (high mismatch score) *(bottom-right)* Low time difference (high match score) but high mismatches (low mismatch score)

be many possible ways to match the events between the two records, we define an event matching process for the M&M measure, which will be explained in Section 5. After the matching is done, the M&M measure is a combination of two measures:

The first measure, *match score*, is for the *matched* events, events which occur both in the target record and the compared record. It captures the time difference between events in the target record and the compared record.

The second measure, *mismatch score*, is for *missing* or *extra* events, events which occur in the target record but do not occur in the compared record, or vice versa. It is based on the difference in number of events in each category between the two records.

Match and mismatch score are combined into *total score*, ranging from 0.01 to 1.00. For all three scores, a higher score represents higher similarity.

4 Similan Interface Design

Given two temporal categorical records, the M&M measure returns a score which represents the similarity between that pair of records. However, the score alone does not help the users understand why records are similar or dissimilar. Also, the M&M measure can be adjusted by several parameters. Furthermore, one of the users' goals is to find the similar records from a database, which contains multiple records. Hence, a tool to assist the users to understand the results, customize the parameters, and perform a similarity search in a database is needed. To address these issues, Similan was developed to provide a visualization of the search results to help the users understand the results, and an interface that facilitates the search and parameter customization. Similan is written in C# .NET using the Piccolo.NET [4] visualization toolkit. The key design concept of Similan follows the Information Visualization Mantra [24] : overview first, zoom and filter, details on demand.

4.1 Overview

Similan consists of 4 panels: main, comparison, plot and control, as shown in Figure 1. Users can start from selecting a target record from the main panel. After that, the main and plot panels give an overview of the similarity search result. Filtering and ranking mechanisms help users narrow down the search result. Users then can focus on fewer records. By clicking on a particular record, the comparison panel shows relationships between that record and the target record on demand. Moreover, mouse hovering actions on various objects provide details on demand in the form of tooltips.

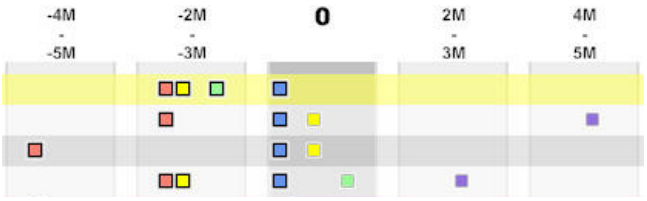

Figure 4: Relative Timeline: Time scale is now relative to sentinel events (blue). Time zero is highlighted in dark gray.

4.2 Events and Timeline

Colored squares are used to represent events. Each color represents a category. Users can customize the colors and check the checkboxes in the control panel (Figure 2) to select interesting categories. The number of events in each category is displayed behind the category name.

Similan's timeline is not a continuous timeline but divided into bins. The bin interval is automatically calculated by the size of application window and total time range of the data. As shown in Figure 1, the timeline is divided into years (05, 06, 07, 08). In each bin, events are grouped by category and categories are placed in the same order. Maintaining the same order allows visual comparison between records.

4.3 Main Panel

Each record is vertically stacked on alternating background colors and identified by its name on the left (see Figure 1). Ranking scores (more details in Section 4.5) appear on the left hand side before the name. Events appear as colored squares on the timeline. By default all records are presented using the same absolute time scale (with the corresponding years or month labels displayed at the top) and the display is sized so that the entire date range fits in the screen.

A double-click on any record marks that record as a target record. A target mark will be placed in front of the target record instead of a ranking score. Clicking on any record selects that record as a compared record. Both the target record and compared record will be highlighted. Users can move the cursor over colored squares to see details on demand in the form of tooltips. Also, zooming on the horizontal axis and panning are possible using a range slider provided at the bottom of the main panel.

4.4 Alignment

Users can select a sentinel category from a drop-down list as shown in Figure 2. By default, the sentinel category is set to none. When the sentinel category is selected, the time scale will change from an absolute time, i.e. real time, into a relative time. The sentinel event becomes time zero and is highlighted (Figure 4).

4.5 Rank-by-feature

Similan is inspired by the idea of *rank-by-feature* from Hierarchical Clustering Explorer (HCE) [23]. These following ranking criteria are derived from the M&M measure proposed in this paper.

1. **Total Score** *ranging from 0.01 to 1.00*
   Total score is the final output of the M&M measure. It is a weighted sum of match and mismatch score. The weight can be adjusted and users can see the result in real-time (Figure 2).

2. **Match Score** *ranging from 0.01 to 1.00*
   This is a score derived from the distance (time difference) between matched events. We choose to display match score instead of distance because the distance is a large number, so it can be difficult to tell the difference between two large numbers and understand the distribution.

3. **Number of Mismatches (#Mismatch)** *ranging from 0 to n*
   This is the total number of missing and extra events compared to the target record. The #mismatch is shown instead of the mismatch score because it is more meaningful to the users. Furthermore, we break down the #mismatch into categories. Positive and negative values correspond to the number of extra and missing events, respectively.

Users can click on the ranking criteria on the top of the main panel to sort the records. By clicking on the same criteria once more, the order is reversed. A triangle under the header shows current ranking criterion. Legends in the control panel show the range of each ranking score and how they are color-coded. (See Figure 1.)

4.6 Plot Panel

In addition to displaying results as a list in the main panel, Similan also visualizes the results as a scatterplot in the plot panel (Figure 1). Each record is represented by a "+" icon. Horizontal axis is the match score while vertical axis is the number of mismatches (#mismatch). Records in the bottom-left area are records with high match score and low number of mismatches, which should be considered most similar according to the M&M measure.

Moving the cursor over the + icon will trigger a tooltip to be displayed. Clicking on a + will set that record to be the compared record and scroll the main panel to that record. Users can also draw a region on the scatterplot to filter records. The main panel will show only records in the region. Clicking on the plot panel again will clear the region and hence clear the filter.

4.7 Comparison Panel

The comparison panel is designed to show similarity and difference between the target record and the compared record. Lines are drawn between pairs of events matched by the M&M measure. Line style is used to show the distance value. *Strong links*, or links with short distance, are shown as solid lines. *Weak links*, or links with large distance, are shown as dashed lines. Events without any links connected to them are missing or extra events. Users can adjust the distance threshold for strong links in the control panel. (See Figure 2.) Moving the cursor over a link will display a tooltip showing the event category, time of both events and distance.

Furthermore, users can filter the links by using the filters (Figure 2). Users can filter by setting the minimum and/or maximum distance. By selecting link types, only the selected types are displayed. *Strong links* are links with a distance in the range specified by the slider. *Forward Links* are links which are not strong links and the event in target record occurs before the event in compared record while *Backward Links* are the opposite.

5 M&M (Match & Mismatch) Measure in Details

This section explains how we define the M&M measure. Our base idea is that similar records should have the same events and the same events should occur almost at the same time. Therefore, the M&M measure uses the time difference and number of missing and extra events as the definition of similarity.

The notation below is used to describe a temporal categorical record, which is a list of temporal categorical events (t,c). The i-th event in the record is denoted by x_i or (t_i,c_i).

$$X = \{(t,c) \mid t \in Time \text{ and } c \in Categories\}$$

5.1 Event Matching Process

The first step of our approach is to match the events in the target record with events in the compared record. There can be many possible ways to match the events into pairs. Therefore, we define a distance function based on a sum of time difference to guide the matching. The matching which produces the minimum distance (time difference) will be selected. Note that the distance from the M&M distance function is not the final result of the M&M measure, but only part of it. This distance is later converted to a match score.

5.1.1 M&M Distance Function

We first define a distance function between each pair of events, as follows:

$$d((t,c),(u,d)) = \begin{cases} |t-u| & \text{if } c = d \\ \infty & \text{if } c \neq d \end{cases} \quad (1)$$

The distance is computed from the time difference if both events have the same category. The granularity of time difference (years, months, days, etc.) can be set. Currently, we do not allow matching between different categories, so we set the distance between every pair of events that comes from different categories to infinity.

Then the distance function between the target record X and the compared record Y

$$X = \{(t_1,c_1),(t_2,c_2),...,(t_m,c_m)\} \quad Y = \{(u_1,d_1),(u_2,d_2),...,(u_n,d_n)\}$$

is described as the following:

$$D(X,Y) = min \sum_{i \in [1,m], j \in [1,n]} d(x_i,y_j) \quad (2)$$
$$\text{each value of } i \text{ and } j \text{ is used exactly once.}$$

A distance function between two records is calculated by matching events from the two records into event pairs and summing up the distances $d(x_i,y_j)$ between each pair. However, this distance function works only when $m = n$ because it requires a perfect match between the two records. Also, even when $m = n$, this case can occur:

$$X = \{(t_1,\text{“}A\text{”}),(t_2,\text{“}A\text{”}),(t_3,\text{“}B\text{”})\}$$

$$Y = \{(u_1,\text{“}A\text{”}),(u_2,\text{“}B\text{”}),(u_3,\text{“}B\text{”})\}$$

$$\text{“}A\text{”},\text{“}B\text{”} \in Categories$$

This will certainly create at least one pair of different category events, which is not preferred. Hence, the distance function fills in some null events $(null,null)$ to equalize numbers of events between the two records in each category. The two lists above become.

$$X = \{(t_1,\text{“}A\text{”}),(t_2,\text{“}A\text{”}),(t_3,\text{“}B\text{”}),(null,null)\}$$

$$Y = \{(u_1,\text{“}A\text{”}),(null,null),(u_2,\text{“}B\text{”}),(u_3,\text{“}B\text{”})\}$$

The distance function between each pair of events is revised.

$$d'((t,c),(u,d)) = \begin{cases} \infty & \text{if } c \text{ and } d = null \\ 0 & \text{if } c = null,\ d \neq null \\ 0 & \text{if } c \neq null,\ d = null \\ d((t,c),(u,d)) & \text{if } c \text{ and } d \neq null \end{cases} \quad (3)$$

The null events should not be paired together so the distance is infinity. The pairs that have one null event indicate missing or extra events. The distance function does not include extra penalty for missing or extra events. Penalty for missing and extra events will be handled separately by the mismatch score (Section 5.2.2). Therefore, the distance is zero in these cases. Last, if the pair does not contain any null event, the original distance function is used.

Finally, a distance function between a target record X and a compared record Y becomes:

$$D'(X,Y) = min \sum_{i \in [1,m], j \in [1,n]} d'(x_i,y_j) \quad (4)$$
$$\text{each value of } i \text{ and } j \text{ is used exactly once.}$$

5.1.2 Minimum Distance Perfect Matching

The problem is how to match every event in X to an event in Y to yield minimum distance. This problem can be converted into an assignment problem. (See Section 2.)

Let events from X ($x_i = (t_i, c_i)$) become agents and events from Y ($y_j = (u_j, d_j)$) become tasks. Cost of the assignment is $d'(x_i, y_j)$. Then use the Hungarian Algorithm to solve the problem. The distance matrix between X and Y is displayed below.

$$\begin{array}{c} \\ x_1 \\ x_2 \\ \vdots \\ x_m \end{array} \begin{array}{c} \begin{array}{cccc} y_1 & y_2 & \cdots & y_n \end{array} \\ \begin{pmatrix} d'(x_1,y_1) & d'(x_1,y_2) & \cdots & d'(x_1,y_n) \\ d'(x_2,y_1) & d'(x_2,y_2) & \cdots & d'(x_2,y_n) \\ \vdots & \vdots & \ddots & \vdots \\ d'(x_m,y_1) & d'(x_m,y_2) & \cdots & d'(x_m,y_n) \end{pmatrix} \end{array}$$

The time complexity of the Hungarian Algorithm is $O(n^3)$ when n is the number of events in each record. If there are m records in the database, the time to perform a matching between the target record and all records, assuming that each record has approximately n events is $O(mn^3)$.

5.2 Scores

Once the event matching process is completed. The match, mismatch and total score can be derived from the matching.

5.2.1 Match Score

The distance from M&M distance function captures the time difference between the two records. However, the distance can be a large number, which users find difficult to compare. Hence, we normalize the distance into a *match score*, ranging from 0.01 to 1.00. A higher score represents higher similarity. Only records with zero distance will yield a score of 1.00. Otherwise, the highest possible match score for non-zero distance is bounded to 0.99. The lowest score is bounded to 0.01 because we think that zero score may mislead the users to think that the target and compared record are not similar at all. Let n be total number of records in the dataset. X and Y are target and compared record, respectively. The match score ($M(X,Y)$) is calculated from the following equations:

$$D'_{max} = Max_{j\in[1,n]} D'(X,Y_j) \tag{5}$$

$$M(X,Y_i) = \begin{cases} 1.00 & \text{if } D'(X,Y_i) = 0 \\ \frac{D'_{max} - D'(X,Y_i)}{D'_{max}} * .98 + .01 & \text{otherwise} \end{cases} \tag{6}$$

5.2.2 Mismatch Score

When the number of events in two records are not equal, there are missing or extra events. A *missing* event is an event that occurs in a target record but does not occur in a compared record. An *extra* event is an event that does not occur in a target record but occurs in a compared record. For example, imagine a target record for a patient who has chest pains, followed by elevated pulse rate, followed by a heart attack diagnosis. If the compared record has only chest pains and heart attack diagnosis, it has one missing event.

We count a *number of mismatches* ($N(X,Y)$), a sum of number of missing or extra events in each category, and normalize it into a *mismatch score* ($MM(X,Y)$), ranging from 0.01 to 1.00. Only records with no mismatch events will yield a score of 1.00. Other records will have score within range 0.01 to 0.99.

$$N_{max} = Max_{j\in[1,n]} N(X,Y_j) \tag{7}$$

$$MM(X,Y_i) = \begin{cases} 1.00 & \text{if } N(X,Y_i) = 0 \\ \frac{N_{max} - N(X,Y_i)}{N_{max}} * .98 + .01 & \text{otherwise} \end{cases} \tag{8}$$

5.2.3 Total Score

The *match score* and *mismatch score* are combined into *total score* ($T(X,Y_i)$) using weighted sum.

$$T(X,Y_i) = w * M(X,Y_i) + (1-w) * MM(X,Y_i)\ ;\ w \in [0,1] \tag{9}$$

Increasing the weight w gives match score more significance while decreasing w gives mismatch score more significance. The default value for weight is 0.5. (Both are equally significant.) For example, the users may not care whether there is any missing or extra event so the weight should be set to 1. Therefore, the Similan interface allows users to manually adjust this weight and see the results in real-time. (See Section 4.5.)

5.3 Discussion

Our concept that the similar records should have the same events (low number of mismatches) and the same events should occur almost at the same time (low time difference) is transformed into the M&M measure. Time difference and number of mismatches are two important aspects of similarity captured by the M&M measure. Records with high match score are records with low time difference while records with high mismatch score are records with low number of mismatches. The M&M measure can be adjusted to give significance to match or mismatch score. By default, the match score and mismatch score are assigned equally weights, so the most similar record should be the record with low time difference and also low number of mismatches.

6 Evaluation

A usability study for Similan was conducted with 8 participants. The goals in this study were to examine the learnability of Similan, assess the benefits of a scatterplot, learn how the number of events and categories affect user performance, and determine if users could understand the M&M measure in the context of its use. We also observed the strategies the users chose and what problems they encountered while using the tool. Synthetic data based on graduate school academic events, such as admission, successful dissertation proposal, and graduation, are used. This choice of data was intended to make the tasks more comprehensible and meaningful to participants, who were technically oriented graduate students.

6.1 Usability Study Procedure and Tasks

Two versions of Similan were used in this usability study: one with full features (S-Full) and another without a scatterplot (S-NoPlot). All usability sessions were conducted on an Apple laptop (15 inch widescreen, 2.2 Ghz CPU, 2GB RAM, Windows XP Professional) using an optical mouse.

The study had two parts. In the first part, participants had an introduction to the M&M measure and training with the Similan interface without a scatterplot (S-NoPlot). Then, the participants were asked to perform this task with different parameters:

Given a target student and dataset of 50 students. Each student record has x categories of events and the total number of events is between y to z events. Find 5 students that are most similar to the target student using S-NoPlot.

Task 1 : $x = 2$, $y = 4$ and $z = 6$; Task 2 : $x = 4$, $y = 6$ and $z = 10$; Task 3 : $x = 6$, $y = 8$ and $z = 16$

In the second part, participants were introduced to the scatterplot and asked to perform task 4, 5 and 6 which are performing task 1, 2 and 3, respectively, but using S-Full instead of S-NoPlot.

The datasets used in task 1-3 and 4-6 were the same but the students were renamed and the initial orderings were different. Task 1 and 4 were used only for training purpose. The results were collected from tasks 2, 3, 5 and 6.

In addition to observing the participants behavior and comments during the sessions, we provided them with a short questionnaire,

which asked specific questions about the Similan interface. Answers were recorded using a seven-option Likert scale and free response sections for criticisms or comments.

6.2 Results

For the first part of this 30-minute study, all participants were observed to use the following strategy: first select the target student, and then use the ranking mechanisms to rank students by the total score. In their first usage, some participants also selected the student who had the highest total score to see more detail in the comparison panel. Afterwards, they just studied the visualization and reported that these students with high total score are the answers.

For the second part of the study, which focused on the scatterplot, most of the participants were observed to use the following strategy: first select the target student, draw a selection in the plot panel, and then use main panels ranking mechanisms to rank students by the total score. However, a few participants did not use the scatterplot to do the task at all. They used the same strategy as in the first part.

Users spent comparable time on tasks 2 and 3 and on tasks 5 and 6. There was no difference in performance times between tasks 2 and 3 or between tasks 5 and 6, even though there were more events in tasks 3 and 6. This is understandable since participants reported that they trusted the ranking provided by the interface. However, users spent more time doing the tasks while using the scatterplot.

All of the participants trusted the *total score* ranking criterion and used it as the main source for their decisions. They explained that the visualization in the main panel convinced them that the ranking gave them the correct answers. Therefore, in the later tasks, after ranking by total score and having a glance at the visualization, they simply answered that the top five are the most similar.

All of them agreed that the main panel is useful for its ranking features and the comparison panel is useful in showing the similarity between the target and a compared student. However, they had different opinions about the scatterplot. Some of the participants mentioned that it was useful when they wanted to find similar students. They explained that the similar students can easily be found at the bottom left of the scatterplot. One participant said that she had to choose two parameters (*#mismatch* and *match score*) when she used the scatterplot. On the other hand, while using the main panel, she had to choose only one parameter (*total score*), which she preferred more. A few of them even mentioned that it is not necessary to use the scatterplot to find similar students. Although they had different opinions about its usefulness in finding similar students, they all agreed that the scatterplot gives a good overview of the students' distribution. It can show clusters of students, which could not be discovered from other panels. Also, one participant pointed out that the main and comparison panels are helpful in showing how students are similar, while the plot is more helpful in explaining how students are dissimilar.

Participants had positive comments on Similan's simple, intuitive and easy to learn interface. Most of the participants got started without assistance from the experimenter. Nevertheless, some user interface design concerns were noted. Some participants noticed that the binned timeline could be misleading in some situations.

Overall, participants liked the simple yet attractive Similan's interface and strongly believed that Similan can help them find students who are similar to the target student. Ranking in the main panel appears to be useful. By contrast, participants had difficulties in learning the M&M measure, since it combines two kinds of scores. The scatterplot did not benefit the tasks in this study but we believe it may prove useful for more complex databases.

7 New Version

According to the user feedback, using a binned timeline can be misleading in some situations. A pair of events in the same bin can have a longer distance than a pair of events in different bins. Also, order of events within the same bin is hidden. Therefore, we develop a new prototype that adopts the continuous timeline used in Lifelines2 [26] and includes several improvements (Figure 5.)

We did a pilot study with 5 participants to compare the binned timeline in original version and continuous timeline in the new version and received these comments: The continuous timeline requires more space for each record and looks more complicated. The binned timeline is more compact, simpler and therefore more readable. It gives users less detail to interpret. However, the continuous timeline does not mislead users when comparing distances or ordering. Both types of timeline have advantages and disadvantages depending on the task. The binned timeline is suitable for general tasks that do not require fine-grain information while the continuous timeline is more suitable when fine-grain information is required.

8 Future Work

Currently, the M&M measure takes all missing and extra events into account. In some situations, missing events are not considered important but extra events are, or vice versa. In a more complex situation, missing 1 of 2 events is not important but missing 2 of 2 events can be considered critical. Moreover, missing an event of category A may be not as critical as category B. The M&M measure also does not allow matching of events between different categories. Allowing matching between different categories may make the similarity measure become more flexible.

Dealing with a large database is a challenging problem. Temporal categorical databases, e.g. EHRs, can contain millions of records. The $O(n^3)$ time complexity of the Hungarian algorithm is a concern, but the number of events for each match is low, so the time to compare against a large data set grows only linearly with the number of records. Anyway, we are improving the algorithm to reduce its time complexity. A signature-based approach may also be used to reduce unnecessary computation.

More filtering mechanisms can be added to help users explore the search results. Also, records may be classified into groups or clusters according to total score, match score, #mismatch, etc.

The existing tools [9, 26] allow users to specify an exact query and retrieve records that satisfy the query. However, by using an exact query, some records may be overlooked because there are some minor details that make them dissatisfy the query. We propose a way to let users specify a loosely defined query by creating a custom record that contains the events that they are interested in. The M&M measure then can be used to calculate the similarity score of between each record and the query record.

9 Conclusion

Temporal categorical data are continuously being gathered by various organizations. Finding similar records within these large-scale databases is a challenging task, especially defining the similarity measure. This paper proposes the M&M measure, a novel similarity measure for temporal categorical data. Briefly, the M&M measure is a combination of time differences between events, and number of missing and extra events.

We also introduce Similan, an interactive tool that facilitates similarity searching and search results visualization for temporal categorical data. The alignment feature allows users to pre-process the dataset by aligning events by a sentinel category. Users are allowed to rank the temporal categorical records by many ranking criteria derived from the M&M measure. The scatterplot provides an overall distribution of search results. The comparison panel provides advanced exploration of relationships between records.

A usability study had been conducted to evaluate the interface. Users found Similan to be comprehensible but they had a hard time understanding the M&M measure. Users expressed strong opinions that Similan can help them find similar records from temporal categorical data.

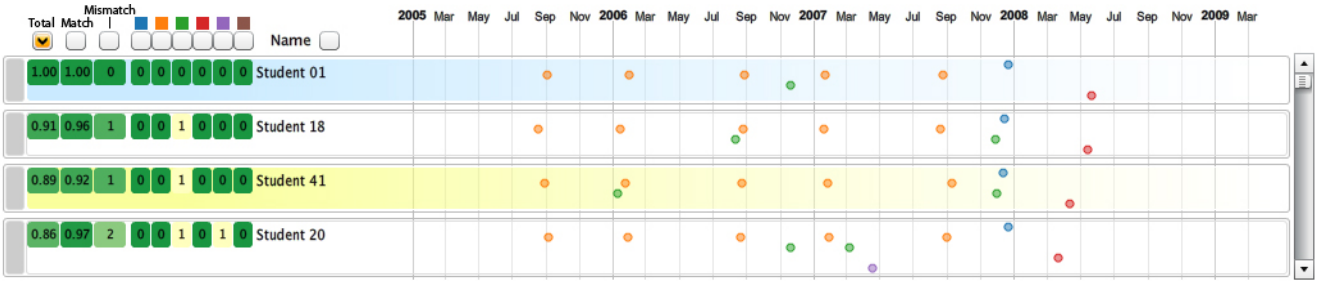

Figure 5: New version: The timeline is continuous and events are split into rows by category.

Learning from the evaluation, we developed a new prototype that has a continuous timeline and ran a pilot study to assess the benefits of the continuous timeline. The result shows that the binned timeline has advantage in its more compact and simpler look while the continuous timeline is more complex but gives the users more fine-grain information.

The M&M measure can be extended further to increase the capability to handle more complex conditions and Similan can be extended into a more powerful tool that allows users to explore various temporal categorical databases using both similarity searching and loosely defined queries. For example, clinicians can use Similan to find patients with similar symptoms to a target patient in order to guide the treatment of the target patient while graduate student committees uses Similan to query for students who published a paper about three months before graduation.

Acknowledgements

We would like to thank Dr. Samir Khuller for his guidance regarding the algorithms, Dr. Amol Deshpande for his guidance on the similarity measures, Dr. Catherine Plaisant, Taowei David Wang and Darya Filippova for their thoughtful comments, and our usability study participants for their time. We appreciate the collaboration and support from physicians at Washington Hospital Center.

References

[1] D. L. Alonso, A. Rose, and C. Plaisant. Viewing personal history records: A comparison of tabular format and graphical presentation using lifelines. *Behaviour and Information Technology*, 17(5):249–262, September 1998.

[2] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic local alignment search tool. *J. Mol. Biol*, 215(3):403–410, 1990.

[3] H. André-Jönsson and D. Z. Badal. Using signature files for querying time-series data. In *Proc. 1st European Symp. on Principles of Data Mining and Knowledge Discovery*, pages 211–220, London, UK, 1997. Springer-Verlag.

[4] B. B. Bederson, J. Grosjean, and J. Meyer. Toolkit design for interactive structured graphics. *IEEE Trans. Software Eng.*, 30(8):535–546, 2004.

[5] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. *AAAI-94 Workshop on Knowledge Discovery in Databases*, pages 229–248, 1994.

[6] D. P. Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 21:152–171, Dec. 1981.

[7] R. E. Burkard, D. Mauro, and S. Martello. *Assignment Problems*. Society for Industrial and Applied Mathematics, 2008.

[8] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh. Querying and mining of time series data: Experimental comparison of representations and distance measures. In *Proc. of the VLDB Endowment Archive*, volume 1, pages 1542–1552, August 2008.

[9] J. A. Fails, A. Karlson, L. Shahamat, and B. Shneiderman. A visual interface for multivariate temporal data: Finding patterns of events across multiple histories. In *Proc. IEEE Symp. Visual Analytics Science and Technology*, pages 167–174, 2006.

[10] H. J. Frost and P. M. Massagli. Social uses of personal health information within patientslikeme, an online patient community: What can happen when patients have access to one another's data. *J. Med. Internet Res.*, 10(3):e15, May 2008.

[11] R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950.

[12] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[13] V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, Feb. 1966.

[14] H. Liu, Z. Ni, and J. Li. Time series similar pattern matching based on empirical mode decomposition. In *Proc. 6th Int. Conf. Intelligent Systems Design and Applications*, volume 1, pages 644–648, Los Alamitos, CA, USA, 2006. IEEE Computer Society.

[15] S. Ma, J. L. Hellerstein, C. shing Perng, and G. Grabarnik. Progressive and interactive analysis of event data using event miner. In *Proc. IEEE Int. Conf. Data Mining*, volume 00, page 661, Los Alamitos, CA, USA, 2002. IEEE Computer Society.

[16] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[17] W. Pearson and D. Lipman. Improved tools for biological sequence comparison. In *Proc. of the National Acad. of Sciences*, volume 85, pages 2444–2448. National Acad Sciences, 1988.

[18] C. Plaisant, R. Mushlin, A. Snyder, J. Li, D. Heller, and B. Shneiderman. Lifelines: using visualization to enhance navigation and analysis of patient records. *Proc. AMIA Symp.*, pages 76–80, 1998.

[19] S. M. Powsner and E. R. Tufte. Graphical summary of patient status. *The Lancet*, 344:386–389, Aug. 1994.

[20] I. Rigoutsos and A. Floratos. Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. *BIOINFORMATICS OXFORD*, 14:55–67, 1998.

[21] E. Ristad and P. Yianilos. Learning string-edit distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(5):522–532, May 1998.

[22] M. Saeed and R. Mark. A novel method for the efficient retrieval of similar multiparameter physiologic time series using wavelet-based symbolic representations. In *AMIA Annual Symp. Proc.*, volume 2006, pages 679–683, 2006.

[23] J. Seo and B. Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Information Visualization*, 4:96–113, 2005.

[24] B. Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. In *Proc. IEEE Symp. Visual Languages*, pages 336–343, Sep 1996.

[25] A. C. Tsoi, S. Zhang, and M. Hagenbuchner. Pattern discovery on australian medical claims data-a systematic approach. *IEEE Trans. Knowl. Data Eng.*, 17(10):1420–1435, 2005.

[26] T. D. Wang, C. Plaisant, A. J. Quinn, R. Stanchak, S. Murphy, and B. Shneiderman. Aligning temporal data by sentinel events: discovering patterns in electronic health records. In *Proc. 26th Annual SIGCHI Conf. Human Factors in Computing Systems*, pages 457–466, New York, NY, USA, 2008. ACM.

[27] W. E. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Census Bureau, 1999.