Journal of Biomedical Informatics 48 (2014) 148–159

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin

A methodology for interactive mining and visual analysis of clinical event patterns using electronic health record data

David Gotz *, Fei Wang, Adam Perer

IBM T.J. Watson Research Center, 1101 Kitchawan Road, P.O. Box 218, Yorktown Heights, NY 10598, USA

ARTICLE INFO

Article history:
Received 22 August 2013
Accepted 17 January 2014
Available online 28 January 2014

Keywords:
Outcome analysis
Pattern mining
Interactive visualization
Visual analytics

ABSTRACT

Patients' medical conditions often evolve in complex and seemingly unpredictable ways. Even within a relatively narrow and well-defined episode of care, variations between patients in both their progression and eventual outcome can be dramatic. Understanding the patterns of events observed within a population that most correlate with differences in outcome is therefore an important task in many types of studies using retrospective electronic health data. In this paper, we present a method for interactive pattern mining and analysis that supports ad hoc visual exploration of patterns mined from retrospective clinical patient data. Our approach combines (1) visual query capabilities to interactively specify episode definitions, (2) pattern mining techniques to help discover important intermediate events within an episode, and (3) interactive visualization techniques that help uncover event patterns that most impact outcome and how those associations change over time. In addition to presenting our methodology, we describe a prototype implementation and present use cases highlighting the types of insights or hypotheses that our approach can help uncover.

1. Introduction

Patients' medical conditions can evolve in complex and seemingly unpredictable ways. Variations in symptom and diagnosis progression vary widely even within a cohort of patients battling the same underlying illness. Similarly, clinicians may use a range of procedures, medications, and other interventions as they develop treatment plans that yield the desired patient outcomes.

For this reason, scientists have long studied how variations in disease progression and care lead to different outcomes. Often, studies are conducted as randomized controlled trials (RCTs) [10,34]. RCTs offer statistical rigor and serve as a "gold standard" for evidence-based medicine, but they are expensive and time-consuming to conduct. This makes it cumbersome to generate and explore new hypotheses.

Researchers have therefore started to perform retrospective studies using the ever-growing repositories of observational data stored in electronic medical record (EMR) systems. For example, a number of platforms have been developed to analyze and make available such data for ad hoc retrospective analyses [4,15].

A common type of retrospective study is temporal event analysis. These investigations represent each patient's medical history as a sequence of time-stamped events. The temporal properties of these events, such as sequence and timing, are then analyzed to discover associations with patients' eventual outcomes. A variety of techniques have been used to gain insights from clinical event sequence data, ranging from data mining systems [27] to interactive visualizations [5].

While mining-based and visualization-based methods have proven useful, they both suffer from significant limitations. Mining-based methods often identify short snippets of frequently occurring patterns. However, the context in which these patterns occur is typically lost. This makes it hard to answer many meaningful questions: "*Do the patterns typically appear early in an episode? Late in an episode? Does the importance of a pattern change over time?*" In contrast, visualization-based methods can illustrate episodes from start to finish, making clear the context surrounding intermediate events. However, temporal visualization technologies are typically limited to a small number of event types before becoming so complex that they are difficult to interpret.

In this paper, we propose a novel visual analytics technique that combines both mining and visualization techniques to overcome the limitations outlined above. Our approach integrates visual queries, on-demand analytics, and interactive visualization to enable exploratory analysis of clinical event sequence data. The query capabilities allow users to intuitively and quickly retrieve cohorts

* Corresponding author. Present address: University of North Carolina at Chapel Hill, School of Information and Library Science, Manning Hall, CB #3360, Chapel Hill, NC 27599, USA.

E-mail addresses: dgotz@us.ibm.com, gotz@unc.edu (D. Gotz), fwang@us.ibm.com (F. Wang), adam.perer@us.ibm.com (A. Perer).

http://dx.doi.org/10.1016/j.jbi.2014.01.007
1532-0464/

of patients that satisfy complex clinical episode constraints. The system then automatically leverages event pattern mining algorithms to uncover important events within the returned cohort. Finally, an interactive visual interface lets users answer a range of interesting questions.

The remainder of this paper is organized as follows. We begin with a review of related work before presenting the details of our methodology. We then briefly describe our prototype implementation and highlight sample use cases demonstrating the utility of our approach. We conclude with a summary and discussion of future work.

2. Related work

This section provides an overview of the related work most relevant to the techniques presented in this paper. This includes systems designed to organize and manage large collections of clinical data for analysis, data mining techniques for temporal event patterns, and interactive visualization applied to clinical event sequences.

2.1. Clinical data analysis systems

Recognizing the potential value hidden within clinical institutions' large collections of electronic medical data, a number of analysis systems have been proposed. For example, i2b2 [15] is a system created by the National Center for Biomedical Computing that combines medical records with genomic data and allows users to create mini-databases targeted for specific research projects. Another NIH-funded system is iDASH [17] which provides a secure platform for data sharing and analysis. *i*HealthExplorer [9] allows users to query for electronic data and choose from a set of analytic algorithms to extract insights from the query results. Support for unstructured data analysis and coding conflict resolution are part of the SHARPn platform [21], making it useful for structuring clinical notes and normalizing data to common standards. Similarly, ICDA [4] is a system for clinical data analysis that provides a normalized data model and allows for the creation of analytic plugins for specific use cases. In addition to these general purpose platforms, many systems are designed specifically for individual conditions (e.g., carcinomas [8] or transplant recipients [24].

Systems such as those cited here are essential for high performance data analysis. The work presented in this paper builds upon the normalized ICDA data model [4], leveraging the data transformation and normalization capabilities that it provides. However, we go beyond ICDA's core capabilities by enabling ad hoc temporal-constraint-based queries to let users interactively define specific patient cohorts of interest. Moreover, we then dynamically perform multiple rounds of data analysis on the retrieved patient cohort and provide an interactive visualization of the results. This contrasts significantly with the batch-oriented analysis workflow supported in the original ICDA system.

2.2. Temporal data mining

Mining knowledge from temporal event sequences is one of the fundamental problems in data mining [11] and Frequent Pattern Mining (FPM) is the key problem in temporal data mining. The goal of FPM is to find a set of frequently occurring subsequences within a collection of longer event sequences. Generally, "frequent" is defined with a pre-specified *minimal support* value, meaning that the mined pattern should appear in at least a certain percentage of the *event sequences*. Here we use the term event sequence we refer to a set of temporally ordered events. Note that there are also approaches looking for frequent episodes from a sequence of continuous values. For example, [25] and [26] look for patterns from SOFA scores. The focus of this paper is to detect patterns from event sequences, which are nominal and we do not consider their values. Note that quite a few FPM methodologies have been proposed, including SPADE [33], PrefixScan [18], SPAM [2], TSKM [12] and temporal abstraction based methods [14].

FPM has been applied within the medical informatics domain to examine the sequentially of medical events (e.g., diagnosis codes, procedure codes and lab tests). For example, Norén et al. [16] proposed a statistical approach for summarizing the temporal associations between the prescription of a drug and the occurrence of a medical event. Moskovitch and Shahar [14] proposed to use temporal abstraction [23] for time interval mining [1] that employed set of features inspired by the Bag-of-Words approach in the text analysis domain [13]. Mörchen and Ultsch [12] proposed a knowledge based temporal mining method which requires a pre-defined temporal grammar and logic with prior knowledge. Wang et al. [27] proposed a FPM framework based on matrix approximation, but the computational complexity of their approach is too high for many real world applications.

This paper focuses on temporal pattern mining from time point-based event sequences, where there is a time point associated with each event in every sequence indicating the time of occurrence, but there are no details about the duration of each event. Moreover, our analytics focus only about nominal events and values associated with events are not considered. For example, for lab test and medication events, our analysis only considers the time point and type of lab test or medication but the corresponding value/dosage is not utilized. However, our approach offers flexibility to incorporate such values, e.g., for each lab test, it is possible to discretize the values into several bins according to clinical references (e.g., critical high, high, normal, low, and critical low lab test thresholds), and then append the bin description after the lab test name to form a new lab name. For example, an HbA1c test with value 6.5 could become HbA1c_normal, while an HbA1c test with value 12 could become HbA1c_critical_high. Using this technique, lab tests values are transformed into clinically relevant events.

2.3. Interactive visualization of event sequences

Many researchers have developed visualization methods for temporal events in the healthcare domain. Early work focused on depicting an individual's medical record, including LifeLines [19] and its derivatives. Similarly, others have visualized an individual's care plan [6]. Such tools typically organize data hierarchically to summarize the complex set of values associated with an individual patient.

More recently, attention has shifted to visualizations of cohorts of patients. This includes a range of tools for visualizing, querying, and sorting through groups of patient event data [28,31,32]. Perhaps most relevant is Outflow, a technique for visualizing aggregate patient evolution patterns in terms of treatments, symptoms, or any other set of temporal event types [5,31].

Such aggregate cohort visualization techniques are very powerful. However, they are constrained in one critical aspect: they typically handle only a limited number of event types. For example, LifeFlow [29] uses color-coding to distinguish between event types and is demonstrated with just six event types. Outflow is not restricted by color-coding, but still has trouble with high numbers of event types (e.g., over 20) which can lead to an extremely complex web of event pathways [5].

This restriction on the number of event types is problematic for medical data given the very large space of possible feature types. For instance, considering diagnoses alone, there are thousands of ICD-9 codes (and even more in ICD-10). As a result, the pre-selection of a small number of event types is typically required. The

approach described in this paper helps in part to address this challenge. In particular, our visual query method retrieves event sequences without filtering to a small subset of event types. As a result, the event sequences visualized in our approach typically contain hundreds or thousands of unique event types. However, rather than include these directly within the visualization, we use automated FPM methods to detect the events and event patterns that are both frequent and statistically significant for the given analysis. We then use a second linked visualization to visualize these detailed statistics.

In other work, visual systems have been used to support temporal pattern search. For example, Fails et al. [3] proposed a visual interface for finding temporal patterns in multivariate temporal clinical data. The interface was further used in [20] for searching temporal patterns in patient histories. However, these systems have no automated "mining" procedure and are best viewed as analogous to the visual query stage of our approach. In contrast, we apply automated FPM algorithms to the sequences returned by a visual query and visualize these automatically mined results.

3. Methods

Our approach consists of three key components: a visual query module, a pattern-mining module, and an interactive visualization module. We combine these three technologies together within a single framework (see Fig. 1) that enables ad hoc event sequence analysis. With this capability, users are able to discover patterns of clinical events (e.g. sequences of treatments or medications) that most impact outcome. Moreover, our approach allows users to better understand how those associations change as patients progress through an episode of interest.

3.1. Visual query

The first component in our episode analysis system is a visual query module. This component has two major features: (1) an easy-to-use user interface component enabling the definition of a clinical episode specification, and (2) a query engine that converts the episode specification to an executable query and retrieves matching patient data from a clinical data warehouse.

3.1.1. Episode specification

We define an *episode* as a sequence of clinical events for an individual patient that matches a specific set of constraints. For example, an episode may include all events for a patient between the initial onset of angina and an eventual diagnosis of heart failure. The rules that define which event sequences should be considered an episode are expressed as an *episode specification*. A valid specification consists of three elements: (1) milestones, (2) preconditions, and (3) an outcome measure.

Milestones are the most important feature of an episode specification. Each specification has at least two milestones to represent the start and end of the episode. For instance, in the earlier example the onset of angina would be the start milestone and heart failure would be the end milestone. In addition, intermediate milestones can be included to encode additional constraints. For example, an arrhythmia could be included as an intermediate milestone to consider only patients who suffered from an irregular heartbeat prior to heart failure. Finally, time gaps can be included to ensure temporal constraints. By default, if no time gap constraints are specified in the episode definition, then all patients matching the milestones will be returned by the query. Patients having a duration of 10 years between milestones would be treated the same as patients with a corresponding duration of just 2 years. For tasks where this is not desirable, time gap constraints allows users to set (a) upper, (b) lower, or (c) exact time limits for the period between milestones. For example, a user could specify a time gap to require at least 10 years between milestones to focus an analysis on only slowly progressing patients. In contrast, a time gap constraint requiring an upper limit of 3 months would allow only quickly progressing patients.

Preconditions are a set of constraints that must be satisfied prior to the starting milestone. For example, a precondition could specify that only patients with a diagnosis of diabetes prior to the onset of angina be included.

The *outcome measure* specifies the way to evaluate the eventual result of an episode. Continuing our heart failure example, the outcome measure for a patient could be, for instance, the presence of an eventual heart valve replacement procedure.

The proposed method is applicable to a wide range of electronic medical data representations. It is agnostic to the level of granularity at which events are represented as long as they are temporally anchored as events in time. For example, our prototype implementation specifies milestones and outcome measures using ICD-9 codes, CPT codes, lab tests, and medication orders. Our approach has also been applied to diagnosis data represented using the higher-level Clinical Classifications Software (CCS) system that clusters patient ICD-9-CM diagnosis codes into a set of clinically meaningful categories.

3.1.2. User interface

Based on this episode specification structure, we have developed an easy-to-use interface that allows users of our system to interactively specify the types of episodes that they wish to analyze. The user interface, shown in Fig. 2(a), includes areas for each of the three portions of a specification. Using the "Add Event" and

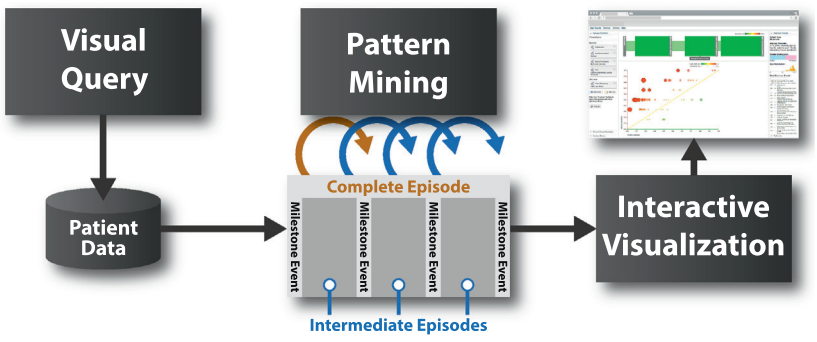

Fig. 1. The visual query component provides an intuitive user interface for authoring episode constraints. Patient data that matches the episode definition is retrieved and passed to the event pattern-mining module. Mining is first performed on the complete episode. The same mining algorithm is then performed on all intermediate episodes. Finally, an interactive visualization lets users explore the results and uncover temporal trends.

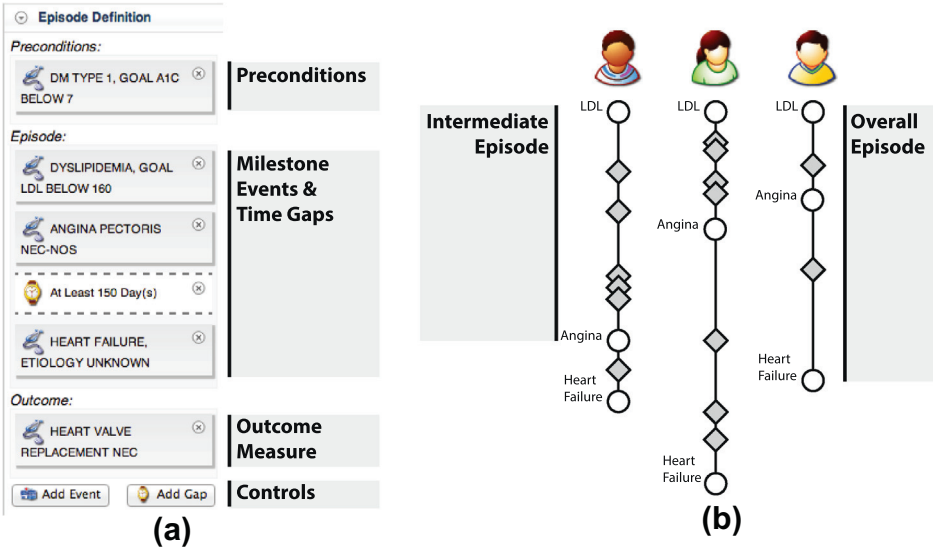

Fig. 2. (a) The visual query interface allows users to supply preconditions, milestones and time gaps, and an outcome measure. When a user submits a query, the specification is converted to SQL and a set of matching patient event sequences is returned. (b) For each patient, the returned event sequence contains the specified milestone events (white circles) and a variable number of intermediate events (gray diamonds) that occur between the milestones. Each overall episode can be subdivided at milestone events into a series of intermediate episodes.

"Add Gap" controls at the bottom of the query panel, users can insert new elements into the specification. Drag-and-drop interaction is used to re-order elements of the query specification, or to move elements between the precondition, milestone, and outcome sections.

3.1.3. Query execution and episode data structure

Once the user has finished defining the episode specification via the user interface, it is translated to a formal query, expressed in SQL, that retrieves matching patient event episodes from a clinical data repository. Except for the step of translating to SQL, our system is independent of the underlying data source. This allows for easy migration between data sources.

The query returns a cohort of patients whose medical records satisfy the episode specification as shown in Fig. 2(b). For each patient, a list of events is retrieved which contains the required milestone events (white circles in the figure), starting with the specification's first milestone and ending with the last milestone. The patient's list of events also includes intermediate events (gray diamonds in the figure) that take place between the episode milestones. We refer to the full sequence of events as the *overall episode*. Only one overall episode is returned for each matching patient. We call the spans of intermediate events between any pair of neighboring milestones *intermediate episodes*.

3.2. Temporal pattern mining

The second stage in our methodology is temporal pattern mining. By pattern, we refer to a specific ordered tuple of events. For the scope of work described in this paper, we consider only the order in which the events occur, not the time between neighboring events. In this stage, *Frequent Pattern Mining* (FPM) is performed first on the overall episode, then again on each of the intermediate episodes retrieved by the visual query module. The FPM engine contains two major elements: the *Frequent Pattern Miner* and the *Statistical Pattern Analyzer*. Before introducing the details of these two modules, we begin with a preliminary discussion to formally define several key concepts.

3.2.1. Preliminaries

We use a number of terms throughout this paper including Event Sequence, Event Subsequence, Support, and Frequent Pattern. We formally define these concepts in this section before moving onto the details of our approach.

Definition 1 (*(Event Sequence)*). An event sequence $S = \langle e1, e2, \ldots, em\rangle$ is an ordered list of events, such that event ei happened no later than $ei + 1$. Here, all the events are from a predefined event space D.

In our investigation, the event space D is a concatenation of all possible medical event types in a dataset of electronic health records. This includes all possible diagnoses, medications, lab tests, and procedures. An event type is represented using a unique label representing the type of event, such as a specific International Classification of Diseases (ICD) or Current Procedural Terminology (CPT) code.

Definition 2 (*(Event Subsequence)*). A sequence $s = \langle r1, r2, \ldots, rt\rangle$ is said to be a subsequence of another sequence $S = \langle e1, e2, \ldots, em\rangle$ if there exists $i1, i2, \ldots$, it such that $1 \leqslant i1 \leqslant i2 \leqslant \cdots \leqslant it \leqslant m$, $r1 = ei1$, $r2 = ei2, \ldots, rt = eit$.

Definition 3 (*(Support)*). Given a set of event sequences $A = \langle S1, S2, \ldots, Sn\rangle$, the support of a sequence s is the percentage of sequences in A that contain s as a subsequence.

Definition 4 (*(Frequent Pattern)*). A frequent pattern s is an event subsequence that appears in A with minimum support α.

From Definition 4 we can see that in order to define a frequent pattern we need to set a minimum support value α, which means that the pattern should appear in α percent of all the event sequences.

3.2.2. Frequent pattern miner

The frequent pattern miner is responsible for detecting frequent patterns occur in a set of input sequences (here an input sequence is just a patient sequence returned by the query). The miner looks

for patterns with a support value above a threshold (which is the minimum support value provided by the user). In our system, the minimum support value is configurable. Users can also specify a minimum pattern length (any integer greater than or equal to one). This can be used to filter out very short patterns that may be less interesting to the user. The core algorithm used for pattern discovery is the bitmap-based *Sequential PAttern Miner* (SPAM) [2] which uses a search strategy that integrates a depth-first traversal of the search space with effective pruning mechanisms. In the following we will introduce more details of the SPAM algorithm.

SPAM adopts a pattern growing strategy, where initially, it starts with an empty frequent pattern set F. Then each single event in the event dictionary (which is constructed by all different events appearing in the event sequences) is checked, and events that are frequent are added to F. These patterns are referred to as frequent patterns of length 1, as the length of a pattern is defined as the number of events it contains. Then the SPAM will grow those frequent patterns with two types of extensions: an S-extension and an I-extension. Both extensions append one event to the end of an existing frequent pattern, but S-extension requires that event happen after the last event in the pattern, while I-extension requires that event happen simultaneously with the last event in the pattern. Basically with these two types of extension, SPAM grows each pattern in F with length 1 and then check whether the new patterns pass the support threshold. If yes, they will be added to F and SPAM will grow them with S- and I-extensions. This procedure will be repeated until no patterns that pass the threshold can be found.

It is important to note that SPAM has been proven to be faster than traditional pattern mining approaches by an order of magnitude, especially when applied to relatively long episodes. This is mainly because SPAM uses a smart bitmap representation for every event sequence before the pattern mining procedure starts. In this representation, each sequence x will be represented as an $|x|$ by $|D|$ bitmap, where $|x|$ is the number of events contained in x and $|D|$ is the number of distinct events in the whole event sequence set. With this representation, all the S- and I-extensions can be performed with bitwise AND/OR operations, which greatly improves the efficiency of the pattern mining procedure. The bitmap representation is illustrated in Fig. 3.

The SPAM algorithm takes as input a set of event sequences (i.e. the episode data) and a user-specified support value, and produces as output a set of frequent patterns. The user-supplied minimum length threshold is then applied to filter out patterns that are too short.

3.2.3. Statistical pattern analyzer

Once a set of frequent patterns has been identified, the next step is to apply the pattern analyzer. This module looks for correlations between the mined patterns and the episode specification's outcome measure. The pattern analyzer first constructs a *Bag-of-Pattern* (BoP) representation for each episode. The BoP representation is an n-dimensional vector for each patient, where n is the number of patterns and the ith element of the vector stores the frequency of the ith pattern found in the episode. If there are m episodes (corresponding to m distinct patients), then we can construct an $m \times n$ episode-pattern matrix $X = [x_1, x_2, \ldots, x_n]$ whose (j, i)th element indicates the number of times the ith pattern appeared in the jth episode. Thus the ith column in $\boldsymbol{X}$ summarizes the frequency of the ith pattern in all m episodes. We can also construct an m dimensional episode outcome vector $\mathbf{y}$, such that y_j is the outcome of the jth episode. In the binary case, $y \in \{+1, -1\}$ with $+1$ representing a positive outcome and -1 representing a negative outcome. The outcome values are determined by the user-specified outcome measure included in the episode specification. Given this formulation, we compute statistics measuring the correlation between each x_i and $\mathbf{y}$. For example, we can compute the Pearson correlation, p-value (to measure the significance of a correlation), information gain (the Kullback–Leibler divergence between the pattern vector and the outcome vector [7]), and odds ratio. Our prototype implementation computes all of the above statistical measures and provides them to the user for interpretation though the user interface described below.

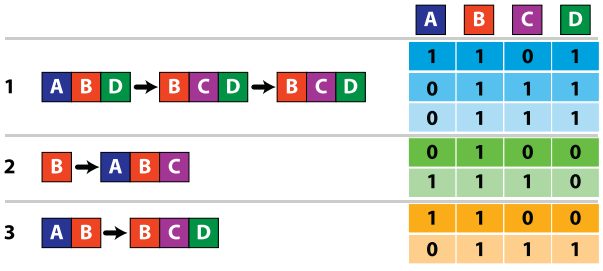

Fig. 3. An example of the bitmap representation for event sequences. In this example, there are four different event types and three sequences.

Definition 5 (*(Pearson Correlation)*). The Pearson correlation between two vectors $x = [x_1, x_2, \ldots, x_m]^T$ and $y = [y_1, y_2, \ldots, y_m]^T$ can be computed as

$$r = \frac{\sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{m}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{m}(y_i - \bar{y})^2}}$$

where $\bar{x} = 1/m\sum_{i=1}^{m} x_i$ is the mean value of all the elements in x, and $\bar{y} = 1/m\sum_{i=1}^{m} y_i$ is the mean value of all the elements in y.

Definition 6 (*(Odds Ratio)*). In order to define the odds ratio of a pattern with respect to a specific outcome, we first construct the following contingency table, where a is the number of patients who have the pattern with positive outcome, b is the number of patients who have the pattern with negative outcome, c is the number of patients who do not have the pattern with positive outcome, d is the number of patients who do not have the pattern with negative outcome. Then the odds ratio of the pattern with respect to the outcome can be computed as (see Table 1)

$$o = \frac{a \times d}{b \times c}$$

Definition 7 (*(Information Gain)*). The information gain, or Kullback–Leibler divergence, between a pattern vector $x = [x_1, x_2, \ldots, x_m]^T$ and an outcome vector $y = [y_1, y_2, \ldots, y_m]^T$ is $IG(y|x) = -H(y) - H(y|x)$ where $H(y)$ is the entropy of the outcome vector, while $H(y|x)$ is the conditional entropy of the outcome given the pattern vector.

3.3. Interactive visualization

The pattern mining results are passed to the system's interactive visualization component. The user interface contains three

Table 1
Contingency table for computing odds ratio.

Pattern existence	Outcome	
	Positive	Negative
Yes	a	b
No	c	d

linked visualization elements as shown in the screenshots in Fig. 6: a cohort overview, a milestone timeline, and a pattern diagram.

3.3.1. Cohort overview

The cohort overview, which is visible in the right sidebar of Fig. 6(a), shows gender and age distributions for the set of patients returned by the query module. In addition, the overview shows other statistics such as the number of patients in the cohort and average outcome.

3.3.2. Milestone timeline

The milestone timeline visualization illustrates the sequence of milestones that define the overall episode. As illustrated in Fig. 4, each milestone is represented as a vertical gray bar that is labeled with the corresponding event type. The milestone bars are arranged in sequential order from left to right and equally spaced horizontally. The milestone bars are then connected by color-coded edges. Each edge has two portions: (a) a full-height section, called a *time edge*, encodes the mean duration between milestone events, and (b) a shorter portion, called a *link edge*, connects the pieces of the view to convey sequentially. The time edge color encodes mean outcome using a red-to-yellow-to-green color scale (green for positive outcomes; red for negative outcomes). Link edges are colored with a less saturated version of the time edge color to make it easy to distinguish the end of the time edge for temporal comparisons. This approach is similar to the encoding adopted in [30]. The milestone timeline is interactive, allowing users to select the overall episode or individual intermediate episodes for display in the pattern diagram. Selections are displayed in the visualization as heavy black outlines. For example, Fig. 6 shows a timeline with three milestones with each panel showing a different selection.

3.3.3. Pattern diagram

The final visualization element is the pattern diagram, which is rendered beneath the milestone timeline. It visualizes the set of patterns mined from the portion of the episode selected in the milestone timeline. Each pattern is represented with a circle in a scatterplot. The *x* and *y* axes reflect the level of support for a given pattern for patients with positive and negative outcomes, respectively. Therefore, patterns that appear primarily in patients with poor outcomes are located toward the top left. Patterns that appear primarily in patients with positive outcomes are displayed toward the bottom right.

The size of each circle represents the corresponding pattern's correlation to outcome (each circle's radius maps to the absolute value of the corresponding pattern's correlation measure). Circle color represents the odds ratio and follows the same green-to-yellow-to-red color gradient used in the timeline. As a result, large red circles represent patterns that led to poor outcomes, while large green circles represent patterns that led to positive outcomes. Circles can be selected via mouse clicks to retrieve more detailed information. Upon selection, a sidebar is displayed to the right of the scatterplot showing both the sequence of events that form that pattern as well as the full set of statistics computed by the mining algorithm. Selection is displayed in the visualization using a bold black outline on the selected circle.

Coupled with the milestone timeline, the pattern diagram provides hierarchical access to a complex set of mined pattern statistics. At the top level of this hierarchy, users can view statistics for patterns found at any time in the overall episode. In addition, users can select specific regions of the episode (i.e., intermediate episodes) via the timeline to see the corresponding set of patterns that appeared only during that portion of the episode. The visual encoding quickly highlights salient events that are most strongly associated with outcome, and selections provide users with access to detailed statistics such as *p*-values.

3.3.4. Temporal comparison for trend analysis

A key feature of the mined pattern diagram is its support for temporal comparison. The significance of an event pattern can change over time. For example, a specific pattern might have a very strong association with outcome during an early intermediate episode despite having absolutely no correlation with outcome later in time. Understanding these trajectories of significance for various patterns is a key goal of our technique.

To help illustrate these temporal trends, the pattern diagram adopts animated transitions whenever the milestone timeline selection changes. Upon any such change, the pattern diagram component compares the "before" and "after" pattern sets and computes three distinct sets: incoming patterns, outgoing patterns, and remaining patterns. Incoming patterns are ones that only exist in the newly selected portion of the episode. Circles representing these patterns are added to the diagram. Outgoing patterns are ones that only exist in the previously selected portion of the episode. Circles for these patterns are removed from the diagram. Most critical are the remaining patterns. The circles for these patterns are animated to new locations, colors and sizes to reflect the change in statistics for the patterns. Therefore, as users click from early to late term intermediate episodes, the pattern diagram shows the trajectory of the discovered patterns as they becomes more (or less) significant and/or prevalent. If an individual pattern is selected (as in Fig. 6b and c), the selection is maintained across the animation. This approach provides features for ad hoc analysis of temporal pattern trends that are analogous to what GapMinder affords for simpler static public health data [22].

For example, consider the hypothetical episode shown in Fig. 5 in which we visualize an episode with three milestones. In panel (a) we see that four patterns have been detected in the overall episode. After the user clicks the timeline to select a specific intermediate episode, three distinct animation steps take place as part of the transition to the final panel (d). First, (b) outgoing patterns—those present in panel (a) but not panel (d)—are removed as shown in panel (b). Next, as illustrated in panel (c), patterns that exist in both (a) and (d) are animated to reflect their new values. This includes changes in position (due to changes in support), color (due to changes in odds ratio), and size (due to changes in correlation). Finally, new patterns that exist only in (d) fade into complete the transition.

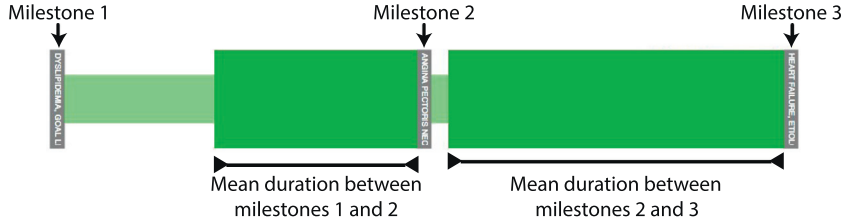

Fig. 4. The Milestone timeline illustrates the sequence of milestones using a set of ordered, equally-spaced, vertical gray bars. The width of the dark green bars between the milestones reflect the mean duration between milestone events across the full set of event sequences.

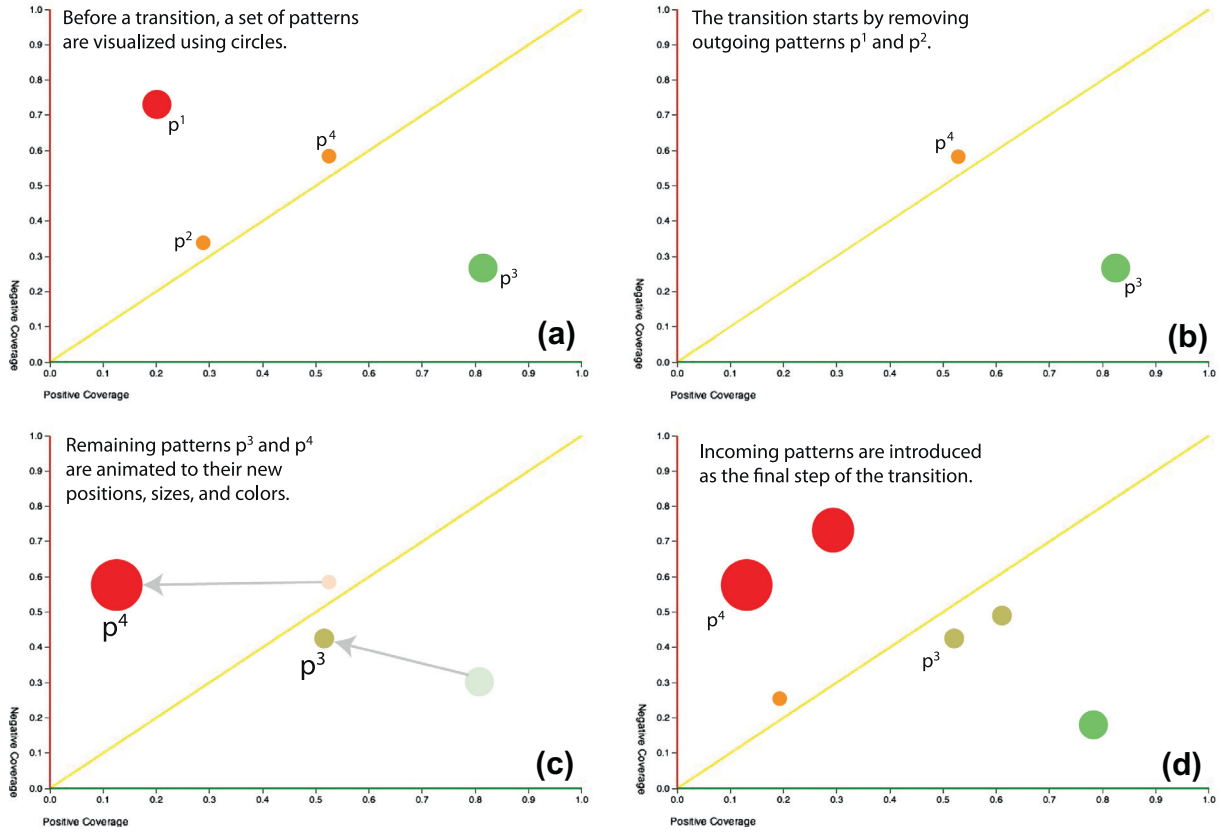

Fig. 5. A three-stage animation process is used to transition between views in the Pattern Diagram view. This process can help highlight temporal trends by enabling comparison between mining results at different stages of an episode.

4. Prototype

We have built a web-based prototype system following the methodology outlined above. The system uses Servlet technology and uses the open-source Apache Tomcat web server. All server-side functionality is implemented in Java. Data is stored in an ICDA-based data warehouse [4] employing widely used standards such as ICD (International Classification of Diseases), CPT (Current Procedural Terminology, and NDC (National Drug Code). This platform was chosen because a version of the ICDA system has been previously deployed at our institution. Adaptations to alternative warehouse solutions could be easily made without impact to the methods described in this paper.

Client-side functionality is developed using standard web technologies including HTML, CSS, and JavaScript technologies. The Dojo toolkit is used for user interface widgets and D3.js is used for our custom SVG-based visualizations.

As users interact with the query interface to add event constraints to an episode specification, type-ahead find is used to constrain the selections to only the event types present within the data. This allows users to quickly see what event types are available in a given dataset without deep prior knowledge of the data source.

5. Example patterns and trends

Our method allows users to perform a wide range of ad hoc analysis tasks. To demonstrate these capabilities, we applied our prototype implementation to a de-identified data set containing longitudinal diagnosis information for a population of over 32,000 cardiology patients from a United States-based care provider. Available data for these patients includes demographics, labs, medications, and diagnoses. We focus primarily on diagnoses for these use cases. The diagnosis information is encoded using the ICD-9 classification system. In addition, we use mappings that aggregate these relatively low-level ICD-9 codes into higher level categories using the Hierarchical Condition Category (HCC) system. For all examples, we use a minimum support threshold of twenty percent. The remainder of this section includes brief descriptions of three examples that show the variety of investigations that can be performed with our approach.

5.1. Example one: A single pattern over time

The example illustrated in Fig. 6 investigates a cohort of heart failure patients using ICD-9 codes. The user begins by defining an episode specification with no preconditions and three milestone events: dyslipidemia, angina, and heart failure. The user also specifies an outcome event of heart valve replacement. After entering this specification, the user clicks the "Analyze" button to submit the query to the system. In response, the system queries against the 32,000 patient database to retrieve a cohort of patients who match the episode constraints. Of the matching patients, eight percent were found to suffer from the outcome measure: a heart valve replacement diagnosis. As shown in the right sidebar of the screenshots in Fig. 6, the cohort is roughly half male with a majority over the age of 70. A large number of frequent patterns (minimum length of 2) were found in the overall episode, but all were either neutral or negative indicators (yellow or red circles in Fig. 6a). In Fig. 6(b), the user selects the

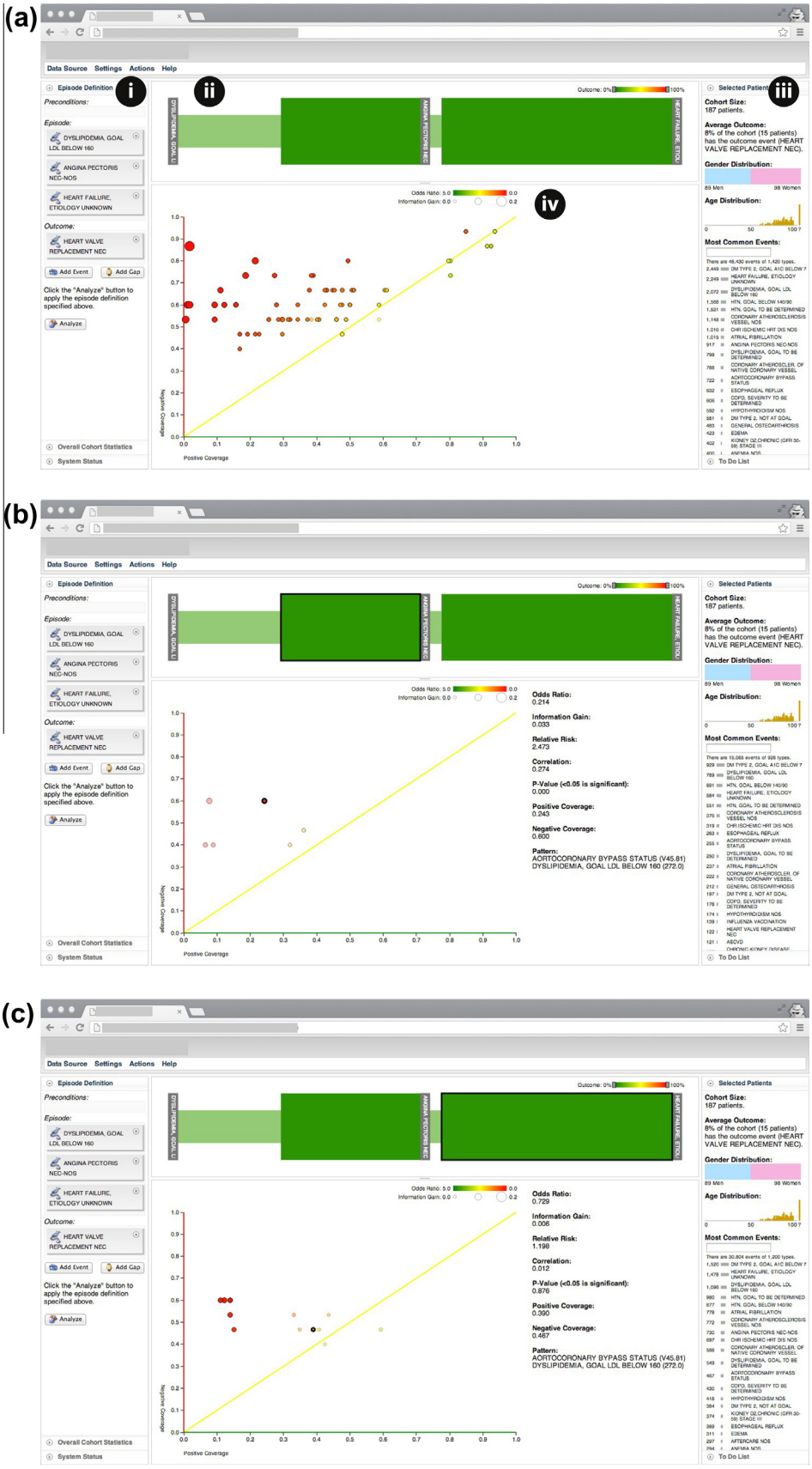

Fig. 6. As illustrated in panel (a), the user interface in our implementation has four main components: (i) the visual query panel, (ii) the milestone timeline, (iii) the cohort overview, and (iv) the pattern diagram. An analysis of heart failure patients shows several patterns detected in (a) the overall episode. Examining the first intermediate episode, the user finds (b) a statistically significant pattern associated with an eventual heart valve replacement. The pattern remains (c) later in the episode, but is no longer significant.

first intermediate episode using the timeline as shown by the black selection box. As is typical, focusing on specific intermediate episodes returns fewer frequent patterns. In this case, one interesting pattern (which includes an aortocoronary bypass event) is frequent in both the first and last intermediate episodes. The corresponding event pattern is selected in the pattern diagram in both Fig. 6b and c). However, comparing the location, size, and color of the selected circle makes it clear that the pattern is far more significant in terms of its association with outcome when observed toward the start of the episode. As shown in Fig. 6(c), the pattern remains common when focusing only on the period of time after the onset of angina. However, the pattern is nearly equal in its prevalence in both the positive and negative outcome groups. For this reason, its statistical significance nearly disappears during this latter portion of the episode.

One interesting observation from this example is that the majority of patterns highlighted by the system are associated with poor outcomes (i.e. more red circles, located in the top and left section of the pattern diagram). This is a common occurrence and can also be observed in the two other examples included below. This is a reflection, we believe, of the basic fact that sicker patients tend to have larger numbers of medical events in their record including increased numbers of diagnoses, procedures, and medications. Meanwhile, healthier patients tend to have less recorded medical activity within their electronic medical data.

5.2. Example two: All patterns over time

The example illustrated in Fig. 7 investigates a cohort of hypothyroidism patients using ICD-9 codes. The user begins by authoring an episode specification that has no preconditions and four milestone events: obesity, hypertension, type-2 diabetes, and hypothyroidism. The user also enters an outcome event of anemia. Using this specification, the user begins her analysis of anemia in obese patients by clicking the analyze button. The system performs a query to retrieve the matching cohort of patients and finds, as shown in the right sidebar of the screenshots in Fig. 7, that in 11.6% of the cohort has developed anemia after being diagnoses with hypothyroidism. Patterns were computed using a minimum length threshold of two. In terms of demographics, the system shows that the cohort is mostly women, and largely over the age of 50. As illustrated in the progression from Fig. 7(a–c), the user discovers an interesting trend. In the first panel, the user selects the first intermediate episode and sees only a handful of frequent patterns (seven patterns represented by the seven circles in the pattern diagram). In Fig. 7(b), the user selects the second intermediate episode to discover a larger number of frequent patterns. However, these patterns still had only a weak correlation to outcome. This is evident in the way the circles cluster along the diagonal of the pattern diagram. It is only when the user selects the last intermediate episode in Fig. 7(c) that we see strong predictive indicators for anemia. This is evident from the much greater number of patterns and the greater odds ratios as shown by the large circles. This example shows that for some disease progressions, the emergence of significant patterns can happen at different stages. In this case, there are few statistically significant patterns to be observed until late in the episode.

5.3. Example three: Comparing two patterns

The example illustrated in Fig. 8 investigates a cohort of hypertensive patients using Hierarchical Condition Category (HCC) data. Unlike ICD-9 data, which is very fine grained, HCC provides a higher-level categorization of diagnoses to group related diagnoses into a smaller number of distinct event types. The user begins by specifying an episode with no preconditions and four episode milestones: hypertension, hypertensive heart disease, angina, and finally heart infection/inflammation. The outcome measure specified by the user is cardio-respiratory failure and shock. This episode is processed by the system that returns a cohort of patients who match the specified disease progression. The returned data shows that just over 7% having negative outcomes. A large number of patterns (with minimum length of 1) are found in the overall sequence, including the negatively associated arrhythmias highlighted in Fig. 8(a). The very same pattern is found in the first intermediate episode and, as shown in Fig. 8(b), the significance was even stronger than in the overall data. Specifically, the *p*-value for this pattern dropped from 0.034 in the overall episode to 0.002 in the first intermediate episode. The position of the pattern's circle in the pattern diagram shows that this increase in significance is due to the fact arrhythmias were quite rare during the first intermediate episode in patients with positive outcomes. This contrasts strongly with the nearly 56% occurrence rate for arrhythmias within the positive outcome patients when considering the entire episode.

A further analysis shows that while the patients suffering arrhythmias during the first intermediate episode were among those most likely to have a negative outcome, there was another subgroup that had much better outcomes than average. As shown in Fig. 8(c), a group of patients who experienced endocrine/metabolic disorders during the first intermediate episode had much lower rates of cardio-respiratory failure or shock. This is illustrated by the green circle that is selected in the pattern diagram.

Comparing *p*-values for these two patterns, we can see that only the negative pattern (arrhythmias) is statistically significant. In contrast, the encocrine/metaboloic disorder pattern had a *p*-value of 0.094, well above the common 0.05 threshold for significance. However, it is potentially a hypothesis to target with additional investigation.

6. Discussion

As demonstrated by the use cases outlined above, the prototype system we have developed using the proposed methods provides an interactive visual environment for the exploration and analysis of temporal medical event data. The method is specifically focused on analyzing collections of temporal event sequences, mining for patterns of events that are strongly associated with outcome, and visualizing the detected patterns.

Based on feedback from users who have experimented with our prototype, there is value in our approach for both confirming expected associations and discovering unexpected patterns within the data. Users found the speed and flexibility of the system helpful, allowing them to quickly and simply perform data experiments that would otherwise take significant effort. While users were interested in the results, however, they also felt that insights discovered with the tool required more validation. The system was deemed most useful for the generation of hypotheses to which more thorough analysis techniques could be applied. It was also interesting to users to find patterns (multiple sequential events) associated with better or worse outcomes rather than the typical individual events that come from more traditional analysis methods.

Beyond speed, flexibility, and the detection of patterns, there are other advantages of our method. In particular, as described in the related work section, prior attempts at visualizing event sequences using temporal pathways typically handle only small numbers of event types (5–20 distinct types). Such an approach requires *a priori* event selection. In contrast, the method described in this paper is effective for extremely large numbers of event types (10,000+). The proposed approach succeeds despite these large

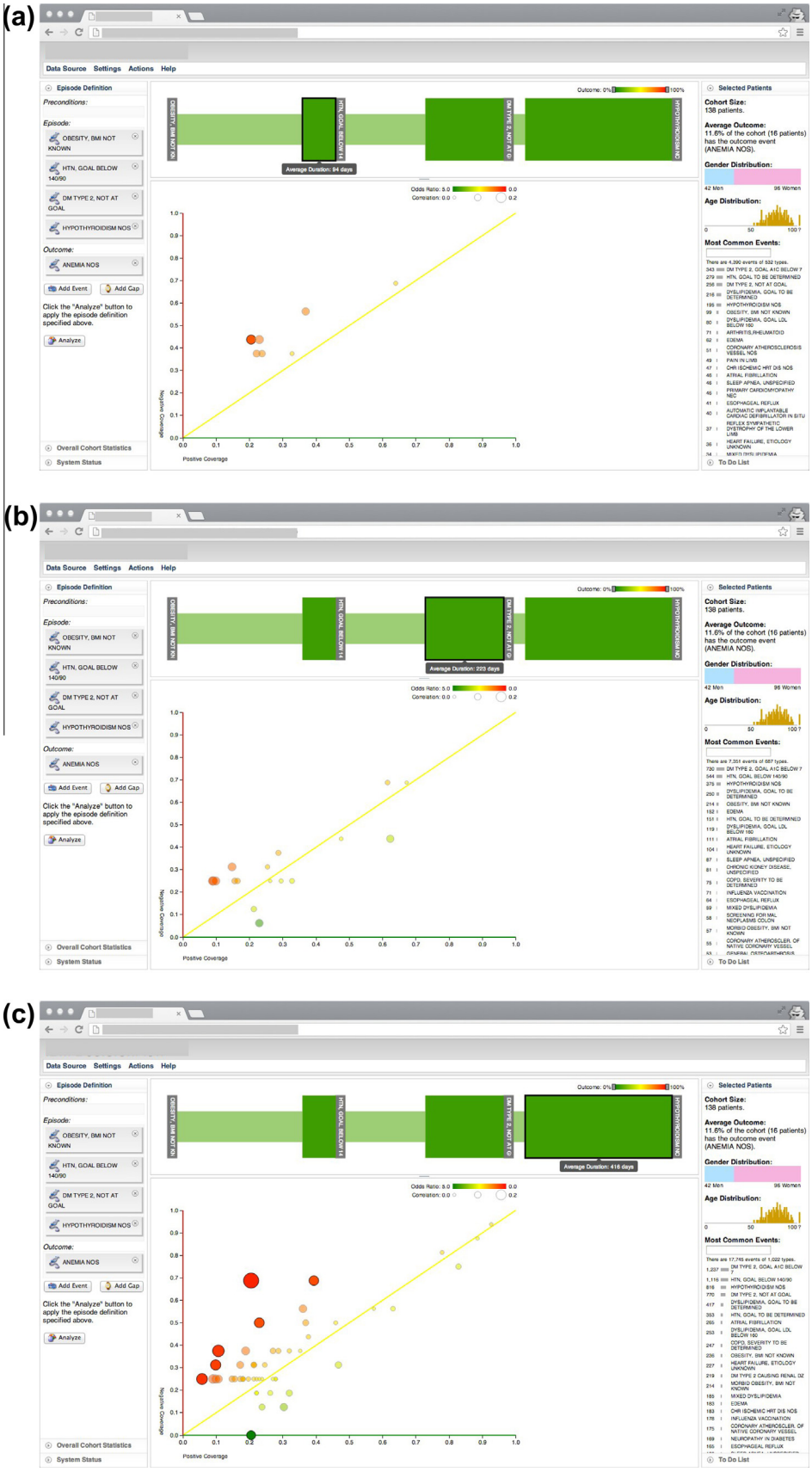

Fig. 7. An analysis of a hypothyroidism cohort containing mostly older women. (a) At the start, only a few patterns are found and all have negative associations. (b) The next intermediate episode shows more patterns, including some that are associated with better outcomes. (c) In the latter stages of the episode, event patterns become more numerous and significant.

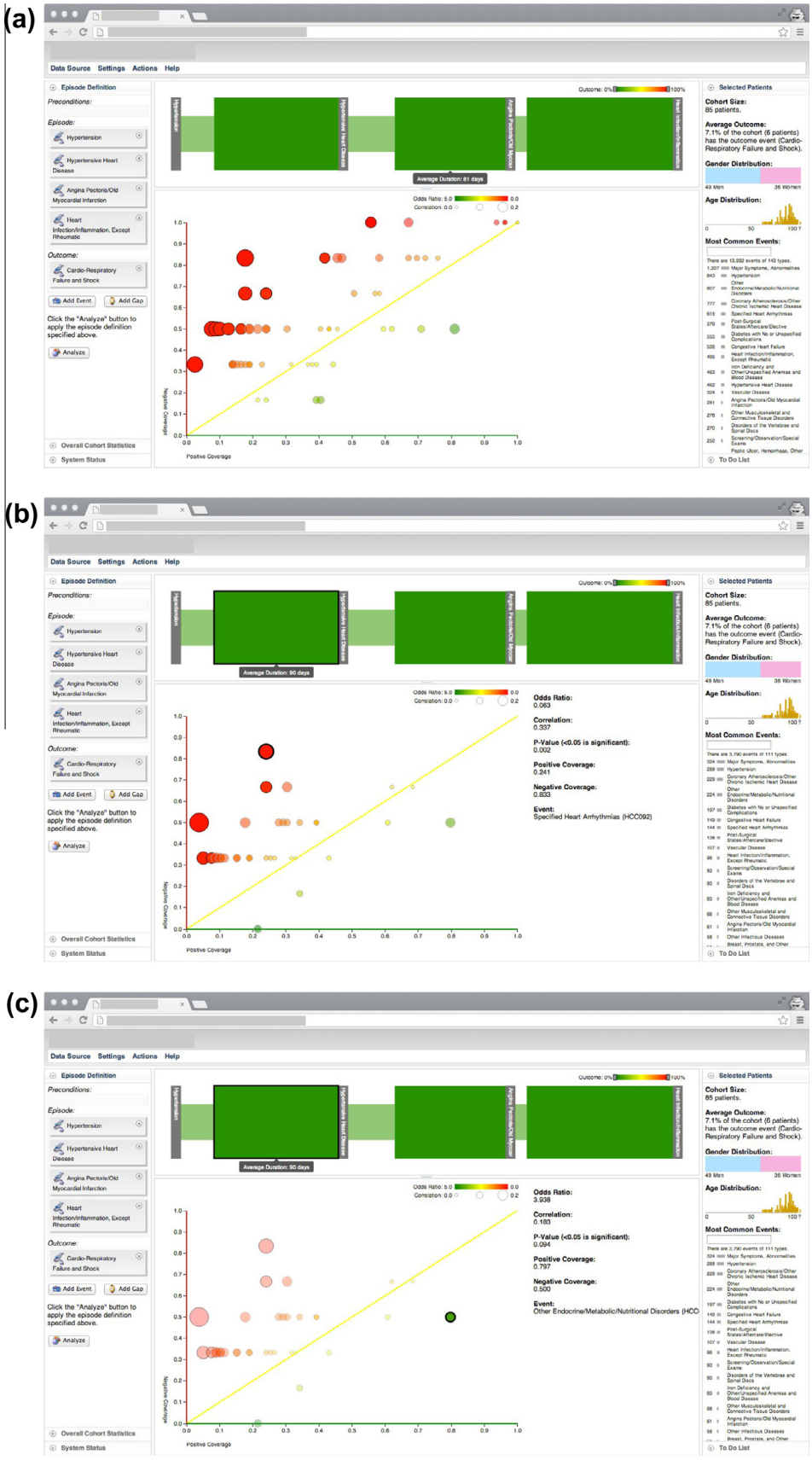

Fig. 8. An analysis of hypertension patients with additional heart-related diagnoses. (a) Overall, arrhythmias were significantly correlated with cardio-respiratory failure. (b) However, the significance was strongest early in the episode. (c) Meanwhile, endocrine/metabolic disorders were the strongest positive indicator in the first intermediate episode.

numbers of event types for two reasons. First, a two-view visualization space is used so that the temporal timeline is only used to directly visualize the milestone events. A more scalable Pattern Diagram is then used to show the more detailed event information. Selection in the timeline retains the users' abilities to detect temporal changes despite the omission of intermediate events from direct inclusion in the temporal timeline. Second, the FPM module is used in conjunction with the above to both (a) identify patterns that most impact outcome and (b) filter out less significant event types and patterns based on rarity or lack of statistical association with outcome.

Of course, significant limitations remain in the proposed approach and should be the subject for future research. First, and perhaps most significant, the method assumes linear episode progression. Yet in practice, pathways within a group of patients often branch and have cycles. Such artifacts are not captured in the current methodology unless they manifest themselves as linear, exact-match patterns. Related to this limitation is the fact that our current approach performs strict pattern detection that is not robust to minor differences between patients. For example, two patients may undergo identical treatment plans that are coded differently in the electronic data. We can partially overcome this fragmentation by using aggregate coding systems such as HCC to group related code, but more sophisticated pattern mining algorithms that can detect "similar" sequences rather than identical sequences would be highly useful.

7. Conclusion

Patients' medical conditions can evolve over time in complex and seemingly unpredictable ways. To help investigators better understand how variations in sequences of events can impact medical outcomes, we have developed an exploratory visual analytics system for clinical episode analysis that combines an easy-to-use graphical query interface, powerful event pattern mining techniques, and interactive visualization techniques. We described the key aspects of our methodology, which is designed to overcome some of the more restrictive limitations of prior work on event-based mining and visualization. We also reviewed details of our prototype implementation, and presented use cases using real-world datasets to demonstrate the power of our approach. While our initial results are promising, it is clear that additional research is required. In future work, we plan to conduct more in-depth user studies and report on the significant clinical findings that our users uncover. Finally, we will explore extensions to our system that enable more advanced types of episode analysis tasks.

Funding

This work was funded entirely by IBM. In addition, all authors are IBM employees and received a salary from the IBM Corporation.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.jbi.2014.01.007.

References

[1] Allen JF. Towards a general theory of action and time. Artificial Intell 1984;23(2):123–54.
[2] Ayres J, et al. Sequential PAttern mining using a bitmap representation. In: Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining; 2002. p. 429–35.
[3] Fails J, et al. A visual interface for multivariate temporal data: finding patterns of events across multiple histories. In: IEEE symposium on visual analytics science and technology; 2006. p. 167–74.
[4] Gotz D, et al. ICDA: a platform for intelligent care delivery analytics. In: AMIA annual symposium proceedings; 2012.
[5] Gotz D, Wongsuphasawat K. Interactive intervention analysis. In: AMIA annual symposium proceedings; 2012.
[6] Kosara R, Miksch S. Visualizing complex notions of time. Studies Health Technol Inform 2001;1(2001):211–5.
[7] Kullback S, Leibler RA. On information and sufficiency. Ann Math Stat 1951;22(1):79–86.
[8] Kurosaki M et al. Data mining model using simple and readily available factors could identify patients at high risk for hepatocellular carcinoma in chronic hepatitis C. J Hepatol 2012;56(3):602–8.
[9] McAullay D, et al. A delivery framework for health data mining and analytics. In: Proceedings of the twenty-eighth australasian conference on computer science – vol. 38. Darlinghurst, Australia, Australia; 2005. p. 381–87.
[10] McKee PA et al. The natural history of congestive heart failure: the Framingham study. New Engl J Med 1971;258(26):1441–6.
[11] Mitsa T. Temporal data mining. Chapman & Hall/CRC; 2010.
[12] Mörchen F, Ultsch A. Efficient mining of understandable patterns from multivariate interval time series. Data mining and knowledge discovery 2007;15(2):181–215.
[13] Moskovitch R, et al. Classification of ICU Patients via temporal abstraction and temporal patterns mining. Verona, Italy; 2009.
[14] Moskovitch R, Shahar Y. Medical temporal-knowledge discovery via temporal abstraction. In: AMIA annual symposium proceedings; 2009. p. 452–56.
[15] Murphy SN et al. Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). J Am Med Inform Assoc 2010;17(2):124–30.
[16] Norén G et al. Temporal pattern discovery in longitudinal electronic patient records. Data Mining Knowledge Discov 2010;20(3):361–87.
[17] Ohno-Machado L et al. iDASH: integrating data for analysis, anonymization, and sharing. J Am Med Inform Assoc 2011.
[18] Pei J, et al. PrefixSpan: mining sequential patterns by prefix-projected growth. In: Proceedings of international conference on data engineering; 2001. p. 215–24.
[19] Plaisant C, et al. LifeLines: using visualization to enhance navigation and analysis of patient records. In: Proceedings of the AMIA symposium; 1998. p. 76–80.
[20] Plaisant C, et al. Searching electronic health records for temporal patterns in patient histories: a case study with microsoft amalga. In: AMIA annual symposium proceedings; 2008. p. 601–05.
[21] Rea S et al. Building a robust, scalable and standards-driven infrastructure for secondary use of EHR data: the SHARPn project. J Biomed Inform 2012.
[22] Rosling H, Zhang Z. Health advocacy with Gapminder animated statistics. J Epidemiol Global Health 2011;1(1):11–4.
[23] Shahar Y. A framework for knowledge-based temporal abstraction. Artificial Intell 1997;90(1):79–133.
[24] Tang H et al. Predicting three-year kidney graft survival in recipients with systemic lupus erythematosus. ASAIO J (Am Soc Artificial Internal Org 1992) 2011;57(4):300–9.
[25] Toma T et al. Discovery and inclusion of SOFA score episodes in mortality prediction. J Biomed Inform 2007;40(6):649–60.
[26] Toma T et al. Learning predictive models that use pattern discoveryÑA bootstrap evaluative approach applied in organ functioning sequences. J Biomed Inform 2010;43(4):578–86.
[27] Wang F et al. A framework for mining signatures from event sequences and its applications in healthcare data. IEEE Trans Pattern Anal Machine Intell 2013;35(2):272–85.
[28] Wang TD et al. Temporal summaries: supporting temporal categorical searching, aggregation and comparison. IEEE Trans Visual Comput Graph 2009;15(6):1049–56.
[29] Wongsuphasawat K, et al. LifeFlow: visualizing an overview of event sequences. In: Proceedings of the 2011 annual conference on Human factors in computing systems. New York, NY, USA; 2011. p. 1747–56.
[30] Wongsuphasawat K, Gotz D. Exploring flow, factors, and outcomes of temporal event sequences with the outflow visualization. IEEE Trans Visual Comput Graph 2012;18(12):2659–68.
[31] Wongsuphasawat K, Gotz D. Outflow: visualizing patient flow by symptoms and outcome. In: IEEE VisWeek workshop on visual analytics in healthcare. Providence, Rhode Island, USA; 2011.
[32] K. Wongsuphasawat, B. Shneiderman, Finding Comparable Temporal Categorical Records: A Similarity Measure with an Interactive Visualization. (2009).
[33] Zaki MJ. SPADE: an efficient algorithm for mining frequent sequences. Machine Learning J 2001;42(1/2):31–60.
[34] Streptomycin treatment of pulmonary tuberculosis. British Med J 1948;2(4582): 769–82.