Experiences with Mining Temporal Event Sequences from Electronic Medical Records: Initial Successes and Some Challenges

Debprakash Patnaik[1]
patnaik@vt.edu

Patrick Butler[1]
pabutler@vt.edu

Naren Ramakrishnan[1]
naren@cs.vt.edu

Laxmi Parida[2]
parida@us.ibm.com

Benjamin J. Keller[3]
bkeller@emich.edu

David A. Hanauer, MD[4]
hanauer@umich.edu

[1]Department of Computer Science and Discovery Analytics Center, Virginia Tech, VA 24061
[2]IBM T.J. Watson Research Center, Yorktown Heights, NY 10598
[3]Department of Computer Science, Eastern Michigan University, Ypsilanti, MI 48197
[4]Department of Pediatrics and Communicable Diseases, University of Michigan, Ann Arbor, MI 48109

ABSTRACT

The standardization and wider use of electronic medical records (EMR) creates opportunities for better understanding patterns of illness and care within and across medical systems. Our interest is in the temporal history of event codes embedded in patients' records, specifically investigating frequently occurring sequences of event codes across patients. In studying data from more than 1.6 million patient histories at the University of Michigan Health system we quickly realized that frequent sequences, while providing one level of data reduction, still constitute a serious analytical challenge as many involve alternate serializations of the same sets of codes. To further analyze these sequences, we designed an approach where a partial order is mined from frequent sequences of codes. We demonstrate an EMR mining system called EMRView that enables exploration of the precedence relationships to quickly identify and visualize partial order information encoded in key classes of patients. We demonstrate some important nuggets learned through our approach and also outline key challenges for future research based on our experiences.

Categories and Subject Descriptors

J.3 [**Computer Applications**]: Life and Medical sciences—*Medical information systems*

General Terms

Algorithms, Design, Management

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
KDD'11, August 21–24, 2011, San Diego, California, USA.
Copyright 2011 ACM 978-1-4503-0813-7/11/08 ...$10.00.

Keywords

Medical informatics, temporal data mining, partial orders.

1. INTRODUCTION

The increased use of electronic medical records (EMRs) to capture standardized patient information creates an opportunity to better understand both patterns of illness among similar patients, as well as patterns of care within and across medical systems. While the first allows us to identify "fingerprints" of clinical symptoms that may help us capture the progression toward catastrophic clinical transitions, the second can help us understand how diagnoses and procedures are applied within and across different clinical settings.

Two primary categories of information embedded in an EMR are referred to as ICD and CPT codes. ICD 9 (International Classification of Diseases and Related Health Problems, v9) is a popular classification system used in health care which is also heavily used for a wide variety of research activities [8]. This system is primarily intended to code signs, symptoms, injuries, diseases, and conditions. CPT (Current Procedural Terminology) refers to a similar categorization of medical procedures performed on a patient. CPT codes are 5 digit numeric codes, which are published by the American Medical Association. The purpose of this coding system is to provide uniform language that accurately describes medical, surgical, and diagnostic services (including radiology, anesthesiology, and evaluation/management services of physicians, hospitals, and other health care providers). There are about 20,000 distinct ICD 9 codes and about 10,000 CPT codes in use today. Thus an electronic medical record consists, in its most basic sense, of an interleaved sequence of diagnostic (ICD) and procedure (PCT) codes assigned to the patient every time he/she received care along with other associated test reports and doctor's notes.

An EMR is intrinsically temporal in nature yet research on temporal data mining in medical records is scarce. This work seeks to fill this void. Our goal is to extract clinically relevant *sequences* of event codes embedded in the patient histories of more than 1.6 million subjects of the University of Michigan's health system. A straightforward implementation of sequential mining approaches [1, 21] does help provide one level of data reduction, but the resulting sequences are still too many for human consumption because they are permutations or nearly permutations of each other representing different serializations of the same sets of codes. For instance, it is more likely to see a diagnosis of cardiac ischemia before an aortic coronary bypass, but there are patients for which these occur in the opposite order. It is also not satisfactory to simply lump together codes on the evidence of multiple permutations, because not all sequences are equally likely. We therefore present an approach to derive a partial order from frequent sequences of codes, along with a tool called EMRView that supports exploration of these orders based on a physician's selections of supporting sequences.

2. INITIAL EXPLORATION

After obtaining approval from the Institutional Review Board at the University of Michigan Medical School, we organized a dataset of information about 1.6 million patients in the health system. The actual medical records contained about 100 million time stamped ICD 9 and CPT 4 codes and also other categories of data such as doctor's notes, test results, prescriptions and x-ray images. There are two main concerns in using these other categories of data. The first is to ensure the privacy of the patients concerned. With freeform text this is a very hard problem to address. The second concern pertains to transcribing the handwritten notes into electronic form before they can be analyzed. Hence we began our initial focus on the encoded procedure and diagnostic codes.

First, we replaced patient medical record numbers with artificial research identifiers not tied to the medical record. Time intervals were then used to replace specific dates (thus, resetting the start of a patient's record to time zero). See Table 1 for an example. These codes represented over 10 years of clinical encounters in our electronic medical record system. Data was securely transferred to Virginia Tech for analysis. Specific details of the data are shown in Table 2. The range of the number of entries/codes per patient varies from 1 to 10K, i.e., there were many patients with few isolated visits and also patients who have received their entire lifetime of care from the particular healthcare provider. These and other power law behaviors are shown in Fig. 1. Some patient records extend over 20 years! In terms of the diagnostic and procedure codes, there are over 40 million and 38 million entries respectively of each type of code in the data.

After a preliminary analysis, we decided to focus exclusively on the diagnostic (ICD) codes. The reason for this decision is that most of the procedure codes (CPT) appear to have been initiated in response to a diagnosis and hence actually served as proxy for conditions. This aided in simplifying the search for patterns downstream although we run the risk of missing infrequent classes of relationships (e.g., the administration of a procedure causing a complication down the road).

Let us denote the medical record of the i^{th} patient as an ordered sequence of diagnostic codes:

$$S_i = \langle (E_1, t_1), \ldots, (E_j, t_j), \ldots, (E_{|S_i|}, t_{|S_i|}) \rangle$$

where E_j is the j^{th} diagnostic code and t_j is the associated time stamp. An entire EMR database consisting of several patient records is denoted by $\mathcal{D} = \{S_1, \ldots, S_{|\mathcal{D}|}\}$.

Table 1: Example EMR.

Patient ID	Code	Type	Desc.	Timestamp (in days)
149970	99243	CPT	Office consultation	0
149970	145.9	ICD	Malignant neoplasm of mouth, unspecified.	0
149970	88321	CPT	Microslide consultation.	1
149970	792.9	ICD	Other nonspecific abnormal findings in body substances.	1

Table 2: Characteristics of the EMR database.

Property	Value
Number of patients	1,620,681
Number of diagnostic (ICD) codes	41,186,511
Number of procedure (CPT) codes	38,942,605
Max. number of codes in a record	10,430
Min. number of codes in a record	1
Max. span of a record in days	8202 days≈ 22.5 years
Min. span of a record in days	1

One interesting characteristic of the data was observed when we plotted the distribution of time differences between consecutive diagnostic codes across all records (see Fig. 2). Note the different modes at multiples of 7 days, an artifact of how followup visits to the health care provider are typically scheduled.

Finally, just as stopword elimination is employed in information retrieval, we conducted 'stopcode elimination' to remove diagnostic codes that are too frequently used. Specifically, codes that surfaced in more than 0.49% of patient records were eliminated (the most frequent code surfaced in 9% of patients).

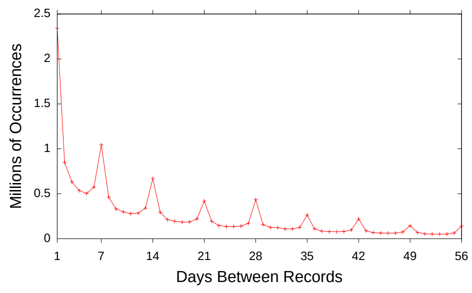
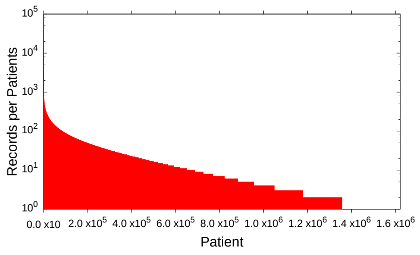


Figure 2: Distribution of interarrival times

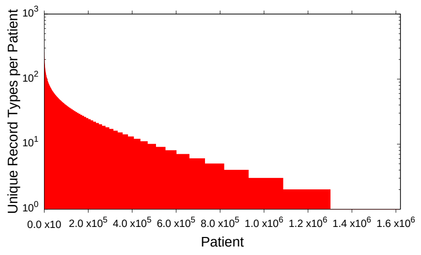
3. METHOD OVERVIEW

Figure 3 provides an overview of our proposed methodology for unearthing partial orders from EMR data. There are three broad stages: i) mining parallel episodes, ii) tracking serial extensions, and iii) learning partial orders.

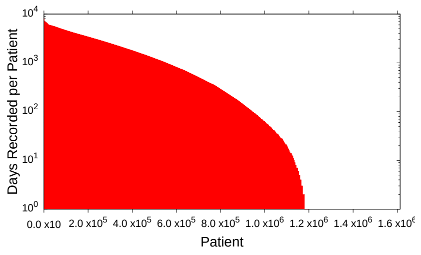
Mining parallel episodes: Parallel episodes are very similar to itemsets but take temporal information into account in the form of expiry constraints. We present a counting algorithm in Section 4.1 that discovers parallel episodes which are frequent above a given threshold. It is well known that patterns with support higher than a threshold may not always be interesting because certain high frequency items can randomly co-occur. Hence, we apply the maximum entropy



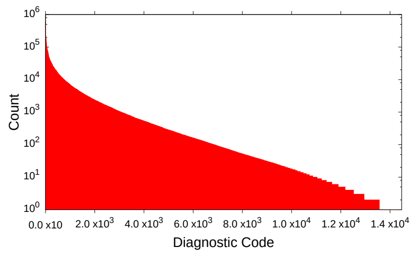
(a) No. of ICD codes per record



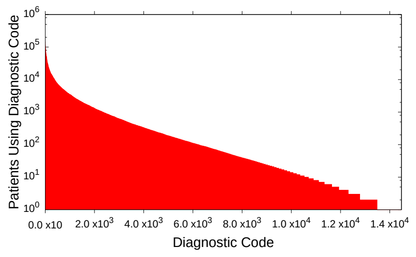
(b) No. of distinct ICD codes per record



(c) Span of record in days



(d) No. of occurrences of an ICD code



(e) No. of patients assigned an ICD code

Figure 1: Plots of various distributions seen in the real EMR data.

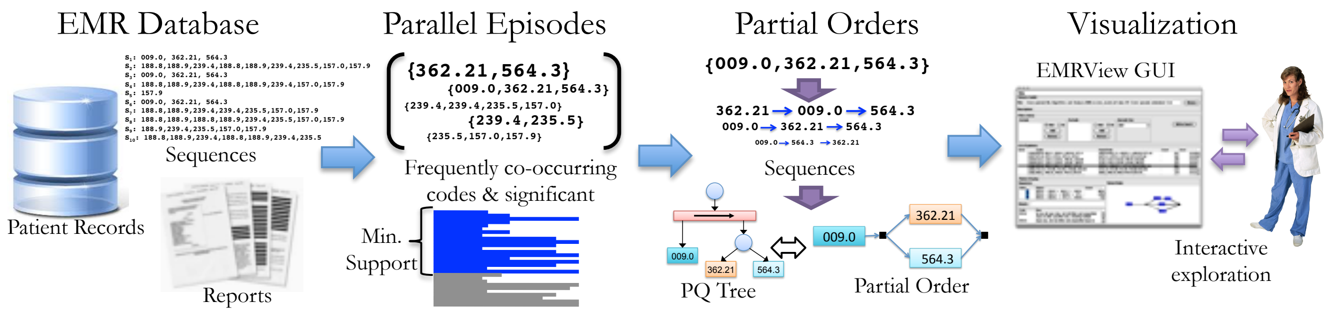

EMR Database
S_1: 009.0, 362.21, 564.3
S_2: 188.8,188.9,239.4,188.8,188.9,239.4,235.5,157.0,157.9
S_3: 009.0, 362.21, 564.3
S_4: 188.8,188.9,239.4,188.8,188.9,239.4,157.0,157.9
S_5: 157.9
S_6: 009.0, 362.21, 564.3
S_7: 188.8,188.9,239.4,239.4,235.5,157.0,157.9
S_8: 188.8,188.9,188.8,188.9,239.4,235.5,157.0,157.9
S_9: 188.9,239.4,235.5,157.0,157.9
S_{10}: 188.8,188.9,239.4,188.8,188.9,239.4,235.5
Sequences
Patient Records
Reports
Parallel Episodes
{362.21,564.3}
{009.0,362.21,564.3}
{239.4,239.4,235.5,157.0}
{239.4,235.5}
{235.5,157.0,157.9}
Frequently co-occurring codes & significant
Min. Support
Partial Orders
{009.0,362.21,564.3}
362.21 → 009.0 → 564.3
009.0 → 362.21 → 564.3
009.0 → 564.3 → 362.21
Sequences
009.0
362.21
564.3
PQ Tree
009.0
362.21
564.3
Partial Order
Visualization
EMRView GUI
Interactive exploration

Figure 3: Overview of the proposed EMR analysis methodology.

principle[14] to predict the support of an episode and examine the actual support to impart a score of significance. This score is used to order episodes and is carried out offline.

Tracking serial extensions: After mining a set of frequent parallel episodes, we find order information embedded in these frequently co-occurring sets of diagnostic codes. The intuition behind searching for order is to unearth potential evidence (or lack thereof) for sequentiality in the data. For a n sized frequent parallel episode, there could potentially be $n!$ permutations exhibited in the data.

Learning partial orders: The set of all serial extensions determined in the previous step is compacted into a partial order using a special PQ tree algorithm. PQ trees[4] enable an expressive class of partial orders to be learnt and the user has the flexibility to interactively aid the search for partial orders by selecting the set of extensions to be modeled. This part of the procedure is thus interactive so that the medical professional can iteratively generalize and specialize partial orders in order to arrive at an understanding of the underlying orders.

4. FREQUENT EPISODE MINING

In this section we briefly review the frequent episode mining framework and related terminology.

Definition 4.1 (General Episode). *An episode α is defined as a tuple $\{V_\alpha, \le_\alpha\}$, where $V_\alpha \subseteq \mathcal{A}$, $\mathcal{A}$ is the set all symbols (i.e. diagnostic codes). The relation $\le_\alpha$ defines an ordering of the codes in V_α. A general episode is a partial order over V_α.*

For example, consider an episode α given by $V_\alpha = \{A, B, C\}$ with $A \le_\alpha B$ and $A \le_\alpha C$. This defines an episode where A must occur before B and C but (B, C) can occur in any order and can be denoted as $A \to (BC)$.

Definition 4.2 (Serial Episode). *A serial episode is an ordered sequence of symbols/codes. It represents a total order where for any pair $a_i, a_j \in V_\alpha$ and $a_i \neq a_j$, either $(a_i \le_\alpha a_j)$ or $(a_j \le_\alpha a_i)$ holds.*

An example of a serial episode is $A \to B \to C$ where the relation $\le_\alpha$ is such that $A \le_\alpha B$, $A \le_\alpha C$ and $B \le_\alpha C$.

Definition 4.3 (Parallel Episode). *A parallel episode is an unordered set of symbols/codes i.e. $\le_v = \phi$.*

A sequence S supports an episode α, if $V_\alpha \subseteq \{A_i : A_i \in S\}$ and for no pair A_j preceding A_i in S, $A_i \le_\alpha A_j$ holds. In other words, the order of the symbols in S does not violate the precedence order encoded by $\le_\alpha$. The support of α is the fraction of sequences in D that support it. In addition, an expiry constraint T_X can be specified for episodes which requires that the symbols in S that constitute an occurrence of the episode occur no further than T_X time units apart from each other.

Example 4.4. *Consider an EMR dataset of 4 sequences (Fig. 4). The highlighted codes constitute occurrences of the parallel episode (ABC) (in S_1,S_2,S_3 and S_4), serial episode $A \rightarrow B \rightarrow C$ (in S_2 and S_3), and the partial order $A \rightarrow (BC)$. Later we shall use this example to illustrate our method.*

```
S1: W,U,C,I,B,C,K,A,Y,E,K,J,H,A
S2: K,J,A,D,K,E,W,K,B,C,R,H
S3: Q,T,B,J,A,C,O,J,B,A,C,K,F,N,F
S4: L,M,A,N,C,V,J,H,B,I,U,W,G,B,G,G
```

Figure 4: Database of four sequences S_1, S_2, S_3 and S_4. The time-stamps are not shown. Note that $support(ABC) = 4/4$**,** $support(A \rightarrow B \rightarrow C) = 2/4$ **and** $support(A \rightarrow (BC) = 3/4$**.**

4.1 Parallel Episodes with Expiry

Frequent episode mining follows the Apriori-style itemset mining algorithm: the set of n-size frequent episodes are used to generate the $(n+1)$-size candidate episodes. The counting algorithm then determines the support of each candidate episode over the database of ordered sequences D. Typically, frequent episode mining is carried out over one long sequence of time-stamped events. In a database of ordered sequences, episodes essentially reduce to sequential patterns [1] so that support is defined as the fraction of sequences that have the pattern as a subsequence. Our expiry constraint gives us an additional handle to control the explosion in candidates besides the support threshold.

Algorithm 1 Count parallel episodes of size-k.

```
Input: Set of candidate $k$ size parallel episodes $C$, Sequence
    database $D$, Expiry time constraint $T_X$, Min support $\theta$.
Output: Set of frequent parallel of episodes with counts.
 1: $\mathcal{A}$ = Set of all codes in episodes in $C$
 2: Initialize $waits[A] = \phi, \forall A \in \mathcal{A}$
 3: for all $\alpha \in C$ do
 4:    Set $count[\alpha] = 0$
 5:    Set $wait\_count[\alpha, A] = 1$ and $T[\alpha, A] = \phi, \quad \forall A \in \alpha$
 6:    Add $\alpha$ to $waits[A]$
 7: for all $S \in D$ do
 8:    for all $(E, t) \in S$ do
 9:       for all $\alpha \in waits[E]$ do
10:          Update $wait\_count[\alpha, E] = 0$ and $T[\alpha, E] = t$
11:          for all $A \in \alpha$ do
12:             if $(t - T[\alpha, A] > T_X)$ or $sid(A)! = S.id$ then
13:                $wait\_count[\alpha, E] = 1$
14:          if $\left(\sum_{A \in \alpha} wait\_count[\alpha, A]\right) = 0$ then
15:             $count[\alpha] = count[\alpha] + 1$
16:             Reset $wait\_count(\alpha, A) = 1, \forall A \in \alpha, \forall \alpha \in C$
17: return $\{\alpha, count(\alpha) : count(\alpha) \geq \theta|D|\}$
```

Algorithm 1 sketches the counting algorithm for parallel episodes with expiry time constraint. The algorithm makes one pass of the database to determine the support of all episodes in the set of candidates. It uses a hash-map data structure to efficiently reference episodes that need to be updated for each code seen in a sequence. The data structure $wait\_count[.]$ tracks the codes that are remaining to complete an occurrence and $T[.]$ stores the latest time stamp of each code. Note that tracking the latest time-stamp ensures that the span of the resulting occurrence is minimum. Hence we correctly track all episode occurrences that satisfy the expiry constraint.

4.2 Significance of parallel episodes

We utilize the maximum entropy formulation introduced in [33] to assess significance of our episodes. Consider an n-size parallel episode $\alpha = \{E_1, \ldots, E_n\}$, where E_i's are the diagnostic codes. Let Ω be the sample space of all n-size binary vectors and has a distribution $p : \Omega \rightarrow [0, 1]$. In the context of the database D, $p(\omega)$ gives the fraction of the sequences which support the binary vector ω where each 0/1 represents the presence or absence of the corresponding code E_i in a sequence. We compute the empirical distribution q_α over ω using the support of α and those of its sub-episodes. Example 4.5 illustrates the use of inclusion-exclusion principle to obtain q_α.

Example 4.5 (Empirical distribution). *Consider a 3-size parallel episode ABC. If ABC is frequent, all the subepisode supports are known. Table 3 illustrates the computation of the empirical distribution q_{ABC} using the inclusion-exclusion principle over support (σ) of subepisodes.*

Table 3: Computing the empirical distribution q_α

A B C	Distribution q_{ABC}
0 0 0	$\sigma(\phi) - \sigma(A) - \sigma(B) + \sigma(AB) - \sigma(C) + \sigma(AC) + \sigma(BC) - \sigma(ABC)$
0 0 1	$\sigma(C) - \sigma(AC) - \sigma(BC) + \sigma(ABC)$
0 1 0	$\sigma(B) - \sigma(AB) - \sigma(BC) + \sigma(ABC)$
0 1 1	$\sigma(BC) - \sigma(ABC)$
1 0 0	$\sigma(A) - \sigma(AB) - \sigma(AC) + \sigma(ABC)$
1 0 1	$\sigma(AC) - \sigma(ABC)$
1 1 0	$\sigma(AB) - \sigma(ABC)$
1 1 1	$\sigma(ABC)$

We estimate the distribution p for α using only the support of its sub-episodes. This is done by estimating the maximum entropy distribution p_α^{ME} (defined in Eq 1) that represents a log-linear model.

$$p_\alpha^{ME}(\omega) = \frac{1}{Z_\alpha} \exp\left(\sum \lambda_i f_i(\omega)\right) \tag{1}$$

Here each factor $f_i(w)$ is an indicator function:

$$f_i(w) = I(\omega \text{ covers } i^{th} \text{ subepisode of } \alpha)$$

We learn the maximum entropy model under the constraint that the expectations $\mathbb{E}\langle f_i \rangle = \langle$support of i^{th} proper subepisode of $\alpha\rangle$. We use the KL-divergence measure between the empirical distribution and the maximum entropy distribution estimated using only subepisodes to score the significance of α (Eq 2).

$$score(\alpha) = \sum_\omega q_\alpha(\omega) \log \frac{q_\alpha(\omega)}{p_\alpha^{ME}(\omega)} \tag{2}$$

A higher score implies that the support of the episode is more surprising given the subepisodes. This significance score is used to rank the discovered patterns.

4.3 Tracking sequential extensions

After mining the lattice of frequent parallel episodes, we make another pass through the data to record the number of times different linear (serial) extensions of each frequent parallel episode occur in the data. This is illustrated below for the sequences in Example 4.4.

Example 4.6 (Sequential Extensions). *Consider the parallel episode (ABC). There are $3! = 6$ possible permutations or serial extensions of this episode. Now consider the sequences shown in Example 4.4, where only 3 of the 6 permutations of (ABC) are seen in the data. The supports of these sequences is noted below.*

$A \to B \to C : 2/4$; $\quad A \to C \to B : 1/4$; $\quad B \to C \to A : 1/4$

5. LEARNING PARTIAL ORDERS

At this stage, we have mined sets of frequent episodes that are permutations of each other, e.g. $A \to B \to C$, $A \to C \to B$ and $B \to C \to A$. Since the parallel episode (ABC) and each of these serial episodes is known to be frequent in the data, we aim to find a partial order that describes these sequences (and as few other sequences as possible). A trivial solution is the already obtained null partial order: (ABC) which, besides covering the above three extensions, also covers $B \to A \to C$, $C \to A \to B$, and $C \to B \to A$. A better solution is the partial order $(A(BC))$ which describes only one other sequence in addition to above: $C \to B \to A$.

Mining partial orders is a difficult problem in general because the space of partial orders grows very quickly with alphabet size. As a result, researchers have proposed focusing on special cases of partial orders (e.g., series-parallel) to make the problem tractable [22]. Here, we present a PQ tree approach[17] for finding a recursive partial order over the set of frequent serial episodes over the same set of symbols.

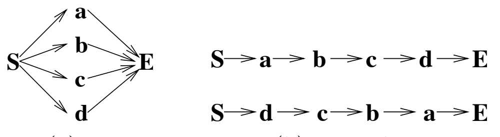
In the context of sequential data, a sequence S is *compatible* with a partial order α, if for no pair a_2 preceding a_1 in S, $a_1 \leq_\alpha a_2$ holds in α. In other words, the order of the elements in S does not violate the precedence order encoded by $\leq_\alpha$. A compatible S is also called an *extension* of $\leq_\alpha$. Sequence S is called a complete extension of $\leq_\alpha$ when S contains all the elements of V_α. In the episodes framework, only a complete extension is considered to support the episodes α. We shall define S' as the subsequence of S that is still a complete extension of α but contains only the elements of V_α. Subsequently, we shall refer to S' as the serial extension of α in the sequence S and denote it as $S' \in \alpha$.

A partial order can be represented by a directed graph where each edge (a_1, a_2) represents the relation $a_1 \leq_\alpha a_2$. Since the relation $\leq_\alpha$ is antisymmetric, it is easy to see that this graph is acyclic (i.e., a DAG). Further, the transitive reduction G'_α of the graph G_α also encodes the partial order α such that when $a_1 \leq_\alpha a_2$, there is a directed path from a_1 to a_2 in G'_α.

5.1 Maximal partial order problem

It is easy to see that S is a serial extension of the parallel episode α where $\leq_\alpha = \phi$ for any sequence S on the alphabet V_α. Also if $S \in \alpha(V_\alpha, \leq_\alpha)$, then $S \in \beta(V_\alpha, \leq_\beta)$ where $(\leq_\beta \subset \leq_\alpha)$. Thus it is natural to talk about a *maximal* partial order. We identify the following problem:

Problem 1. *Given a set $\mathcal{S}$ of sequences S_i, $1 \leq i \leq m$, each of length n defined over the same alphabet, find a maximal partial order α such that all $S_i \in \alpha$ and this does not hold for any partial order β with $V_\alpha = V_\beta$ and $(\leq_\beta \supset \leq_\alpha)$.*



(a) α_0 (b) α_1 and α_2

Figure 5: (a) One partial order, α_0 and (b) two partial orders, α_1, α_2 representing $s_1 = \langle a\,b\,c\,d \rangle$ and $s_2 = \langle d\,c\,b\,a \rangle$.

Here maximality implies that: given a pair (a_i, a_j), if a_i precedes d_j in *all* the given m sequences then $(a_i \leq_\alpha a_j)$ must hold. A straightforward approach to solving the problem is to collect all pairs (a_1, a_2) such that a_1 precedes a_2 in *all* sequences $s_i \in \mathcal{S}$ and then construct a (transitive reduction) DAG from these pairs.

5.2 Excess in partial orders

Consider $s_1 = \langle a\,b\,c\,d \rangle$ and $s_2 = \langle d\,c\,b\,a \rangle$. Notice that no pair of characters preserve the order in s_1 and s_2, hence the only partial order that s_1 and s_2 do not violate is shown in Figure 5(a). However, any permutation of the characters $\{a, b, c, d\}$ also does not violate this partial order.

Let $A(\alpha)$ be the set of all complete extensions $s \in \alpha$. Thus it is clear that the partial order captures some but not all the subtle ordering information in the data and the maximal partial order α of $\mathcal{S}$ is such that $A(\alpha) \supseteq \mathcal{S}$. The gap, $A(\alpha) \backslash \mathcal{S}$ is termed *excess*.

One way to handle excess in partial order is allowing for multiple partial orders, say α_1 and α_2 in Figure 5(b). Clearly, given S, there has to be a balance between the number of partial orders and the excess in the single maximal partial order α_0.

It is easy to see that given n distinct sequences, n partial orders (chains or total orders) give zero excess. Obviously this is not an acceptable solution, and a trade off between excess and the number of multiple partial orders needs to be made. However getting an algorithmic handle on this combinatorial problem, using excess, is non trivial. Other researchers have assumed further restrictions, e.g., Mannila and Meek [22] restrict their partial orders to have an MSVP (minimal vertex series-parallel) DAG. Their aim is to count the size of of $A(\alpha)$ (number of complete extensions of α) since they are looking for a probabilistic model of an underlying generator.

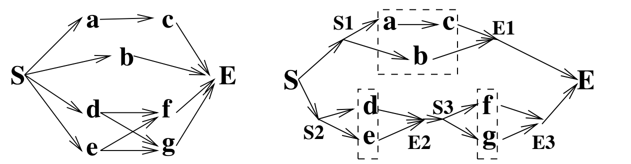
5.3 Handling excess with PQ structures

We illustrate this approach starting with an example.

Example 5.1. *Let $\mathcal{S} = \{\langle abcdegf \rangle, \langle bacedfg \rangle, \langle acbdefg \rangle, \langle edgfabc \rangle, \langle degfbac \rangle\}$. It is easy to see that the maximal partial order is α_0 shown in Figure 6. In α_1, the alphabet is augmented with $\{\mathbf{S_1}, \mathbf{E_1}, \mathbf{S_2}, \mathbf{E_2}, \mathbf{S_3}, \mathbf{E_3}\}$ to obtain a tighter description of $\mathcal{S}$.*

Here the solution to handling excess is to augment the alphabet V_α with $\mathcal{O}(|V_\alpha|)$ characters in the maximal partial order DAG, G_α. Let this new set be V'_α. All complete extension S' are now on $(V_\alpha \cup V'_\alpha)$ and can be easily converted to S on V_α by simply removing all the symbols $\mathbf{S}_k, \mathbf{E}_k \in V'_\alpha$.

The boxed elements in Figure 6 are clusters of symbols that *always* appear together, i.e., are uninterrupted in $\mathcal{S}$.



(a) Partial order, α_0 (b) Augmented partial order, α_1

Figure 6: α_1 is a specialization of α_0: If $S \in \alpha_1$, then $S \in \alpha_0$, but not *vice-versa*. If $S = \langle abdefcg \rangle$, then $S \in \alpha_0$ but $S \notin \alpha_1$.

Our scheme exploits this property to reduce excess. We use the systematic representation of these recursive clusters in reducing excess in partial orders. The recursive (order) notation of clusters is naturally encoded by a data structure called the PQ tree [4]. The equivalence of this order notation to a PQ structure, denoted by T_c, along with efficient algorithms is discussed in [17].

A PQ tree is a rooted tree whose internal nodes are of two types: P and Q. The children of a P node occur in no particular order while those of a Q node must appear either in a strict left to right or right to left order. We designate a P node by a circle and a Q node by a rectangle. The leaves of T_c are labeled bijectively by the elements of $\mathcal{S}$. The *frontier* of T_c, denoted by $Fr(T_c)$, is the permutation of the elements of $\mathcal{S}$ obtained by reading the labels of the leaves from left to right. Two PQ trees T_c and $T_{c'}$ are equivalent, denoted $T_c \equiv T_{c'}$, if one can be obtained from the other by applying a sequence of the following transformation rules: (1) Arbitrarily permute the children of a P node, and (2) Reverse the children of a Q node. $\mathcal{C}(T_c)$ is defined as follows: $\mathcal{C}(T_c) = \{Fr(T_{c'}) | T_{c'} \equiv T_c\}$. An *oriented PQ tree* is a tree where all the Q nodes are oriented similarly (in our case left-to-right).

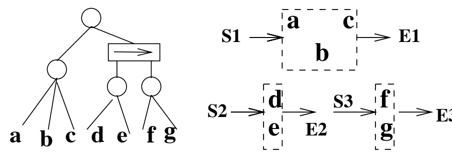
Given $\mathcal{S}$, the *minimal consensus PQ Tree* is a structure that captures all the common clusters that appear in each $s \in \mathcal{S}$. A linear time algorithm, $\mathcal{O}(|\mathcal{S}|\,|V_\alpha|)$, to compute this tree is discussed in [17]. We make a simple modification (not discussed here) in the algorithm to give oriented PQ (oPQ) trees where each Q node follows the strict left-to-right order. The number of augmented alphabet pairs $\mathbf{S}_k, \mathbf{E}_k$, in the augmented DAG correspond to the number of internal nodes in a PQ tree.

Algorithm 2 PQ Tree construction

```
Input: A set of sequences $\mathcal{S}$ each consisting of all symbols in $V$.
Output: A minimal consensus PQ Tree $T_c$ for $\mathcal{S}$.
1: Compute common intervals in sequences $\mathcal{S} = C_\Pi$ using algorithm in [11].
2: Initialize $T_c = \{$Universal tree$\}$
3: for all $c \in C_\Pi$ do
4:     $T_c = REDUCE(T_c, c)$\{See reference [4]\}
5: return $T_c$
```

To summarize, given $\mathcal{S}$, our scheme works as follows: (1) We first construct the minimal consensus oriented PQ tree T_c of $\mathcal{S}$ using Algorithm 2. (2) For each internal P node we construct an instance of Problem 1 and construct the maximal partial order. (3) Using the PQ structure of step (1) we combine all the maximal partial order DAGs of (step 2), with augmented characters $\mathbf{S}_k, \mathbf{E}_k$ for each to obtain the augmented maximal partial order.

Figure 7 shows the steps on the data of Example 5.1. (a)



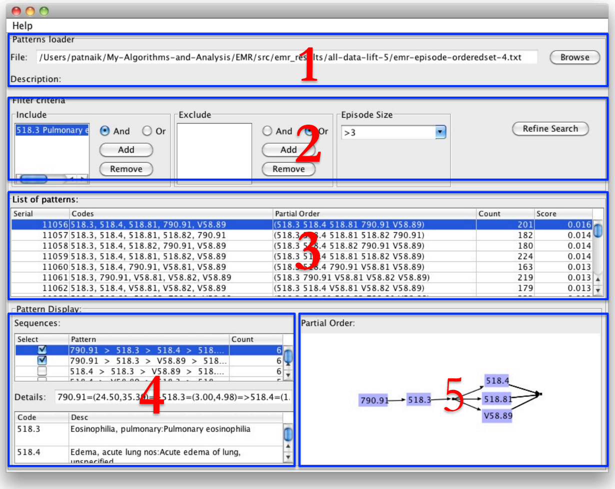
(a) (b)

Figure 7: (a) The minimal consensus oriented PQ structure for the data in Example 5.1. (b) The P nodes as placed in parallel and the Q nodes are placed serially.

shows the minimal consensus oriented PQ tree for the input data. Then the P nodes are arranged in parallel and the Q nodes are arranged serially as shown in (b). For each cluster, the partial order problem is solved and the resulting DAG is shown in Figure 6(b).

6. EMRVIEW

In this section we describe the features of the EMRView graphical user interface (GUI) software that serves as a partial order exploration tool. The parallel episode mining algorithm runs offline to generate a set of frequent parallel episodes for a user-defined support threshold. The parallel episodes are ordered in decreasing order of the significance score discussed earlier. This allows the user to focus his attention on the most surprising patterns in the data first. The offline mining algorithm also determines the support of the sub-sequences in the EMR sequences that constitute the occurrences of the given parallel episode. We learn a partial order over a subset of these sequences chosen so as to account for a sufficient fraction of the support of the parallel episode. This partial order brings out the ordering information embedded in the EMR data and is displayed via its Hasse diagram.



Patterns loader

File: /Users/patnaik/My-Algorithms-and-Analysis/EMR/src/emr_results/all-data-lift-5/emr-episode-orderedset-4.txt

Description:

Filter criteria

Include: 518.3 Pulmonary e [✓ And] Or

Exclude: And [✓ Or]

Episode Size: >3

List of patterns:

Serial	Codes	Partial Order	Count	Score
11056	518.3, 518.4, 518.81, 790.91, V58.89	(518.3 518.4 518.81 790.91 V58.89)	201	0.016
11057	518.3, 518.4, 518.81, 518.82, 790.91	(518.3 518.4 518.81 518.82 790.91)	182	0.014
11058	518.3, 518.4, 518.82, 790.91, V58.89	(518.3 518.4 518.82 790.91 V58.89)	180	0.014
11059	518.3, 518.4, 518.81, 518.82, V58.89	(518.3 518.4 518.81 518.82 V58.89)	224	0.014
11060	518.3, 518.4, 790.91, V58.81, V58.89	(518.3 518.4 790.91 V58.81 V58.89)	163	0.013
11061	518.3, 790.91, V58.81, V58.82, V58.89	(518.3 790.91 V58.81 V58.82 V58.89)	219	0.013
11062	518.3, 518.4, V58.81, V58.82, V58.89	(518.3 518.4 V58.81 V58.82 V58.89)	179	0.013

Pattern Display:

Sequences:

Select	Pattern	Count
✓	790.91 > 518.3 > 518.4 > 518....	6
✓	790.91 > 518.3 > V58.89 > 518...	6
	518.4 > 518.3 > V58.89 > 518....	6

Details: 790.91=(24.50,35.30)=>518.3=(3.00,4.98)=>518.4=(1.

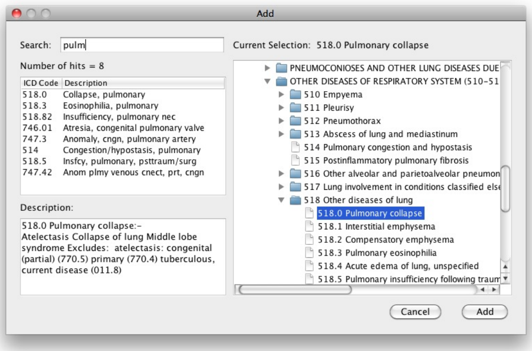
Code	Desc
518.3	Eosinophilia, pulmonary:Pulmonary eosinophilia
518.4	Edema, acute lung nos:Acute edema of lung, unspecified

Partial Order:

Figure 8: Screenshot of EMRView tool showing different panels of the interface.

Pattern loader (Panel 1): The offline result of parallel episode mining together with counts of the sequences constituting the occurrences of each parallel episode is loaded into the visualization tool. The tool also allows reading data in compressed/zip format.

Filtering results (Panel 2): For the parameters noted in the result section, the number of frequent parallel episodes is over 10,000. Looking through each of the parallel episodes and its supporting subsequences can be a daunting task. The EMRView tool allows the user to filter the list of episodes based on multiple criteria. The user can specify a set of diagnostic codes that must be present in the reported episodes with the option for enforcing all or any of the codes. Similarly the user can specify a set of diagnostic codes that must not be present. Further the user can specify the size of episodes that must be displayed.



Search: pulm

Current Selection: 518.0 Pulmonary collapse

Number of hits = 8

ICD Code	Description
518.0	Collapse, pulmonary
518.3	Eosinophilia, pulmonary
518.82	Insufficiency, pulmonary nec
746.01	Atresia, congenital pulmonary valve
747.3	Anomaly, cngn, pulmonary artery
514	Congestion/hypostasis, pulmonary
518.5	Insfcy, pulmonary, psttraum/surg
747.42	Anom plmy venous cnect, prt, cngn

Description:

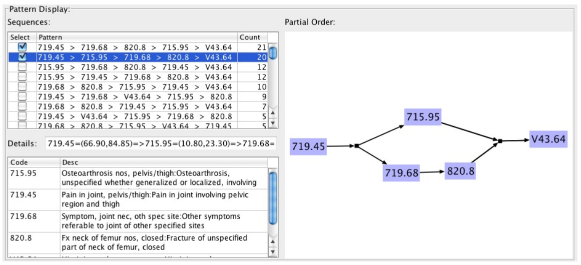
518.0 Pulmonary collapse:–
Atelectasis Collapse of lung Middle lobe syndrome Excludes: atelectasis: congenital (partial) (770.5) primary (770.4) tuberculous, current disease (011.8)

- PNEUMOCONIOSES AND OTHER LUNG DISEASES DUE
- OTHER DISEASES OF RESPIRATORY SYSTEM (510–51
  - 510 Empyema
  - 511 Pleurisy
  - 512 Pneumothorax
  - 513 Abscess of lung and mediastinum
  - 514 Pulmonary congestion and hypostasis
  - 515 Postinflammatory pulmonary fibrosis
  - 516 Other alveolar and parietoalveolar pneumon
  - 517 Lung involvement in conditions classified else
  - 518 Other diseases of lung
    - 518.0 Pulmonary collapse
    - 518.1 Interstitial emphysema
    - 518.2 Compensatory emphysema
    - 518.3 Pulmonary eosinophilia
    - 518.4 Acute edema of lung, unspecified
    - 518.5 Pulmonary insufficiency following traum

Figure 9: Screenshot of EMRView diagnostic code lookup interface.

A specially designed search dialog is presented to the user to select diagnostic codes. This dialog allows the user to search for the codes by entering either text describing the diagnostic code or the code if available. One could also pick codes by browsing through the hierarchy of the ICD codes. Figure 9 shows the search dialog illustrating the different options available for finding and adding diagnostic codes.

Result display (Panel 3): The filtered parallel episodes are displayed here. The first column gives an identifier for the row. The second column shows the list of codes in the parallel episode. Since there is no order imposed on the codes, we sort them in alphabetical order. The third column displays a partial order learnt from the set of sequences which together account for 75% of the support of the parallel episode. The same set of sequences are shown selected by default in Panel 4. Note that these sequences are ordered in descending order of support. The forth and fifth columns present the support (or count) and significance scores respectively. The rows are presorted in decreasing order of the significance score.



Select	Pattern	Count
✓	719.45 > 719.68 > 820.8 > 715.95 > V43.64	21
✓	719.45 > 715.95 > 719.68 > 820.8 > V43.64	20
	715.95 > 719.68 > 820.8 > 719.45 > V43.64	12
	715.95 > 719.45 > 719.68 > 820.8 > V43.64	12
	719.68 > 820.8 > 715.95 > 719.45 > V43.64	10
	719.45 > 719.68 > V43.64 > 715.95 > 820.8	9
	719.68 > 820.8 > 719.45 > 715.95 > V43.64	7
	719.45 > V43.64 > 715.95 > 719.68 > 820.8	5
	719.68 > 820.8 > 715.95 > V43.64 > 719.45	5

Details: 719.45=(66.90,84.85)=>715.95=(10.80,23.30)=>719.68=

Code	Desc
715.95	Osteoarthrosis nos, pelvis/thigh:Osteoarthrosis, unspecified whether generalized or localized, involving
719.45	Pain in joint, pelvis/thigh:Pain in joint involving pelvic region and thigh
719.68	Symptom, joint nec, oth spec site:Other symptoms referable to joint of other specified sites
820.8	Fx neck of femur nos, closed:Fracture of unspecified part of neck of femur, closed

Figure 10: Panel 4: Pattern display

Pattern display (Panel 4): This panel lists the sequences (more appropriately subsequence of the EMR sequences) that constitute the occurrences of the parallel episode. This panel is populated when the user selects a parallel episode in Panel 3. The sequences are listed in decreasing order of support. By default, sequences are selected until the sum of their support values exceeds a threshold of 75% of the total support of the parallel episode. The rationale is that these are likely the most important orders encountered in the data for the parallel episode under consideration. The partial order learnt from the selected sequences is displayed in the adjacent panel. In addition the user can select/deselect sequences. The partial order is updated online to reflect the selection. This puts the user in control of selecting sequences he deems useful and the algorithm summarizes them into a compatible partial order. As this is the most important part of EMRView, Figure 10 shows an expanded view of Panel 4 (with a different example).

7. RESULTS AND CLINICAL RELEVANCE

For the current study we chose to discover temporal patterns that occurred within a consecutive 200 day (6.5 month) period at support level $\geq 10^{-4}$. We did not identify any truly novel patterns but were able to discover many that reflect the capabilities of our algorithm and the power of our partial order discovery methodology.

Figure 11 shows the progression from a general diagnosis of hip pain to a subsequent diagnosis of osteoarthritis and a femur fracture, ultimately requiring a hip replacement. The code 719.68 is rather nonspecific but is typical of diagnosis codes used for routine care, especially when uncertainty exists in the final diagnosis.

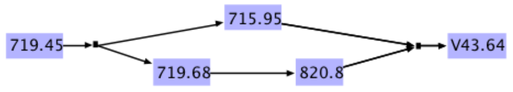
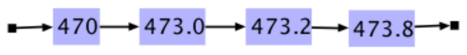


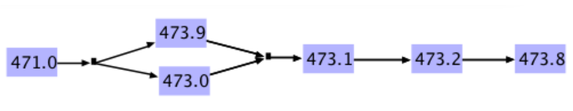
Figure 11: A five-sequence pattern demonstrating an initial diagnosis of pelvic pain (719.45) followed by the variable sequence of (a) a poorly defined joint symptom (719.68); (b) a femoral neck fracture (820.8); and (c) pelvic osteoarthritis (715.95), with a final common pathway converging on a hip joint replacement (V43.64).

Another set of patterns related to sinusitis are shown in Figures 12(a) and 12(b). In both of these sequential patterns, an initial diagnosis (deviated septum or nasal polyps) leads to various chronic sinusitis diagnoses. The sequence in Figure 12(a) occurred 102 times among the patient population, whereas the two sequences shown in Figure 12(b) occurred collectively 27 times. Similar sequences could also be found in the dataset with varying frequencies.

Some sequences converged on a diagnosis or event, whereas others diverged. This can be seen in Figures 13(a) and 13(b). Figure 13(a) shows events related to arm fractures at a school playground (representing 84 distinct patients), whereas Figure 13(b) shows events related to an initial pylenonephritis (kidney infection) event with subsequent diagnoses of calculi (stones) in the kidneys and urinary tracts. In this con-



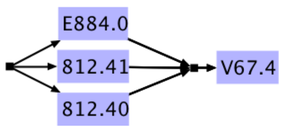
(a)

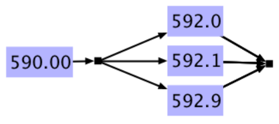


(b)

Figure 12: Sequences showing events leading to a sinusitis diagnosis. In (a) a deviated septum (470) leads to maxillary sinusitis (473.0), followed by ethmoidal sinusitis (473.2), and then a non-specific sinusitis diagnosis (473.8). In (b) nasal polyps (471.0) leads to maxillary sinusitis (473.0) or a non-specific sinusitis (473.9), followed by frontal sinusitis (473.1), then ethmoidal sinusitis (473.2) and finally another non-specific sinusitis (473.8).

text, a convergence means requiring a particular diagnosis before continuing to make other diagnoses.





(a) (b)

Figure 13: Sequences that converge onto a common event or diverge from a common event. The elements stacked vertically were found to occur interchangeably in the patterns found, suggesting that the specific ordering does not matter. In (a) A fall from playground equipment (E884.0) was associated in time with a lower humerus fracture(812.40) as well as a supracondylar humerus fracture (812.41), but all ended up with a follow-up exam for the healed fracture (V67.4). The diagram in (b) shows an initial pylenonephritis event (590.00) occurring, followed by the diagnoses of kidney calculus (e.g., stone) (592.0), ureter calculus (592.1), and non-specific urinary calculus (592.9).

The clinical relevance of the above patterns is quite immediate. For instance, the presence of osteoarthritis is a recommended, although controversial, indication for performing a hip joint replacement after the occurrence of a femoral neck fracture[10, 30, 19]. It is possible to imagine that someday clinicians may wish to interrogate their local data with EMRView to determine local patterns of care in order to answer questions such as "At our institution, what is our routine process of care for patients with a finding of osteoarthritis who subsequently develop a femoral neck fracture?" A key component in such a query would be to include the temporal concept of "subsequently" so that the timing of the fracture in relation to the other relevant diagnoses can be taken into account.

The observations regarding sinusitis are also interesting, and support the use of EMRView for hypothesis generation. It is well known that maxillary sinus is the most common site for such an infection to occur, followed by the ethmoid sinus (87% vs. 65%, respectively) [15], but it is worthwhile to note that in our observations, for the sequences that contained the codes for both maxillary (473.0) and ethmoidal sinusitis (473.2) the maxillary preceded the ethmoidal in the vast majority of cases. It may be that a severe maxillary sinusitis can often lead to an ethmoid infection with the reverse sequence being less common, or it may simply be that maxillary sinusitis is easier to diagnose and only later is an ethmoid sinusitis diagnosed in the same patient. It is also noteworthy that this temporal pattern occurred when the initial diagnoses was a deviated septum as well as with nasal polyps. It is unclear why this temporal pattern emerged. At least one source has reported that about 75% of all polyps are located in the ethmoid sinuses [20], which raises the question why the sinusitis was still primarily occurring in the maxillary sinuses first.

Among the other events we reported, the relationship between arm/humerus fractures and falls from school playground equipment is well documented in the literature [18, 12]. Nevertheless, monitoring how such patterns over time change at a specific medical center might allow for the detection of playgrounds that are especially dangerous, or perhaps the success of safety programs implemented to prevent such accidents. The relationship between pyelonephritis (kidney infections) and urinary system calculi also has important implications. As outlined in a recent report, using ultrasound to discover the calculi may be a useful tool when such infections are diagnosed since calculi may be a complicating factor in the management of the infections [5].

8. RELATED WORK

Many studies have explored temporal patterns in clinical data: some have focused on extracting temporal patterns from natural language descriptions in clinical documents [29, 9] whereas others have used coded administrative data, similar to the data we used [7, 3]. Reis et al. [27] used Bayesian models to predict future risks of abuse based on past episodes. They theorized that such a system could provide an early warning for clinicians to identify patients at high risk for future abuse episodes. Others have used temporal data mining to explore patterns related to specific diseases such as diabetes [6] or specific time frames such as adverse events following immunizations in a large cohort of nearly a million children in Denmark [32].

Plaisant and colleagues [26] built into the Microsoft Amalga software a user interface to help visualize patient specific temporal patterns. This did not involve data mining and the interface itself was significantly different from EMRView. More closely related, however, was the methodology and interface built by Moskovitch and Shahar [24] to identify and visualize temporal patterns in clinical data which was applied to a dataset of about 2000 patients with diabetes. More broadly, several published papers discuss the challenges faced in analyzing EMR data [2] starting from data integration aspects [28, 16] to privacy issues [23, 31]. Interest in the KDD community has blossomed recently, e.g., in [34], a multi label classifier is proposed that learns intercode relationships from clinical free text. In another study it has been shown that using the entropy of ICD9 codes it is possible to categorize diseases as chronic or acute [25]. Our work is novel in the application to a large scale database of EMRs to mine frequent and significant episodes, in the new partial order discovery algorithm, and in the development of EMRView based on user needs.

9. DISCUSSION

Our exercise in using temporal data mining to identify clinically relevant patterns in medical record data did demonstrate the feasibility of the approach. We were able to find hundreds of patterns using real world data that had been collected as a part of routine care. The clinical relevance of the patterns helps to confirm the validity of our mining algorithms.

Our current analysis brings about many future challenges for temporal mining in EMRs. For instance, thus far we have limited the time frame for analysis to 200 days. While this has the advantage, at least theoretically, of yielding greater clinical relevance between events that are temporally correlated, the disadvantage is the danger of missing correlated events that are far apart. Asbestos exposure and subsequent development of lung diseases is one such example [13]. Automatically identifying intervals for analysis is one area of research.

A second major issue with our analysis is that it is easier to obtain the data then it is to explore or assign meaning to the results. There is nothing intrinsic in the data itself to inform if a temporal sequence is novel, important, or even clinically valid. From our initial analysis the results do seem to be valid, but there is no existing database of clinical medicine with which to computationally compare our findings. The more frequently occurring patterns are probably a reflection of an event occurring more frequently among our population. But it may very well be that the less frequently occurring events are the ones that are not well described in the literature and may represent novel discoveries. Finding a scalable approach to explore this space is still an open question.

Last, while we did have nearly 100 million events with which to detect patterns, these codes still represent only a tiny fraction of the clinical parameters that are relevant to each patient. Many clinical details were not present in the dataset that might have a significant impact on the interpretation of the results. It will be important to develop methods to incorporate these additional data into the data mining algorithms to extract more meaningful results in the future.

Acknowledgements

This work is supported in part by the Institute for Critical Technology and Applied Science, Virginia Tech and by US NSF grant CCF-0937133.

Availability of source code

The software is available at http://nostoc.cs.vt.edu/emr-view.

10. REFERENCES

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *11th ICDE*, pages 3—14, Taipei, Taiwan, 1995.

[2] P. A. Bath. Health informatics: current issues and challenges. *J. Inf. Sci.*, 34:501–518, August 2008.

[3] R. Bellazzi et al. Methods and tools for mining multivariate temporal data in clinical and biomedical applications. In *Conf Proc IEEE Eng Med Biol Soc.*, pages 5629–32, 2009.

[4] K. Booth and G. Lueker. Testing for the Consecutive Ones Property, Interval Graphs, and Graph Planarity using PQ-tree Algorithms. *JCSS*, 13(3):335–379, 1976.

[5] K. C. Chen, S. W. Hung, V. K. Seow, C. F. Chong, T. L. Wang, Y. C. Li, and H. Chang. The role of emergency ultrasound for evaluating acute pyelonephritis in the ed. *Am J Emerg Med*, Apr 2010.

[6] S. Concaro et al. Temporal data mining for the assessment of the costs related to diabetes mellitus pharmacological treatment. In *AMIA Annu Symp Proc.*, pages 119–123, 2009.

[7] S. Concaro et al. Mining health care administrative data with temporal association rules on hybrid events. *Methods Inf Med*, 50(2), Dec 2010.

[8] C. De Coster et al. Identifying priorities in methodological research using ICD-9-CM and ICD-10 administrative data: report from an international consortium. *BMC Health Serv Res.*, 6:77, Jun 2006.

[9] J. Denny et al. Extracting timing and status descriptors for colonoscopy testing from electronic medical records. *J Am Med Inform Assoc.*, 17(4):383–8, Jul-Aug 2010.

[10] W. L. Healy and R. Iorio. Total hip arthroplasty: optimal treatment for displaced femoral neck fractures in elderly patients. *Clin Orthop Relat Res*, (429):43–48, Dec 2004.

[11] S. Heber and J. Stoye. Finding all common intervals of k permutations. In *Proc. of the 12th CPM'01*, pages 207–218, London, UK, UK, 2001. Springer-Verlag.

[12] A. W. Howard, C. Macarthur, L. Rothman, A. Willan, and A. K. Macpherson. School playground surfacing and arm fractures in children: a cluster randomized trial comparing sand to wood chip surfaces. *PLoS Med*, 6(12), Dec 2009.

[13] E. Jamrozik, N. de Klerk, and A. Musk. Clinical review: Asbestos-related disease. *Intern Med J*, Feb 2011.

[14] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106(4):620–630, May 1957.

[15] W. Kormos. *Primary Care Medicine*, chapter Approach to the Patient With Sinusitis, pages 1402–1407. Lippincot WIlliams and WIlkins, 2009.

[16] R. Kukafka et al. Redesigning electronic health record systems to support public health. *J. of Biomedical Informatics*, 40:398–409, August 2007.

[17] G. Landau, L. Parida, and O. Weimann. Using PQ Trees for Comparative Genomics. In *Proc. CPM*, pages 128–143, 2005.

[18] R. T. Loder. The demographics of playground equipment injuries in children. *J Pediatr Surg*, 43(4):691–699, Apr 2008.

[19] J. A. Lowe, B. D. Crist, M. Bhandari, and T. A. Ferguson. Optimal treatment of femoral neck fractures according to patient's physiologic age: an evidence-based review. *Orthop Clin North Am*, 41(2):157–166, Apr 2010.

[20] T. M and L. PL. *Nasal Polyps: Origin, Etiology, Pathenogensis, and Structure*, chapter Diseases of the sinuses: diagnosis and management, pages 57–68. 2001.

[21] H. Mannila et al. Discovery of frequent episodes in event sequences. *DMKD*, 1:259–289, 1997.

[22] H. Mannila and C. Meek. Global Partial Orders from

Sequential Data. In *Proc. KDD'00*, pages 161–168, 2000.

[23] L. Martino and S. Ahuja. Privacy policies of personal health records: an evaluation of their effectiveness in protecting patient information. In *Proc. of 1st ACM Intl. Health Informatics Symp.*, pages 191–200, New York, NY, USA, 2010. ACM.

[24] R. Moskovitch and Y. Shahar. Medical temporal-knowledge discovery via temporal abstraction. In *AMIA Annu Symp Proc.*, pages 452–456, 2009.

[25] A. Perotte. Using the entropy of icd9 documentation across patients to characterize using the entropy of ICD9 documentation across patients to characterize disease chronicity. AMIA Annual Symp., Nov 2010.

[26] C. Plaisant et al. Searching electronic health records for temporal patterns in patient histories: a case study with microsoft amalgaa. In *AMIA Annu Symp Proc. 2008*, pages 601–605, 2008.

[27] B. Reis et al. Longitudinal histories as predictors of future diagnoses of domestic abuse: modelling study. *BMJ*, 2009.

[28] D. Revere et al. Understanding the information needs of public health practitioners: A literature review to inform design of an interactive digital knowledge management system. *J. of Biomedical Informatics*, 40:410–421, August 2007.

[29] G. Savova et al. Towards temporal relation discovery from the clinical narrative. In *AMIA Annu Symp Proc.*, pages 569–572, 2009.

[30] E. Sendtner, T. Renkawitz, P. Kramny, M. Wenzl, and J. Grifka. Fractured neck of femur–internal fixation versus arthroplasty. *Dtsch Arztebl Int*, 107(23):401–407, Jun 2010.

[31] C. Stingl and D. Slamanig. Privacy enhancing methods for e-health applications; how to prevent statistical analyses and attacks. *Int. J. Bus. Intell. Data Min.*, 3:236–254, December 2008.

[32] H. Svanstrom et al. Temporal data mining for adverse events following immunization in nationwide danish healthcare databases. *Drug Saf.*, 33(11):1015–25, Nov 2010.

[33] N. Tatti. Maximum entropy based significance of itemsets. In *Proc. 7th ICDM'07*, pages 312–321, 2007.

[34] Y. Yan, G. Fung, J. G. Dy, and R. Rosales. Medical coding classification by leveraging inter-code relationships. In *Proc. 16th KDD'10*, pages 193–202, New York, NY, USA, 2010. ACM.