Automatic Detection Of Vaccine Adverse Reactions By Incorporating Historical Medical Conditions

Zhonghua Jiang
Computer Science & Engineering
University of Minnesota, Twin Cities
zjiang@cs.umn.edu

George Karypis
Computer Science & Engineering
University of Minnesota, Twin Cities
karypis@cs.umn.edu

Abstract

This paper extends the state of art by bringing the historical medical conditions into the vaccine adverse reaction discovery process. The goal is to identify evidences which suggest that given adverse reaction is likely to be developed for individuals with certain medical conditions when they are vaccinated with certain vaccines. We propose a novel measure called dual-lift for this task. It is shown that the dual-lift measure can not only identify medical conditions associated with known vaccine adverse reactions, but also have the potential of detecting new adverse reactions that are otherwise hidden. We formulate this problem in the framework of constraint pattern mining. Three constraints are developed. The first is based on the dual-lift measure which aims to discover meaningful patterns, the second is used to remove redundancy, and the third is used to ensure prevalence of generated patterns. We present a pattern mining algorithm DLiftMiner which utilizes a novel approach to upper bound the dual-lift measure for reducing the search space. Experimental results show that our pruning methods lead to dramatic performance improvement. It is also shown that DLiftMiner scales linearly with the size of input database. Some interesting vaccine adverse reactions discovered from VAERS database are presented.

1. INTRODUCTION

Detecting adverse reactions of marketed vaccines is a well known important but challenging task. Due to statistical and clinical limitations, preapproval trials may fail to identify vaccine adverse reactions. This is where the postmarketing surveillance of spontaneous reporting system takes on special importance. The Vaccine Adverse Event Reporting System (VAERS) is such a system co-managed by the Food and Drug Administration (FDA) and the Centers for Disease Control and Prevention (CDC) [1]. It collects information about adverse reactions (possible side effects) that occur after the administration of vaccines licensed for use in the United States.

Existing approaches for adverse reaction detection ([2], [3], [4], [5], [6]) focus on vaccine-adverse reaction pairs that are statistically overrepresented and deserve further investigations. Potentially rich background information associated with the reports are ignored. In fact, in VAERS, each report contains historical medical conditions under which the vaccine adverse reactions are developed, such as, medications the vaccine recipient was taking, pre-existing physician diagnosed allergies and birth defects, and any illness at the time of vaccination. This information may potentially be very informative, as it will allow us to identify sub-groups of individuals (with specific medical conditions) that are more susceptible to given vaccine adverse reactions.

The focus of this paper is to develop efficient methods to identify patterns of the form (medical conditions, vaccines, adverse reactions), which suggests that the adverse reactions are likely to be developed for individuals with medical conditions when they are vaccinated with vaccines. It makes three key contributions. The first contribution is that it introduces a novel measure called dual-lift to capture correlations between vaccines, adverse reactions and medical conditions. The second contribution is that it formulates the problem of discovering vaccine adverse reactions with historical medical conditions as a constraint pattern mining problem. Three constraints are developed. The first is based on the dual-lift measure which aims to discover meaningful patterns, the second is used to remove redundancy, and the third is used to ensure prevalence of generated patterns. The third contribution is that it develops a computationally efficient algorithm called DLiftMiner (which stands for Dual-Lift Miner) for mining constraint patterns in VAERS database, that utilizes a novel approach to upper bound the dual-lift measure for reducing the search space.

The DLiftMiner algorithm is evaluated on a series of synthetically generated datasets. It is shown that the performance is improved dramatically due to the effective pruning of the search space. It is also shown that DLiftMiner scales linearly with the size of input database. Furthermore we evaluate the ability of the dual-lift measure and the DLiftMiner algorithm to find interesting vaccine adverse reaction pairs on VAERS database. It is shown that the dual-lift measure can not only find medical conditions associated with known vaccine adverse reaction pairs, but also have the potential of detecting new adverse reactions that are otherwise hidden.
The remaining of this paper is organized as follows. Section 2 presents related works to constraint pattern mining and adverse reaction detection. Section 3 introduces notations and definitions that are used through out the rest of the paper. Section 4, presents the dual-lift measure. Section 5 formulates the constraint pattern mining problem. The DLiftMiner algorithm is explained in Section 6. Experimental results with both synthetic and real world datasets are included in Section 7. Finally Section 8 provides some concluding remarks.

2. RELATED WORK
Among the vast amount of work related to the problem of pattern mining, the research that is directly relevant to this work are the approaches that have been developed for dealing with tough constraints [7]. A constraint is tough if it is neither anti-monotone nor monotone, and cannot be converted to either anti-monotone or monotone constraint [7]. Implementing effective pruning strategies is crucial for push-verted to either anti-monotone or monotone constraint [7]. is neither anti-monotone nor monotone, and cannot be con-

2. RELATED WORK
Among the vast amount of work related to the problem of pattern mining, the research that is directly relevant to this work are the approaches that have been developed for dealing with tough constraints [7]. A constraint is tough if it is neither anti-monotone nor monotone, and cannot be converted to either anti-monotone or monotone constraint [7]. Implementing effective pruning strategies is crucial for pushing tough constraints into pattern mining process. For example, a new class of tough constraint called Loose Anti-Monotone constraint is introduced in [8]. And the pruning strategy in [8] is based on the fact that a transaction can be deleted if it is not superset of at least one frequent k-itemset satisfying the constraint at any iteration $k \geq 2$. Another example is that LPMiner [9] utilizes smallest valid extension (SVE) property for pruning when dealing with the length decreasing support constraint.

The approach we take in this work is based on the idea of boundable constraint [10]. There are a few other works which explore this same idea. For instance, in [11], an upper bound of chi-square is derived based on convexity of chi-square function. And in [12], the upper bound of Pearson’s correlation is shown to have conditional monotone property. In [13], the pruning strategy is to upper bound the support, confidence and a new measure called improvement.

Another category of related works has to do with adverse reaction discovery. Different measures were proposed for this task, such as, the proportional reporting ratio (PRR) [2], the empirical Bayes geometric mean (EBGM) [3], and the reporting odds ratio (ROR) [4][5]. Bate et al [6] uses information component (IC) as the measure of associations, which is most close to our work since the information component is defined as the logarithm of lift. However, to the best of our knowledge, no prior work has taken into account historical medical conditions explicitly into the adverse reaction discovery problem. And our measure dual-lift is proposed specifically for this task.

3. DEFINITIONS AND NOTATION
A vaccine-adverse reaction database $DB$ is a set of records of the form $(tid, M, V, R)$, where $tid$ is a unique record id. Each record can be interpreted as that an individual with historical medical conditions $M$ developed adverse reactions $R$ after the vaccination of $V$. We will refer to the sets of items in $M$, $V$, and $R$ as the $m$-items, $v$-items and $r$-items, respectively.

A record $(tid, M, V, R)$ is said to contain itemset $I$ if $I \subseteq M \cup V \cup R$. The support of $I$ in $DB$, denoted by $supp(I|DB)$, is the total number of records in $DB$ that contain $I$. The transaction count of $I$ in $DB$, denoted by $tcnt(I|DB)$, is the number of records for which $I = M \cup V \cup R$. An association rule is of the form $I \rightarrow r$, where $I$ is the prefix itemset and $r$ is the suffix itemset. The confidence of $I \rightarrow r$ is defined as

$$conf(I \rightarrow r|DB) = \frac{supp(I \cup r|DB)}{supp(I|DB)}.$$  

(1)

The lift of $I \rightarrow r$ is defined as

$$lift(I \rightarrow r|DB) = \frac{conf(I \rightarrow r|DB)}{conf(\emptyset \rightarrow r|DB)},$$

(2)

where $\emptyset$ is the null set.

A conditional database $DB_{A}$ is a set of records from $DB$ which contain itemset $A$. A projected database $DB_{B}$ is formed by projecting all records from $DB$ onto itemset $B$, that is, removing all items which are not in $B$ from each record in $DB$. A projected conditional database $DB_{B|A}$ is defined as projected database from $DB_{A}$ on itemset $B$. Let set $DB^{(4)}$ contain all itemsets whose transaction counts in $DB$ are non-zero, and similarly $DB^{(4)}_{B|A}$ ($DB^{(4)}_{B|A}$) contain all itemsets whose transaction counts in $DB_{B}$ ($DB_{B|A}$) are non-zero.

4. DUAL-LIFT MEASURE
Our objective is to identify all patterns of the form $\langle M, V, R \rangle$ such that there are evidences to suggest adverse reactions $R$ are likely to be developed if individuals with historical medical conditions $M$ are vaccinated with $V$. The key step towards this goal is to develop a measure to capture correlations among $M$, $V$, and $R$.

One may consider to use $lift(M \cup V \rightarrow R|DB)$ as the measure. However, this measure has the weakness that $M$ and $V$ are not distinguished from each other, so that the correlation may be introduced by one of them, say $V$, but not the other one, say $M$. Another possibility is

$$L = \frac{P(M \cup V \cup R|DB)}{P(M|DB)P(V|DB)P(R|DB)}.$$  

(3)

where $P(I|DB)$ ($I$ is $M$, $V$, $R$ or $M \cup V \cup R$) is the probability of $I$ estimated in $DB$ (for example, $P(I|DB)$ can be estimated by $conf(\emptyset \rightarrow I|DB)$). The problem with this measure is that $L$ can be high if any two of the three sets are highly correlated. In fact, it is easy to see that $L = lift(M \cup V \rightarrow R|DB)lift(M \rightarrow V|DB)$. So $L$ can be high if only $M$ and $V$ are highly correlated.

Instead, we propose the following measure:

**Definition 4.1 (dual-lift).** The dual-lift of a pattern $\langle M, V, R \rangle$, denoted by dual-lift($\langle M, V, R \rangle|DB$), is defined as:

$$\min\{lift(V \rightarrow R|DB_{M}), lift(M \rightarrow R|DB_{V})\}.$$  

(4)

The dual-lift measure combines two quantities. The first, $lift(V \rightarrow R|DB_{M})$, is designed to detects vaccine adverse reaction pairs $\langle V, R \rangle$ in a subset of records containing medical conditions $M$. This is because if individuals with medical conditions $M$ are susceptible to $\langle V, R \rangle$, we should be able to detect $\langle V, R \rangle$ in $DB_{M}$. Another advantage of this lift measure is that we can uncover potentially important vaccine
adverse reaction pairs $(V,R)$ ($\text{lift}(V \rightarrow R|DB_{(M)})$ is high), which are difficult to detect by looking at the whole database ($\text{lift}(V \rightarrow R|DB)$ is low). The second, $\text{lift}(M \rightarrow R|DB_{(V)})$, is designed to identify individuals with medical conditions $M$ which have higher chance of developing adverse reactions $R$ when vaccinated with $V$. This is because $\text{lift}(M \rightarrow R|DB_{(V)})$ is high implies that $\text{conf}(M \cup V \rightarrow R|DB)$ is larger than $\text{conf}(V \rightarrow R|DB)$. However, there is a limitation when these two quantities are used alone, that is, a high value of $\text{lift}(V \rightarrow R|DB_{(M)})$ does not imply that $V$ is correlated with $M$. The dual-lift measure is designed to overcome this limitation by combining them together.

5. PROBLEM FORMULATION

In this section, we formulate the problem of detecting vaccine adverse reactions with historical medical conditions in the framework of constraint pattern mining. In our formulation, there are three components that define the interesting patterns: (i) minimum dual-lift constraint, which utilizes the dual-lift measure developed in Section 4, (ii) minimum improvement constraint, which aims to remove redundant patterns, and (iii) minimum support constraint, which defines the set of frequent patterns. The formal definitions of these components are as follows:

Definition 5.1 (Minimum dual-lift Constraint). The pattern $(M,V,R)$ satisfies the minimum dual-lift constraint of $l_0 > 0$ in database $DB$ if

$$\text{dual}_{\text{lift}}((M,V,R)|DB) \geq l_0.$$  (5)

Definition 5.2 (Minimum improvement constraint). The pattern $(M,V,R)$ satisfies the minimum improvement constraint of $m_0 \geq 0$ if

$$\text{dual}_{\text{lift}}((M,V,R)|DB) - \text{dual}_{\text{lift}}((M',V',R')|DB) \geq m_0.$$  (6)

for any of its proper sub-pattern $(M',V',R')$ (that is, $M' \subseteq M \land V' \subseteq V \land R' \subseteq R$ and $M' \cup V' \cup R' \neq M \cup V \cup R$), satisfying the minimum dual-lift constraint of $l_0$. A pattern which does not satisfy minimum improvement constraint is said to be redundant.

Definition 5.3 (Minimum support Constraint). The pattern $(M,V,R)$ satisfies minimum support constraint of $s_0 > 0$ if

$$\supp(M \cup V \cup R|DB) \geq s_0.$$  (7)

A pattern satisfying minimum support constraint is called a frequent pattern.

The idea of introducing the minimum improvement constraint is borrowed from [13]. The goal is to select a subset of good patterns which can be presented to domain experts. Intuitively, Definition 5.2 says that a pattern should be considered redundant if it does not achieve sufficient improvement over its proper sub-patterns satisfying minimum dual-lift constraint. This is because the correlation observed

Algorithm 1 DLiftMiner

Input: An input vaccine adverse reaction database $DB^0$

1: read records in $DB^0$; count frequencies of $m$-items and $v$-items.

2: read records again and construct database $DB$ as a FP-Tree, where $m$-items and $v$-items are internal nodes and $r$-items are leaves. $m$-items and $v$-items are sorted based on their frequencies from large to small along the tree from top to bottom.

3: remove infrequent items in database $DB$.

4: Let $M_1, V_1, R_1$ be sets of $m$-items, $v$-items, $r$-items in $DB$.

5: $DB^m \leftarrow DB_{M_1 \cup R_1}$; $DB^v \leftarrow DB_{V_1 \cup R_1}$

6: for $r$ in $R_1$ do

7: $l[r] \leftarrow l_0$

8: end for

9: $M_0 \leftarrow \emptyset$; $V_0 \leftarrow \emptyset$

10: branch_and_bound($DB$, $DB^m$, $DB^v$, $l$, $M_0$, $V_0$)

in this pattern may have been captured in its sub-patterns already.

In this paper, we only consider the case when $R$ contains a single reaction. With the aid of above definitions, our problem can be stated as:

Problem 5.4. Given database $DB$ and user specified parameters $l_0$, $m_0$ and $s_0$, identify all possible patterns $(M,V,r)$ satisfying minimum dual-lift constraint of $l_0$, minimum improvement constraint of $m_0$ (non-redundant) and minimum support constraint of $s_0$ (frequent), where $M$ is a set of $m$-items, $V$ is a set of $v$-items, and $r$ is a set containing a single $r$-item.

6. DLIFTMINER ALGORITHM

The DLiftMiner (see Algorithm 1) algorithm follows the depth-first approach for pattern mining and grows the patterns by adding one item at a time [14]. It reduces the size of the dataset successively by constructing the conditional database which is the subset of transactions that contain the current pattern that is being grown. The database is stored using the FP-tree [15] data structure.

The core of the DLiftMiner algorithm is the branch_and_bound algorithm that is sketched in Algorithm 2. The branch_and_bound algorithm starts with calculating for each $r$-item $r$ of $DB$ the dual-lift for the pattern $(M_0, V_0, r)$ (lines 2-6). If the calculated dual-lift is larger or equal to its threshold for the corresponding $r$-item, $l[r]$ is updated (line 9). Then it determines whether the next item $i$ should be conditional on by finding the set $R$ which contains frequent $r$-items in $DB^r$ (or $DB^0$) whose dual-lift upper bound is larger than or equal to current threshold. The dual-lift upper bounds are calculated in lines 24-25 as dual-lift-bound, where bound$^u$ is the upper bound of $\text{lift}(V^0 \cup V \rightarrow r|DB^0_{M_0 \cup M})$ (line 24), bound$^u_{(M \rightarrow r|DB^0_{(V \cup V))}}$ for any frequent patterns $(M,V,r)$ in $DB^r$. If $R$ is not empty, new databases $DB'$, $DB^m'$, $DB^v'$ are constructed with necessary items removed (lines 30-36), and branch_and_bound is called with updated parameters (line 35). Note that there is no need to actually construct $DB'$,
Algorithm 2: branch_and_bound($DB$, $DB^m$, $DB^v$, $l$, $M_0$, $V_0$)

**Input:** $l$ contains the current dual-lift threshold for each $r$-item. $M_0$ and $V_0$ are the sets of $m$-items and $v$-items that have been searched. $DB$ is $DB^0_{M_0 \cup V_0}$ with items in $M_0 \cup V_0$ and other (for example, infrequent) items removed. $DB^m$ is the projected database from $DB^0_{M_0}$ onto $m$-items and $r$-items of $DB$. $DB^v$ is the projected database from $DB^0_{V_0}$ onto $v$-items and $r$-items of $DB$.

```latex
1: for $r$ in all $r$-items of $DB$ do
2:   $S_r \leftarrow supp(r) |DB| / |DB^m|$
3:   $S_r^m \leftarrow supp(r) |DB^m| / |DB^m^m|$
4:   $S_r^v \leftarrow supp(r) |DB^v| / |DB^v^v|$
5:   $L_r^m \leftarrow S_r / S_r^m$; $L_r^v \leftarrow S_r / S_r^v$
6: end if
7: if dual-lift, $\geq l[r]$ then
8:   print $(M_0, V_0, r)$
9:   $l[r] \leftarrow$ dual-lift, + $m_0$
10: end if
11: end for
12: while $DB$ has more $m$-items or $v$-items do
13:   $i \leftarrow$ next $m$-item or $v$-item of $DB$
14:   $DB' \leftarrow DB_i$
15:   if $i$ is $m$-item then
16:      $M_i' \leftarrow M_0 \cup i$; $V_i' \leftarrow V_0$
17:      $DB^{m'} \leftarrow DB^{m'}_i$, $DB^{v'} \leftarrow DB^{v'}_i$
18:   else
19:      $V_i' \leftarrow V_0 \cup i$; $M_i' \leftarrow M_0$
20:      $DB^{m'} \leftarrow DB^{m'}_i$, $DB^{v'} \leftarrow DB^{v'}_i$
21:   end if
22:   let $R$ be the set of frequent $r$-items in $DB'$
23:   for $r$ in $R$ do
24:      $bnd_r^m \leftarrow b_{i|DB'}^m / b_{i|DB^{m'}}^m$, $bnd_r^v \leftarrow b_{i|DB'}^v / b_{i|DB^{v'}}^v$
25:      dual-lift-bnd, $= \min(bnd_r^m, bnd_r^v)$
26:      if dual-lift-bnd, $\geq l[r]$ then
27:         remove $r$ from $R$
28:   end if
29: end for
30: if $|R| > 0$ then
31:    construct $DB'$; $i$ and $r$-items not in $R$ are not copied
32:    remove infrequent items in $DB'$
33:    construct $DB^{m'}$; remove items that are not in $DB'$
34:    construct $DB^{v'}$; remove items that are not in $DB'$
35:    branch_and_bound($DB'$, $DB^{m'}$, $DB^{v'}$, $l$, $M_i'$, $V_i'$)
36: end if
37: remove $i$ from DB
38: if $i$ is $m$-item then
39:   remove $i$ from $DB^m$
40: else
41:   remove $i$ from $DB^v$
42: end if
43: end while
```

$DB^m$ and $DB^v$ in lines 14-20; we can operate on the original databases $DB$, $DB^m$, and $DB^v$ for evaluating lift upper bounds in line 24. Finally, item $i$ is removed (lines 38-42) and the next item is considered (line 13).

Lines 2-6 calculate the dual-lift of the current pattern $(M_0, V_0, r)$. $L_r^m (L_r^v)$ is the quantity $lift(V_0, \rightarrow |DB^0_{M_0}|)$ ($lift(M_0, \rightarrow |DB^0_{V_0}|)$). To see why these calculations are correct, consider:

$$lift(V_0 \rightarrow r|DB^0_{M_0}) = \frac{conf(V_0 \rightarrow r|DB^0_{M_0})}{conf(\emptyset \rightarrow r|DB^0_{M_0})}$$

$$= \frac{conf(\emptyset \rightarrow r|DB^0_{M_0 \cup V_0})}{conf(\emptyset \rightarrow r|DB^0_{M_0})}$$

$$= \frac{\supp(r|DB)|\emptyset \rightarrow r|DB^0_{M_0}}{\supp(r|DB^0_{M_0})/|DB|^0_{M_0}}$$

and similarly,

$$lift(M_0 \rightarrow r|DB^0_{V_0}) = \frac{\supp(r|DB)|\emptyset \rightarrow r|DB^0_{V_0}}{\supp(r|DB^0_{V_0})/|DB|^0_{V_0}}$$

6.1 Pruning Strategies

During pattern growth, DLIghtMiner utilizes various approaches to prune the search space. The support based pruning is used to remove infrequent items. Improvement based pruning is used to set the dual-lift thresholds higher and higher. Finally the dual-lift based pruning finds the upper bound of the dual-lift measure. And if the upper bound is less than current dual-lift threshold, the rest of search space is pruned away. Additional details of these pruning methods are provided in the rest of this section.

6.1.1 Support Based Pruning

Support based pruning can be derived from the minimum support constraint in Definition 5.3. The idea is to remove items that are considered infrequent. Any $r$-item is infrequent if its support is less than $s_0$. In our formulation, every interesting pattern is associated with an $r$-item. For any $m$-item or $v$-item we define its maximum support per class as max, $\supp(i \cup r|DB)$. Any $m$-item or $v$-item whose maximum support per class is less than $s_0$ is considered infrequent and thus removed. Our algorithm applies support based pruning in line 3 of DLIghtMiner and lines 32-34 of branch_and_bound, where infrequent items are removed, and line 22 of branch_and_bound, where only frequent $r$-items are considered.

6.1.2 Improvement Based Pruning

For any $r$-item $r$, patterns to be explored by the branch and bound algorithm are super-patterns of $(M_0, V_0, r)$. From the definition of minimum improvement constraint, the dual-lift thresholds for patterns to be explored should be no less than $m_0$ plus the maximum dual-lift values for any proper sub-patterns of $(M_0, V_0, r)$ that branch and bound has discovered. These dual-lift thresholds are saved in $l$ (line 9). When a new pattern is discovered, its dual-lift has to be no less than current dual-lift threshold. That means, this new
pattern must have largest dual-lift value among its discovered sub-patterns (due to \(m_0\) is non-negative). So updating corresponding dual-lift threshold is straightforward: simply add \(m_0\) to dual-lift value of the new pattern (line 9).

However, it should be noted that the current implementation of our algorithm does not take into account all the redundancy that can be exploited by Definition 5.2 due to the nature of depth first search. This is because that not all sub-patterns of \(\langle M_0, V_0, r \rangle\) were processed before the pattern \(\langle M_0, V_0, r \rangle\). For example, assume \(M_0 = \{m_1, m_2\}\) and \(V_0 = \{v_1\}\) and assume these three items were added to \(M_0 \cup V_0\) in the order of \(m_1, v_1, m_2\), then sub-patterns processed before \(\langle \{m_1, m_2\}, v_1, r \rangle\) are \(\langle \emptyset, \emptyset, r \rangle, \langle m_1, \emptyset, r \rangle,\) and \(\langle m_1, v_1, r \rangle\). Other sub-patterns (for example, \(\langle m_2, v_1, r \rangle\)) are processed after \(\langle \{m_1, m_2\}, v_1, r \rangle\). Extending our current algorithm to properly handle such cases is something that we are currently working on.

6.1.3 Lift Based Pruning

From line 24 in the branch and bound algorithm, we need to find upper bounds for the quantities \(\text{lift}(V_0 \cup V \rightarrow \rightarrow DB_{\mathcal{M}[M_{\cup}, \mathcal{M}]})\) and \(\text{lift}(M_0 \cup M \rightarrow \rightarrow DB_{\mathcal{M}[M_{\cup}, \mathcal{M}]})\) for any frequent patterns \(\langle M, V, r \rangle\) in \(DB^r\).

This problem can be tackled by observing that \(\text{lift}\) is the ratio of two confidences. In fact,

\[
\text{lift}(V_0 \cup V \rightarrow \rightarrow DB_{\mathcal{M}[M_{\cup}, \mathcal{M}]}) = \frac{\text{conf}(V_0 \cup V \rightarrow \rightarrow DB_{\mathcal{M}[M_{\cup}, \mathcal{M}]})}{\text{conf}(\emptyset \rightarrow \rightarrow DB_{\mathcal{M}[M_{\cup}, \mathcal{M}]})}
\]

and similarly,

\[
\text{lift}(M_0 \cup M \rightarrow \rightarrow DB_{\mathcal{M}[M_{\cup}, \mathcal{M}]}) = \frac{\text{conf}(M \cup V \rightarrow \rightarrow DB_{\mathcal{M}[M_{\cup}, \mathcal{M}]})}{\text{conf}(M \rightarrow \rightarrow DB_{\mathcal{M}[M_{\cup}, \mathcal{M}]})},
\]

and

\[
\text{lift}(M_0 \cup M \rightarrow \rightarrow DB_{\mathcal{M}[M_{\cup}, \mathcal{M}]}) = \frac{\text{conf}(M \cup V \rightarrow \rightarrow DB_{\mathcal{M}[M_{\cup}, \mathcal{M}]})}{\text{conf}(M \rightarrow \rightarrow DB_{\mathcal{M}[M_{\cup}, \mathcal{M}]})},
\]

Because \(DB^r\) is \(DB_{[M_{\cup}, \mathcal{M}]v_0}\) with some items removed, \(DB^{r_m}\) (\(DB^{r_m}\)) is \(DB_{[M_{\cup}, \mathcal{M}]v_0}\) with some items removed, and \(\langle M, V, r \rangle\) frequent patterns from \(DB^r\), it is not hard to see that \(DB_{[M_{\cup}, \mathcal{M}]v_0}\), \(DB_{[M_{\cup}, \mathcal{M}]v_0}\), and \(DB_{[M_{\cup}, \mathcal{M}]v_0}\) can be replaced by \(DB^r, DB^{r_m}\), and \(DB^{r_m}\) in equations 10 and 11. So we have

\[
\text{lift}(V_0 \cup V \rightarrow \rightarrow DB_{\mathcal{M}[M_{\cup}, \mathcal{M}]}) = \frac{\text{conf}(M \cup V \rightarrow \rightarrow DB^r)}{\text{conf}(V \rightarrow \rightarrow DB^r)},
\]

and

\[
\text{lift}(M_0 \cup M \rightarrow \rightarrow DB_{\mathcal{M}[M_{\cup}, \mathcal{M}]}) = \frac{\text{conf}(M \cup V \rightarrow \rightarrow DB^r)}{\text{conf}(M \rightarrow \rightarrow DB^r)}.
\]

Equations 12 and 13 suggest that, to find the upper bound of lift, we can find the upper bound of numerator and the lower bound of denominator. So, line 24 is correct if we introduce

- \(\text{bnd}^{r_{DB^r}}\) be upper bound of \(\text{conf}(M \cup V \rightarrow \rightarrow DB^r)\) for any frequent patterns \(\langle M, V, r \rangle\) in \(DB^r\),
- \(\text{bnd}^{r_{DB^{r_m}}}\) be lower bound of \(\text{conf}(M \rightarrow \rightarrow DB^{r_m})\) for any frequent patterns \(\langle M, \emptyset, r \rangle\) in \(DB^{r_m}\),
- \(\text{bnd}^{r_{DB^{r_m}}}_{\mathcal{M}}\) be lower bound of \(\text{conf}(V \rightarrow \rightarrow DB^{r_m})\) for any frequent patterns \(\langle \emptyset, V, r \rangle\) in \(DB^{r_m}\).

6.1.4 Bound the Confidence

The above analysis translates the problem of bounding the lift into the problem of bounding the confidence, which can be defined as:

\[
\text{Problem 6.1. (Bounding the Confidence). For the database } DB, \text{ itemsets } I_0, \text{ and } r, \text{ the problem of bounding the confidence is to find upper and lower bound for the quantity } \text{conf}(I \rightarrow \rightarrow DB) \text{ for any } I \subseteq I_0 \text{ such that } \text{supp}(I \cup \rightarrow \rightarrow DB) \geq s_0.
\]

Note that, \(I_0\)’s for \(\text{bnd}^{r_{DB^r}}_{\mathcal{M}}, \text{bnd}^{r_{DB^{r_m}}}_{\mathcal{M}}, \text{ and } \text{bnd}^{r_{DB^{r_m}}}_{\mathcal{M}}\) discussed above are implicitly defined as all non-\(r\)-items of corresponding databases.

There is a simple solution to this problem. The upper bound of \(\text{conf}(I \rightarrow \rightarrow DB)\) can be chosen as 1. The lower bound can be derived as

\[
\text{conf}(I \rightarrow \rightarrow DB) = \frac{\text{supp}(I \cup \rightarrow \rightarrow DB)}{\text{supp}(I \rightarrow \rightarrow DB)} \geq s_0/|DB|.
\]

However, our goal is to come up with an approach which works significant better than this naive solution.

We introduce the following two lemmas without proof due to limited space:\

**Lemma 6.2.** Given database \(DB\), itemsets \(I_0\), and \(r\), for any \(I \subseteq I_0\), define vector \(V_I\) whose \(x\)-component \(v_{I,x}\) is \(\text{supp}(I \rightarrow \rightarrow DB_{\mathcal{M}[I_{\cup}, \mathcal{M}]}\) and \(y\)-component \(V_{I,y}\) is \(\text{supp}(I \rightarrow \rightarrow DB_{\mathcal{M}[I_{\cup}, \mathcal{M}]}\). We have \(\text{conf}(I \rightarrow \rightarrow DB)\) is equal to slope of \(V_I\) and \(\text{supp}(I \cup \rightarrow \rightarrow DB)\) is equal to \(y\)-component of \(V_I\).

**Lemma 6.3.** Following the settings in lemma 6.2, for any itemset \(A\), we define vector \(V_A\) whose \(x\)-component \(v_{A,x}\) is \(\text{tcnt}(A \rightarrow \rightarrow DB_{\mathcal{M}[I_{\cup}, \mathcal{M}]}\) and \(y\)-component \(v_{A,y}\) is \(\text{tcnt}(A \rightarrow \rightarrow DB_{\mathcal{M}[I_{\cup}, \mathcal{M}]}\). Let

\[
\text{Proofs of lemmas 6.2, 6.3 and theorem 6.4 are included in [46].}
\]
Table 1: Parameters for synthetic dataset.

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>nttrans</td>
<td>number of transactions</td>
<td>100</td>
</tr>
<tr>
<td>nitems</td>
<td>number of items</td>
<td>500</td>
</tr>
<tr>
<td>nptats</td>
<td>number of maximal potentially large itemsets</td>
<td>1000</td>
</tr>
<tr>
<td>patlen</td>
<td>average size of the maximal potentially large itemsets</td>
<td>30</td>
</tr>
<tr>
<td>ttlen</td>
<td>average size of transactions</td>
<td>30</td>
</tr>
</tbody>
</table>

Table 2: Different versions of DLiftMiner algorithms.

<table>
<thead>
<tr>
<th>algorithm</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLiftMiner</td>
<td>applies theorem 6.4 for both upper and lower bounds of confidences</td>
</tr>
<tr>
<td>DLiftMiner-U</td>
<td>identical to DLiftMiner, except confidence upper bound is replaced by 1</td>
</tr>
<tr>
<td>DLiftMiner-N</td>
<td>identical to DLiftMiner except confidence lower bound is replaced by the naive approach</td>
</tr>
<tr>
<td>DLiftMiner-UN</td>
<td>identical to DLiftMiner except confidence lower bound is replaced by zero</td>
</tr>
<tr>
<td>DLiftMiner-L</td>
<td>identical to DLiftMiner except confidence lower bound is replaced by zero</td>
</tr>
</tbody>
</table>

set \( A = \{ A \mid A \in DB(d)_{I_0} \land I \subseteq A \} \), then

\[
V_I = \sum_{A \in A_I} v_A. \tag{15}
\]

Based on lemmas 6.2 and 6.3, Figure 1 illustrates how we can find an efficient solution to problem 6.1. Let us define \( \mathcal{F} \) be the set of vectors \( v_A \) for all \( A \in DB(d)_{I_0} \). Sorting vectors in \( \mathcal{F} \) by slope from large to small and connecting them head to tail gives piece-wise linear curve \( \overline{OBP} \). Similarly, sorting vectors in set \( \mathcal{F} \) by slope from small to large and connecting them head to tail gives piece-wise linear curve \( \overline{OCP} \). If we define another set \( \mathcal{P} \) which contains sum of all possible subsets of vectors in \( \mathcal{F} \), it can be proved that any vectors in set \( \mathcal{P} \) are inside the region enclosed by \( \overline{OBP} \) and \( \overline{OCP} \). Lemma 6.3 suggests that \( V_I \) is also equal to sum of a subset of vectors from \( \mathcal{F} \). So we should have \( V_I \in \mathcal{P} \) and \( V_I \) is inside region \( \overline{OBPCO} \). Line \( y = s_0 \) cuts this region into two parts (we assume that the y-coordinate of point \( P \) is larger or equal to \( s_0 \)). All vectors in \( \mathcal{P} \) whose y-component are larger or equal to \( s_0 \) are inside the top region \( \overline{BCPB} \). From lemma 6.2, any association rule \( I \rightarrow r \) \((I \subseteq I_0)\) whose \( \text{supp}(I \cup r | DB) \) is larger or equal to \( s_0 \) is associated with a vector \( v_I \) in the region \( \overline{BCPB} \) whose slope is \( \text{conf}(I \rightarrow r | DB) \). And it can be easily seen that slopes of all vectors inside region \( \overline{BCPB} \) are upper bounded by the slope of \( \overline{OB} \) and lower bounded by the slope of \( \overline{OC} \). This leads to the conclusion that slopes of \( \overline{OB} \) and \( \overline{OC} \) are upper and lower bounds of \( \text{conf}(I \rightarrow r | DB) \) (Note that slopes of \( \overline{OA} \) and \( \overline{OD} \) are upper and lower bounds provided by naive approach discussed before).

This result is summarized into the following theorem:

**Theorem 6.4.** Following the settings in lemma 6.3, define \( \mathcal{F} = \{ v_A | A \in DB(d)^{I_0} \} \) and sort vectors in \( \mathcal{F} \) based on their slopes from small to large, and label them as \( f_1, f_2, \ldots, f_n \), where \( n \) is the total number of vectors. Define,

\[
b_{I_0, r} = f_1 + f_2 + \cdots + f_{i-1} + \alpha \ast f_i, \tag{16}
\]

where \( 0 < \alpha \leq 1 \), \( 1 \leq i \leq n \) are chosen such that y-component of \( b_i \) is equal to \( s_0 \).

\[
b_{I_0, r} = f_1 + f_2 + \cdots + f_{i-1} + \beta \ast f_i, \tag{17}
\]

where \( 0 < \beta \leq 1 \), \( 1 \leq i \leq n \) are chosen such that y-component of \( b_i \) is equal to \( s_0 \).

Then we have, \( b_{I_0, r} | DB \leq \text{conf}(I \rightarrow r | DB) \leq b_{I_0, r} | DB \), for any \( I \subseteq I_0 \) such that \( \text{supp}(I \cup r | DB) \geq s_0 \).

Our implementation of DLiftMiner algorithm utilizes a slightly modified FPTree data structure. And the bounding approach proposed in theorem 6.4 can be easily applied. This is mainly because each itemset \( A \) in \( DB(d) \) is mapped to a path along the tree excluding leaf nodes. However, due to limited space, we omit the details but include them into [16].

7. EXPERIMENTAL RESULTS

7.1 Synthetic Datasets

DataSets. To evaluate the performance of our algorithm, we use the IBM Quest market-basket synthetic data generator [17] to generate a series of datasets. The generator takes the parameters described in Table 1. We split the generated items into m-items, r-items and r-items from small id’s to large id’s so that m-items cover 57%, r-items cover 33% and r-items cover the rest 10%. We keep only the transactions containing all three types of items in the dataset. The parameters for the first synthetic dataset are shown in the

![Figure 2: Pruning effects on synthetic dataset](image-url)
third column of Table 1 and the actual size of the dataset is 7401 records.

In addition, for scalability analysis, we generated another five datasets by setting ntrans to be 20k, 40k, 60k, 80k and 100k (the other parameters are the same as in Table 1). Due to the fact that only transactions with three types of items are kept, the actual sizes of these datasets are 14824, 29536, 44301, 59036 and 73729.

**Experimental Setup.** The performance of our algorithm was compared against four successively simpler versions of DLiftMiner that incorporate fewer pruning optimizations. These versions are described in Table 2. In our experiment, we set s0 = 100, m0 = 0 and let l0 take the values 2, 3, 4 and 5. And we limited the maximum length of the patterns discovered to 8 (the length is defined to be the sum of sizes of current sets of m-items and v-items).

**Performance of Pruning Methods.** The experimental results are summarized in Table 3 and Figure 2. From Table 3, one can see that the pruning methods incorporated into DLiftMiner lead to dramatic improvements. Our best algorithm DLiftMiner takes only 3.0%, 1.3%, 0.99%, and 0.72% of the running time compared to the baseline approach DLiftMiner-L when l0 takes values 2, 3, 4 and 5, respectively. Even compared with DLiftMiner-UN, where only naive pruning is applied, DLiftMiner takes only about 7% of its running time.

Table 3 also shows how the different pruning methods impact the performance of DLiftMiner. It can be seen that there are dramatic performance improvements when the confidence lower bound is improved (see DLiftMiner-L versus DLiftMiner-UN and DLiftMiner-N versus DLiftMiner-U). However the upper bound of confidence does not seem to play much role in reducing the running time, so that DLiftMiner and DLiftMiner-U (DLiftMiner-N and DLiftMiner-UN) have almost the same results.

Figure 2 plots the number of patterns processed (npattern) versus different pattern length for different versions of the algorithms. The vertical axis has been scaled to log scale (log2 npattern + 1). It can be seen that including confidence upper bound does not help much in pruning additional patterns.

**Scalability Study.** Figure 3 illustrates how the DLiftMiner algorithm scales with respect to the size of input databases for four different l0 values. We set the corresponding mini-
Effect of the Minimum Improvement Threshold. To evaluate how different values of $m_0$ impact the DLiftMiner algorithm, we calculated the numbers of patterns generated by DLiftMiner for different choices of improvements when $l_0$ takes values 2 and 3. These results are summarized in Table 4, in which, $NONE$ means no redundancy is taken into account. One can see that a large fraction of redundant patterns are removed during the mining process when $m_0 = 0$, but setting $m_0$ higher does not reduce as many patterns. However, we are not able to observe significant additional pruning by incorporating the $improvement$ constraint. This may partially due to the fact that our DLiftMiner algorithm does not take into account all redundancy defined in Definition 5.2.

### 7.2 VAERS Dataset

To evaluate the ability of the $dual-lift$ measure to identify interesting patterns, we applied the DLiftMiner algorithm to the VAERS [1] dataset. At the point of this paper, VAERS contains 21 years (from 1990 to 2010) of vaccination reports plus an additional non-domestic dataset. We downloaded all of them. Fields listed in Table 5 were extracted and transformed to the $(M,V,R)$ representation. For example, fields $OTHER_MEDS$, $CUR_ILL$, $HISTORY$, and $PRIOR_VAX$ were converted into a set of historical medical conditions $(m$-items) for each record. The $v$-items and $r$-items were constructed similarly.

One challenge we faced was that the fields we used to extract historical medical conditions or $m$-items were free text and thus not coded. To address this issue, we followed the following process to get a clean set of $m$-items. First, we split the free text into a bag of words (unigrams). Second, in order to capture some important terms beyond unigrams, we split the text by different delimiters: space, comma, semicolon, period and combinations of them. Third, among the set of all terms generated in the above two steps, we kept only those with frequency greater than or equal to 100. Next, we removed those terms that are obviously non-medical related. Finally, when we observed several terms having the same meanings, we mapped them to a normalized term. For example, “allergy codeine”, “allergy to codeine” and “allergic to codeine” were mapped to “codeine allergy”. After cleaning up, the total number of terms we got was 1187. For each record and field, the identified terms that were contained in the textual description were included as $m$-items. For the $r$-items, we removed those symptoms which do not appear very informative, for example, “Unevaluable event”, “Accidental overdose”, “Computerised tomogram normal”, “Drug exposure during pregnancy” and so on. Also we removed all “injection site” symptoms and all symptoms containing the word “negative”.

For our experiment, we set $s_0 = 15$, $l_0 = 3$ and $m_0 = 0$. And we used the DLiftMiner algorithm to generate the following sets of patterns:

- $S_1$ contains all non-redundant frequent patterns $(V,r)$ such that $lift(V \rightarrow r|DB) \geq l_0^2$. $S_1$ is the solution to the traditional adverse reaction detection problem, where historical medical conditions are not taken into account. We found 3617 patterns in $S_1$.
- $S_2$ contains all non-redundant frequent patterns $(M,V,r)$ satisfying minimum dual-lift constraint. $S_2$ is the solution to Problem 5.4. Experiments show that our idea of combining two $lift$ measures with redundancy

### Table 6: Subset of vaccine-symptom associations with medical conditions in $S_3$.

<table>
<thead>
<tr>
<th>Symptom</th>
<th>vaccine code(s)</th>
<th>medical condition(s)</th>
<th>dual-lift</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autism</td>
<td>MMR</td>
<td>HISTORY,&quot;fever&quot;</td>
<td>4.7</td>
</tr>
<tr>
<td>C-reactive protein increased</td>
<td>PNC</td>
<td>HISTORY,&quot;Pregnancy&quot;</td>
<td>9.3</td>
</tr>
<tr>
<td>Drug toxicity</td>
<td>HEP</td>
<td>HISTORY,&quot;Pregnancy&quot;</td>
<td>7.8</td>
</tr>
<tr>
<td>Febrile convulsion</td>
<td>DTPHIB</td>
<td>HISTORY,&quot;seizure&quot;</td>
<td>3.3</td>
</tr>
<tr>
<td>Influenza like illness</td>
<td>LYME</td>
<td>HISTORY,&quot;Anxiety&quot;</td>
<td>3.4</td>
</tr>
<tr>
<td>Injected limb mobility decreased</td>
<td>PPV</td>
<td>CUR_ILL,&quot;allergy&quot;</td>
<td>3.6</td>
</tr>
<tr>
<td>Irritability</td>
<td>HIBV</td>
<td>OTHER_MEDS,&quot;prevacid&quot;</td>
<td>5.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(MMR,OPV)</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(DTP,MMR)</td>
<td>3.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(HIBV,MMR,OPV)</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(DTP,HIBV,MMR)</td>
<td>3.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(DTP,MMR,OPV)</td>
<td>3.9</td>
</tr>
<tr>
<td>Rash vesicular</td>
<td>VARZOS</td>
<td>CUR_ILL,&quot;Hypothyroidism&quot;</td>
<td>3.3</td>
</tr>
<tr>
<td>Swelling</td>
<td>(IPV,MMR)</td>
<td>PRIOR_SYM,&quot;Swelling&quot;</td>
<td>3.8</td>
</tr>
<tr>
<td>Throat tightness</td>
<td>FLU</td>
<td>(HISTORY,&quot;Asthma&quot;,&quot;HISTORY,&quot;allergy&quot;)</td>
<td>3.7</td>
</tr>
<tr>
<td>Urine human chorionic gonadotropin positive</td>
<td>HPV4</td>
<td>HISTORY,&quot;Pregnancy&quot;</td>
<td>8.8</td>
</tr>
<tr>
<td>Varicella</td>
<td>VARCEL</td>
<td>CUR_ILL,&quot;Asthma&quot;</td>
<td>4.2</td>
</tr>
</tbody>
</table>

*Medical conditions are expressed in the form of $FIELD_NAME$"term", where $FIELD_NAME$ is from Table 5.

$PRIOR_SYM$ is symptoms developed from prior vaccination, extracted from $PRIOR_VAX$ field.
is quite selective. We found 169 patterns in $S_2$.

- $S_3$ contains the set of patterns in $S_2$ whose vaccine reaction pairs $(V, r)$ also appear in $S_1$. $S_3$ contains the identified medical conditions under which adverse reactions are more likely to be developed for a subset of patterns in $S_1$. We found 91 patterns $S_3$.

- $S_4$ contains the rest of patterns in $S_2$. $S_4$ is the set of new vaccine adverse reactions that cannot be discovered without incorporation of historical medical conditions. We found 78 patterns in $S_1$.

Due to limited space, we include only some of the patterns and the associated dual-lift values from $S_1$ and $S_4$ into Tables 6 and 7 (see the online supplement \(^5\) for more complete sets of patterns). For each vaccine set $V$, we present the pattern with the highest dual-lift. Note that in Tables 6, 7 (and Tables 9, 10) parenthesis implies vaccine or medical condition interaction. Patterns listed in these two tables can be easily interpreted. For example, the pattern $\langle$OTHER\_MEDS$\text{-}$estradiol$, HPV4$, HOSPITAL$\text{-}$Y$\rangle$ can be interpreted as that individuals taking the medication estradiol are more likely to be hospitalized when vaccinated with HPV4. However, the actual medical significance of these patterns has to be determined by domain experts. To evaluate the generated patterns of our approach, we need a list of vaccine adverse reactions (ideally with associated medical conditions) that have been determined significant by domain experts. The Vaccine Injury Table is the only source of this kind that we are aware of [18]-[19] (see Table 8). The Vaccine Injury Table we are using is from [19] with

\(^\text{3.1}\) Vaccine code(s) (DTAP, HIBV)

\(^\text{3.2}\) $\langle$DTAP, RV$\rangle$

\(^\text{3.3}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.4}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.5}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.6}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.7}\) $\langle$HPV4, MMR$\rangle$

\(^\text{3.8}\) $\langle$HIBV, IPV, MMR$\rangle$

\(^\text{3.9}\) $\langle$HIBV, IPV$\rangle$

\(^\text{3.10}\) $\langle$HIBV, IPV, RV$\rangle$

\(^\text{3.11}\) $\langle$HIBV, MMR$\rangle$

\(^\text{3.12}\) $\langle$HIBV$\rangle$

\(^\text{3.13}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.14}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.15}\) $\langle$HIBV, IPV, MMR$\rangle$

\(^\text{3.16}\) $\langle$HIBV, IPV, RV$\rangle$

\(^\text{3.17}\) $\langle$HIBV, MMR$\rangle$

\(^\text{3.18}\) $\langle$HIBV$\rangle$

\(^\text{3.19}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.20}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.21}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.22}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.23}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.24}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.25}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.26}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.27}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.28}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.29}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.30}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.31}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.32}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.33}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.34}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.35}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.36}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.37}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.38}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.39}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.40}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.41}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.42}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.43}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.44}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.45}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.46}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.47}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.48}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.49}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.50}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.51}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.52}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.53}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.54}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.55}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.56}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.57}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.58}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.59}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.60}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.61}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.62}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.63}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.64}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.65}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.66}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.67}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.68}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.69}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.70}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.71}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.72}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.73}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.74}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.75}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.76}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.77}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.78}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.79}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.80}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.81}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.82}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.83}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.84}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.85}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.86}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.87}\) $\langle$DTAP, IPV, MMR$\rangle$

\(^\text{3.88}\) $\langle$DTAP, IPV, RV$\rangle$

\(^\text{3.89}\) $\langle$DTAP, MMR$\rangle$

\(^\text{3.90}\) $\langle$DTAP, IPV$\rangle$

\(^\text{3.91}\) $\langle$DTAP, IPV, MMR$\rangle$
two patterns are detected, both of which also appear in set of the vaccines appear in the Vaccine Injury Table. Only to use a novel measure called tions as a constraint pattern mining problem. We propose adverse reactions by incorporating historical medical condi-

Next, we present the set of patterns in the Vaccine Injury Table that can be discovered by our method. However, be noted that this evaluation is limited due the small number of cases available. Table 9 contains patterns in set $S_1$ whose symptoms are in Table 8 and vaccine sets have at least one of the reported vaccines for the corresponding symptoms. Our results in Table 9 suggests that vaccine-vaccine interactions play important role when developing these symptoms. Rotavirus vaccine (RV) was licensed on August 31 1998 for routine use in infants in a three-dose series given at two, four and six months of age. In mid-May 1999, nine reports were submitted. On October 14 1999, the vaccine manufacturer voluntarily withdrew its product from the market [20]. Our result in Table 9 shows RV-intussusception can be detected, and it also shows RV can interact with other vaccines and develop intussusception together. Table 10 con-
tains patterns in set $S_2$ whose symptoms and at least one of the vaccines appear in the Vaccine Injury Table. Only two patterns are detected, both of which also appear in set $S_4$. This suggests that by incorporating historical medical conditions, we are likely to identify new patterns other than existing well known ones.

8. CONCLUSIONS

In this paper, we formulate the problem of detecting vaccine adverse reactions by incorporating historical medical condi-
tions as a constraint pattern mining problem. We propose to use a novel measure called dual-lift to evaluate the sign-
ificance of patterns. Our pattern mining algorithm DLift-


Acknowledgment

This work was supported in part by NSF (IIS-0905220, OCI-

REFERENCES


This work was supported in part by NSF (IIS-0905220, OCI-1048018, and IOS-0820730) and the Digital Technology Center and the Minnesota Supercomputing Center and the Minnesota Supercomputing Institute.