# CALCULATING SUPPORT, CONFIDENCE AND LIFT IN MULTI-RELATIONAL XML DATA

**Sikha Bagui*, Sean Spratlin* and Subhash Bagui****

***Abstract:*** *Multi-relational data, when converted to XML, creates repetitive data, hence interestingness measures like support, confidence, and lift cannot be calculated in the same way as they are calculated in regular datasets. In this paper we show how the support, confidence, as well as lift, have to be calculated differently in multi-relational XML data. We tested our algorithm on three different datasets and the results show that thesupport, confidence,and lift measures have to be calculated differently in multi-relational XML data.*

***Keywords:*** *Association rule mining, FP-growth algorithm, XML data, multi-relational database, support, confidence, lift.*

## 1. INTRODUCTION

Association rule mining, introduced by Agrawal *et al.* (1993) in the context of transactional databases, is used for describing interesting patterns in large datasets. Though association rule mining was originally introduced in the context of transactional databases, association rule mining is now used extensively to mine various kinds of datasets, for example, biological data, telecommunications data, census data, etc. In this paper we look at the challenges faced when applying association rule mining to XML data, more specifically, multi-relational XML data.

Association mining algorithms typically generate a large number of patterns. With the size and dimensionality of today's databases, association mining rule algorithms can easily end up with thousands and even millions of patterns, many of which may not be interesting or be of any use. Sifting through patterns using subjective measures to find the interesting ones is not a trivial task, and will often not even be possible, though this may reveal unexpected information about the data or provide useful knowledge. Because so many different frequent patterns or association rules can be derived from any dataset, interest in an association rule is restricted to those rules that apply to a reasonably large number of instances (or transactions) and have a reasonably high accuracy on the instances that they apply to. Two main criteria to measure the strength ofan association rule are in terms of the rule's statistical significance, known as *support* and *confidence* (Han and Kamber, 2012).

Support determines how often a rule is applicable to a given dataset, while confidence determines how frequently items in Y appear in transactions that contain X (Tan, *et al.* 2006). If the support of an itemset meets or exceeds a user-specified minimum support threshold, then the itemset is called a *frequent pattern.* A low support rule is likely to be uninteresting. Confidence ignores the support of the itemset in the rule consequent (Tan, *et al.* 2006).

Since the support-confidence framework is no longer considered sufficient to identify strong association rules (Bagui, *et al.* 2009; Han and Kamber, 2012), we look at an additional objective measure, *lift*, to evaluate the quality of association patternsto augment the support-confidence framework of association rules as applied to multi-relational XML data.

Most data today is in relational databases. Relational databases are composed of groups of

* Department of Computer Science, University of West Florida, Pensacola, FL 32514, *E-mail: bagui@uwf.edu; shs3@students.uwf.edu*

** Department of Mathematics and Statistics, University of West Florida, Pensacola, FL 32514, *E-mail: sbagui@uwf.edu*

related tables or relations, linked via relationships. Each table consists of a number of attributes (columns or fields) and large tables consist of a large number of tuples. Each tuple in a relational table has a unique key that can be used to identify an object with a set of attribute values; hence data in relational databases is considered structured data. Large amounts of data are now being stored in data warehouses. In data warehouse architecture, data is stored in multi-relational tables.But, in today's global business enterprises, as large amounts data travel via the web, data is converted to Extensible Markup Language (XML) and XML has become a major means of data exchange over the web (AliMohammadzadeh, *et al.* 2006; Chen, *et al.* 2005; Ding and Sundarraj, 2006; Feng and Dillon, 2004; Nayak, *et al.* 2002). Hence, data in relational databases and multi-relational databases are converted to XML.

Due to the structure of multi-relational XML data, multi-relational XML data creates its own set of challenges. In this paper we will look at mining multi-relational XML data, specifically, association rule mining multi-relational XML data. Association rule mining depends on the calculations of support, confidence, and lift to present strong association rules. We will look at how calculations of support, confidence, and lift have to be approached differently in multi-relational XML data due to the difference in the nature of multi-relational XML data.

The rest of the paper is organized as follows: Section 2 describes the challenges faced when using multi-relational XML data for association rule mining; Section 3 discusses association rule mining; Section 4 presents related works; Section 5 discusses association rule mining of multi-relational XML data; Section 6 presents the results of our algorithm and Section 7 presents the conclusions.

## 2. CHALLENGES CREATED WHEN USING MULTI-RELATIONAL XML DATA FOR ASSOCIATION RULEMINING

XML data is generally considered semi-structured data since XML data does not usually have a fixed schema and the structure of XML data may be incomplete or irregular. XML data can also be data from various sources like web pages, graphs, geographical data and so on. Data from multi-relational tables, converted to XML, may be considered a more structured form of XML data since the columns and data types may be consistent. XML data from multi-relational tables creates a new set of challenges.

To illustrate the challenges created in using multi-relational XML data, we will present an example using four relational tables.

Suppose we have the following relational tables: STUDENT (stno, sname, major, class, bdate, age, highSchool, campusResident, hrsWorked, GPA); CLASSES (stno, classiID, instrID, ISBN, time, days); INSTRUCTOR (instrID, name, dept, office, ranking); BOOK (isbn, title, author).

We show the tables with some sample data in Figure 1.

**STUDENT**

| STNO | SNAME | MAJOR | CLASS | BDATE | AGE | HIGHSCHOOL | CAMPUSRESIDENT | HRSWORKED | GPA |
|---|---|---|---|---|---|---|---|---|---|
| 2 | Lineas | ENGL | 1 | 1/2/1990 | under_25 | Highland Park | YES | 21_40 | 2 |
| 3 | Mary | COSC | 4 | 7/6/1987 | under_25 | Pensacola | NO | less_10 | 3 |

**CLASSES**

| STNO | CLASSID | INSTRID | ISBN | TIME | DAYS |
|---|---|---|---|---|---|
| 2 | COP3698 | 96 | 2901558609012 | 12 - 1:15 | TR |
| 2 | ART2103 | 70 | 5693256148965 | 8:30 - 9:15 | MW |
| 3 | COP3698 | 96 | 2901558609012 | 5:30 - 8:45 | T |
| 3 | ART2103 | 70 | 5693256148965 | 11:15 - 12:45 | TR |

| INSTRUCTOR | | | | |
|---|---|---|---|---|
| **INSTRID** | **NAME** | **DEPT** | **OFFICE** | **RANKING** |
| 70 | David Ramsey | POLY | 50/127 | Assistant |
| 96 | John Coffey | COSC | 4/4 | Associate |

| BOOK | | |
|---|---|---|
| **ISBN** | **TITLE** | **AUTHOR** |
| 2901558609012 | Data Mining | Jiawei Han |
| 5693256148965 | Materials :Innovation &Design | DimitrisKottas |

**Figure 1: Multi-relational database**

An ER diagram for the multi-relational database in Figure 1 would be as presented in Figure 2.

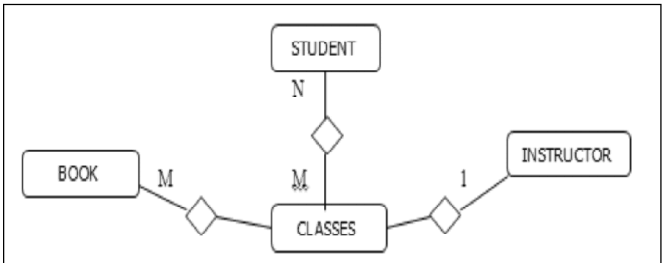


**Figure 2: ER Diagram of Multi-relational Database**

As per the ER diagram presented in Figure 2, a student can take more than one class and a class can have more than one student; one instructor can teach more than one class, and a particular class is taught by one and only one instructor; one class can have more than one book and a book can be used in more than one class.

The multi-relational database presented in Figure 1 was converted to XML using Microsoft Access's XML export function, as shown in Figure 3. This export function has the capabilities to produce not only a Document Type Definition (DTD), but also an Extensible Stylesheet Language (XSL) and an XML Schema Definition (XSD) file. By nature, XML does not contain any formatting information, thus these "languages" provide a means for which other applications can understand its format, structure, and content. Although useful, our proposed Java program does not make use of the above mentioned technologies. Instead, we apply the Document Object Model (DOM) to read in and maintain the XML structure throughout the process.

Even though an XML file does not contain any formatting information, it does however, in our example, have a nested structure which is directly produced from the relationships in the original Access database. These nested XML files can also be produced natively from its original source, not just through the conversion shown here.

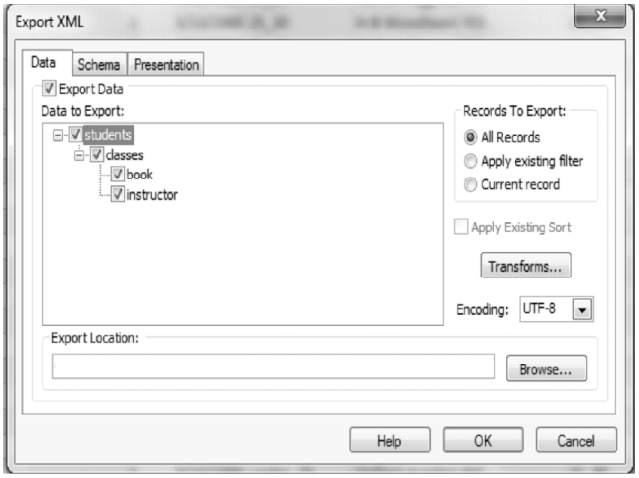


**Figure 3: Export Data to XML**

This produces Listing 1.

```
<student>
        <STNO>2</STNO>
        <SNAME>Lineas</SNAME>
        <MAJOR>ENGL</MAJOR>
        <CLASS>1</CLASS>
        <BDATE>1990-01-02T00:00:00</BDATE>
        <AGE>under_25</AGE>
        <HIGHSCHOOL>Highland Park</
        HIGHSCHOOL>
        <CAMPUSRESIDENT>YES<
        CAMPUSRESIDENT>
        <HRSWORKED>21_40</HRSWORKED>
```

```
<GPA>2</GPA>
<classes>
        <CLASSID>COP3698</CLASSID>
        <TIME>12 - 1:15</TIME>
        <DAYS>TR</DAYS>
        <book>
                <ISBN>2901558609012</ISBN>
                <TITLE>Data Mining</TITLE>
                <AUTHOR>Jiawei Han</AUTHOR>
        </book>
        <instructor>
                <INSTRID>96</INSTRID>
                <NAME>John Coffey</NAME>
                <DEPT>COSC</DEPT>
                <OFFICE>4/4</OFFICE>
                <RANKING>Associate</RANKING>
        </instructor>
</classes>
<classes>
        <CLASSID>ART2103</CLASSID>
        <TIME>8:30 - 9:15</TIME>
        <DAYS>MW</DAYS>
        <book>
                <ISBN>5693256148965</ISBN>
                <TITLE>Materials : Innovation &
                Design</TITLE>
                <AUTHOR>DimitrisKottas</AUTHOR>
        </book>
        <instructor>
                <INSTRID>70</INSTRID>
                <NAME>David Ramsey</NAME>
                <DEPT>POLY</DEPT>

                <OFFICE>50/127</OFFICE>
                <RANKING>Assistant</RANKING>
        </instructor>
</classes>
<classes>
        <CLASSID>INR4523</CLASSID>
        <TIME>2:00 – 3:15</TIME>
        <DAYS>TR</DAYS>
        <book>
                <ISBN>8774596006953</ISBN>
                <TITLE>Chemistry and Chemical
                Reactivity (with CD-ROM)</TITLE>
                <AUTHOR>John Kotz</AUTHOR>
        </book>
        <instructor>
                <INSTRID>114</INSTRID>
                <NAME>Stephen Tanner</NAME>
                <DEPT>CHEM</DEPT>
                <OFFICE>19/84</OFFICE>
                <RANKING>Associate</RANKING>
        </instructor>
    </classes>
</student>
<student>
    <STNO>3</STNO>
        <SNAME>Mary</SNAME>
.
.
.
    <classes>
        <CLASSID>COP3698</CLASSID>
        <TIME>5:30 - 8:45</TIME>
        <DAYS>T</DAYS>
        <book>
                <ISBN>2901558609012</ISBN>
                <TITLE>Data Mining</TITLE>
                <AUTHOR>Jiawei Han</AUTHOR>
        </book>
        <instructor>
                <INSTRID>96</INSTRID>
                <NAME>John Coffey</NAME>
                <DEPT>COSC</DEPT>
                <OFFICE>4/4</OFFICE>
                <RANKING>Associate</RANKING>
        </instructor>
    </classes>
    <classes>
        <CLASSID>ART2103</CLASSID>
        <TIME>11:15 - 12:45</TIME>
        <DAYS>TR</DAYS>
        <book>
                <ISBN>5693256148965</ISBN>
                <TITLE>Materials : Innovation &
                Design</TITLE>
                <AUTHOR>DimitrisKottas</AUTHOR>
        </book>
        <instructor>
                <INSTRID>70</INSTRID>
                <NAME>David Ramsey</NAME>
                <DEPT>POLY</DEPT>
                <OFFICE>50/127</OFFICE>
                <RANKING>Assistant</RANKING>
        </instructor>
    </classes>
</student>
```

Listing 1: XML of multi-relational database

The hierarchical nature of the XML listing presented in Listing 1 is shown in Figure 4.

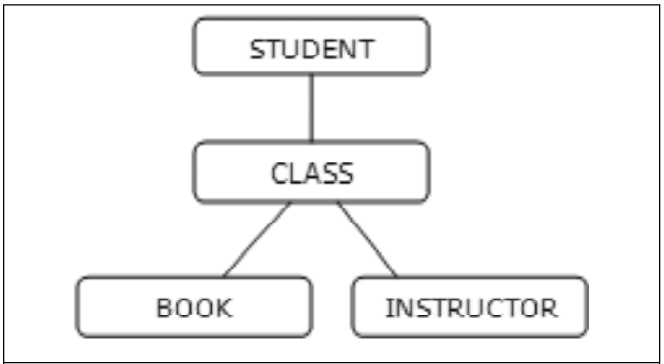


**Figure 4: Hierarchical Relationship of the XML Data**

### 2.1. Repetition within the Data in Xml Format

As illustrated in the above relational tables and corresponding XML, when multi-relational data is converted to XML, the hierarchical nature of XML creates repetitious data.In a multi-relational database, tables are normalized, hence there is supposed to be no repetition in the data except for having the keys of one table in another table as the foreign key, in order to be able to join the tables to get the necessary data or output. So whereas,in a multi-relational database, information on a student would be recorded only once, when the data is converted to XML, for every student, the class that the student takes is listed. For every class (for every student), the book (and details of the book) and the instructor (and details of the instructor) are listed. And this happens for every student for every class…hence the repetition in the data.

For example, the class COP3698 is being taken by both stno 2 and 3, and it is the same instructor, so the instructor information (instrID, name, dept, office and ranking) is listed with both stno 2 as well as 3. Likewise for the ART2103 class which is also taken by both stno 2 and 3. Also, all the fields of the book table are re-listed when the same book is being used. Then there is also the issue of redundancy within each student. Take for instance stno 2, this student isenrolled in three classes, two of these classes have days TR and two have ranking of Associate. Normal calculations would produce a higher count of these items than there were records.

Hence, converting multi-relational databases into XML format creates a lot of duplicate data, and the challenge is,how do we deal with this repetition/duplication when doing the "counts" for association rule mining? Technically, we could have 50 students in the database, but 100 occurrences of an instructor (with all the fields from the respective instructor's table) if each student is taking two classes with the same instructor. How do we deal with the redundant data that is generated when the multi-relational tables are converted to XML when mining association rules? This is the problem that we will be addressing in this paper.

## 3. ASSOCIATION RULES

Association rules are presented in the form $A \Rightarrow B$, where the rule body $A$and the head $B$ are subsets of the set of items $I = \{i_1, i_2, \ldots, i_n\}$ from a set of transactions $D = \{t_1, t_2, \ldots, t_n\}$, where $t_i (i \epsilon [1, N])$ is a transaction and $t_i \subseteq I$,and $A \cap B = \emptyset$. Every subset of $I$ is called an itemset. If an itemset contains $k$ items, then it is called a $k$-itemset.

### 3.1. Support and Confidence

The rule $A \Rightarrow B$ holds in the transaction set $D$ with ***support***$s$, where $s$ is the percentage of transactions in $D$ that contain $A \cup B$ (that is, contain both A and B).This is taken to be the probability, $P(A \cup B)$. (Han and Kamber, 2012).

support $(A \Rightarrow B) = P(A \cup B)$

The rule $A \Rightarrow B$ has ***confidence***$c$ in the transaction set $D$, where $c$ is the percentage of transactions in $D$ containing $A$that also contain $B$. This is taken to be the conditional probability, $P(B \mid A)$. (Han and Kamber, 2012).

$confidence\ (A \Rightarrow B) = P(B \mid A) = support(A \cup B) / support(A) = support\_count(A \cup B) / support\_count(A)$

Rules with high confidence and strong (reasonably large or high) support are referred to as strong rules (Agrawal, *et al.* 1993; Han and Kamber, 2012; Park, Chen and Yu, 1995; Tan, Steinbach and Kumar, 2006). A rule with very low support may occur simply by chance. Confidence, on the other hand, measures the reliability of an inference rule. So, the higher theconfidence, the more likely it is for $B$ to be present in transactions that contains $A$.

### 3.2. Lift

*Lift* computes the ratio between the rules confidence and support of the itemset in the rule consequent. For example, if we want to select samples that will show higher response rates than the rates seen within a general population, *lift* would be a good measure to use. *Lift* measures the change in percent concentration of a desired class, $C_i$, taken from a biased sample, relative to the concentration of $C_i$ within the entire population. We formulate *lift* using conditional probabilities:

$$Lift = P(C_i \mid Sample) / P(C_i \mid Population)$$

where $P(C_i \mid Sample)$ is the portion of instances contained in class $C_i$ relative to the biased sample population and $P(C_i \mid Population)$ is the fraction of

class $C_i$ instances relative to the entire population (Roiger, and Geatz, 2003).

We define the *lift* measure, as:

If we have two itemsets, A and B, we will define the dependency criterion, *lift*, as:

$Lift(A, B) = P(A \cup B) \ / \ P(A)P(B)$

P is the probability, so, given two itemsets, A and B, *lift* is defined as the probability of A and B occurring together divided by the probability of A multiplied by the probability of B.

If the resulting value of above equation is less than 1, then there is a negative dependency between itemsets A and B. If the resulting value is greater than 1, then itemsets A and B are positively dependent on each other, meaning that the occurrence of one implies the occurrence of the other. If the resulting value is equal to 1, than A and B are independent.

## 4. RELATED WORKS

A few works have looked at mining association rules in XML data. Paik, *et al.* (2009) looked at mining association rules in tree structured XML data. Paik, *et al.*'s (2009) approach reduced the number of combinatorial times for obtaining desirable rules and simplified the extraction process.Wan and Dobbie (2004) showed how to extract association rules from XML documents using XQuery and analyzed the XQuery implementation of the Apriori algorithm. Romei and Turini (2010) proposed a general purpose query language in support of mining XML data. Feng and Dillon (2004) presented a technique for template-guided mining of association rules from large XML data. They extended the notion of associated items to XML fragments to present associations among tree-structured items rather than simple-structured items of atomic value. Ding and Sundarraj (2006) presented a Java based implementation for mining XML data that compared two well-known algorithms for association rule mining: Apriori (Agrawal and Srikant, 1994) and Frequent Pattern Growth (FP-growth) (Han, *et al.* 2004). These works mainly focused on how to do association mining in XML datasets and did not address the problem of duplicate data in multi-relational XML datasets.

Interestingness measures in association rule mining were also studied by a few. Tan *et al.* (2002) provided an overview and comparison of the different interestingness measures, and situations for which the interestingness measures are best suited. Bayardo and Agrawal (1999) showed that the most interesting rules reside along the support/ confidence border for a variety of metrics including support, confidence and lift. Geng and Hamilton (2006) reviewed interestingness measures for rules and classified them from several perspectives, compared their properties and identified their role in the data mining process. Lallich, *et al.* (2007) looked at interesting boolean association rules. Omiecinski (2003) discussed three alternative interestingness measures for associations: any-confidence, all-confidence, and bond. They proved that the important downward closure property applied to both all-confidence and bond, but did not hold for any-confidence. Wu, *et al.* (2010) re-examined a set of null-invariant interestingness measures and found that they can be expressed as the generalized mathematical mean, leading to a total ordering of interestingness measures. They proposed a new measure called *Imbalance Ratio* to gauge the degree of skewness of a data set. Though these works covered interestingness measures of association rules including the support–confidence framework, and some covered additional interesting measures like lift, these works worked with regular datasets, not XML data specifically.

There are also a few works that focused on different aspects of the interestingness measure, lift. Hahsler, *et al.* (2005) explored the ability to filter noise of confidence and lift and used this framework to developa new interesting measure hyperlift, which they then compared with regular lift. All their comparisons were on transactions data.Messaoud, *et al.* (2006) proposed a framework for mining inter-dimensional association rules from data cubes. Messaoud, *et al.* (2006) evaluated the interestingnessof mined association rules according to lift and other interestingness criteria. Bagui, *et al.* (2009) presented an algorithm that does not require more than one full scan to derive strong association rules using the lift measure. Nicholas, *et al.* (2008) discussed standardizing lift of an association rule. The upper and lower bounds of lift were used to standardize the lift measure.

The above works on lift, as well as the works on the support-confidence framework, were on regular datasets, not XML datasets. In this paper we address the issue of how interestingness measures, specifically support, confidence, and lift, have to be handled in multi-relational XML data, which has a different nature.

## 5. ASSOCIATION MINING MULTI-RELATIONAL XML DATA

Association rule mining algorithms can broadly be divided into two categories: Apriori and FP-growth. Apriori uses an iterative approach to mine a dataset where *k*-itemsets are used to explore (*k*+1)-itemsets. There are two major drawbacks of the Apriori algorithm: (i) the Apriori algorithm scans the input dataset *k* times to create frequent *k*-itemsets; (ii) a large number of candidate itemsets are generated. Scanning the datasets in the multi-relational XML scenario would be even more computationally expensive since multi-relational XML data typically has more data due to repetition.

The FP-growth algorithm, in contrast to the Apriori algorithm, scans the dataset twice and does not require the overhead of the candidate itemset generation that the Apriori algorithm requires, hence is faster than the Apriori algorithm (Ding and Sundarraj, 2006; Han and Kamber, 2012), hence we used the FP-growth algorithm for this work.

### 5.1. FP-growth Algorithm

In the FP-growth algorithm, the first scan of the database is the same as the Apriori algorithm. The database is scanned and the frequency (support counts) of each item is recorded. Anything below the user support is deleted from this frequent item set and all the frequent items equal to or above the support count are stored in a virtual table that holds both the item and its support count. These items are then ordered in descending order based on their support counts.

The second pass of the FP-growth algorithm scans the database once more, this time ordering the items according to their order within thevirtual table. Thus the most frequent items reside closest to the root,while less frequent items progressively get further away. Once the items are placed in the tree structure based on this ordering, rule mining is performed.

#### *5.1.1. Steps of the FP-growth Algorithm*

Below we present the steps we used in constructing the FP-growth algorithm. Our Java based program is a modified version of the FP-growth algorithm presented by Han and Kamber (2012).

The Method

1. The FP-tree is constructed as follows.

```
Scan the XML document, D, once, collecting all items. Collect F, the frequent
  items, T the tags, and their respective counts for each.
Separate each transaction Trans in D (defined by checking the opening tag and
  the corresponding closing tag of the document structure) to do the following.
     Create the root of an FP-Tree and label it "Null"
     Order F in descending support count order as L.
     Call createTree([p/P]), which is performed as follows.
         Select and sort F in Trans according to the order of L as R.
         Let the sorted frequent item list, R, in Trans be [p/P], where p is
         the first element and P is the remaining list.
         Call insert_tree([p/P]), which is performed as follows.
              If node N has a child C such that C.item-name=p.item-name
                   Then increment C's count by 1
              Else
                   Create a new node M, and let its count be 1, its parent
                   link be linked to N, and its node-link to the nodes with
                   the same item-name via the node-link structure.
              If P is nonempty, call insert_tree(p/P) recursively.
```

2. The FP-Tree is mined by calling getSupport(support) as implemented below:

```
For each Pathin PathList
    While each path is != root
         Add items to generated array
         If items exists
              Increment count of item with end items support count
If newPath size is > 1
    Check support_count is > user defined support
    If support_count < support
         Remove generated path
    Else call calculate(paths,allIitems, support, tags)

Calculate(paths, items, support, tags)
    For each path canidates
         Create string representation of nodes in form (text, text, text)
    For each item in D
         Separate Trans in D
              getSupport(cand, Trans)
         aSupport += getSupport of each Trans for candString
    For each tag
         get lowest tag of A
         get lowest tag of B
         set (A ∪ B) support to end item support count
         get lower tag count of (A ∪ B)
```

### 5.2. Calculating the Support, Confidence and Lift in Multi-relational XML Data

Since multi-relational XML data creates inherent repetitive data, as explained in section two of this paper, the support, confidence, as well as lift, have to be calculated differently in multi-relational XML data.We will illustrate this through an example.

Suppose we have a piece of multi-relational XML data (as shown in Listing 1). For explanation's sake we present,in tabular format, a subset of the data from Table 1.

**Table 1**
**XML Data Read into Table Format for Association Rule Mining**

| *STNO_ID* | *Itemsets* |
|---|---|
| STNO1 | {T, COP3698, 2}, { John Coffey, Data Mining} |
| STNO4 | {M, COP6990}, {T, COP3698} |
| STNO5 | {Associate, John Coffey}, {John Coffey, Associate, COP3698, 2} |
| STNO6 | {John Coffey, Data Mining}, {John Coffey, COP3698} |
| STNO8 | {T, John Coffey, 2} |
| STNO10 | {T, Data Mining, Associate}, {T, John Coffey, Associate, 3.5} |
| STNO12 | {T, COP3698, Associate}, {T, John Coffey, Associate, 3.5} |
| STNO13 | {John Coffey, Data Mining, 2} |
| STNO15 | {T, Data Mining, Associate}, {T, COP3698} |
| STNO16 | {T, COP3698, Associate} |
| STNO22 | {John Coffey, COP3698}, {John Coffey, Data Mining}, {T, COP3698} |
| STNO23 | {John Coffey, COP3698}, {T, COP3698, Associate} |
| STNO24 | {John Coffey, COP3698}, {T, COP3698, 2} |
| STNO25 | {John Coffey, COP3698}, {F, COP6990} |
| STNO28 | {John Coffey, COP3698}, {T, John Coffey, 2} |
| STNO29 | {T, John Coffey}, {T, COP3698} |
| STNO31 | {T, John Coffey}, {T, Data Mining} |
| STNO32 | {Kevin Baker, COP3698}, {T, Data Mining} |

The first step would be to get the support counts of the one-itemsets. Assume the ordered one-itemsets are:

```
Days:T = 19
Name:John Coffey = 18
ClassID:COP3698 = 17
Rank:Associate = 9
Title:Data Mining = 8
Hours_Worked:2= 6
ClassID: COP6990 = 2
GPA:3.5 = 2
Days:M = 1
Days:F = 1
Name: Kevin Baker = 1
```

Simultaneously we also obtain the support counts of the tags. Assume that the tags and their respective counts are:

```
Name = 19
Days = 21
Title = 8
ClassID = 19
Rank = 9
Hours_worked = 6
GPA = 2
```

Then, suppose we kept itemsets with support counts greater than or equal to 2, we would have:

```
Days:T = 19
Name:John Coffey = 18
ClassID:COP3698 = 17
Title:Data Mining = 8
Rank:Associate = 9
Hours_Worked:2 = 6
ClassID: COP6990 = 2
GPA:3.5 = 2
```

Next we present the itemsets with support counts greater than or equal to 2, ordered by the support counts (Table 2).

**Table 2**
**Itemsets >=2, ordered by the Support Counts**

| *STNO_ID* | *Itemsets* |
|---|---|
| STNO1 | {T, COP3698, 2}, { John Coffey, Data Mining} |
| STNO4 | {T, COP3698} |
| STNO5 | {John Coffey, Associate}, {John Coffey, Associate, COP3698, 2} |
| STNO6 | {John Coffey, Data Mining}, {John Coffey, COP3698} |
| STNO8 | {T, John Coffey, 2} |
| STNO10 | {T, Associate, Data Mining }, {T, John Coffey, Associate} |
| STNO12 | {T, COP3698, Associate}, {T, John Coffey, Associate} |
| STNO13 | {John Coffey, Data Mining, 2} |
| STNO15 | {T, Associate, Data Mining }, {T, COP3698} |
| STNO16 | {T, COP3698, Associate} |
| STNO22 | {John Coffey, COP3698}, {John Coffey, Data Mining}, {T, COP3698} |
| STNO23 | {John Coffey, COP3698}, {T, COP3698, Associate} |
| STNO24 | {John Coffey, COP3698}, {T, COP3698, 2} |
| STNO25 | {John Coffey, COP3698} |
| STNO28 | {John Coffey, COP3698}, {T, John Coffey, 2} |
| STNO29 | {T, John Coffey}, {T, COP3698} |
| STNO31 | {T, John Coffey}, {T, Data Mining} |
| STNO32 | {COP3698}, {T, Data Mining} |

From the above itemsets, the FP-growth tree, shown in Figure 5, was generated.

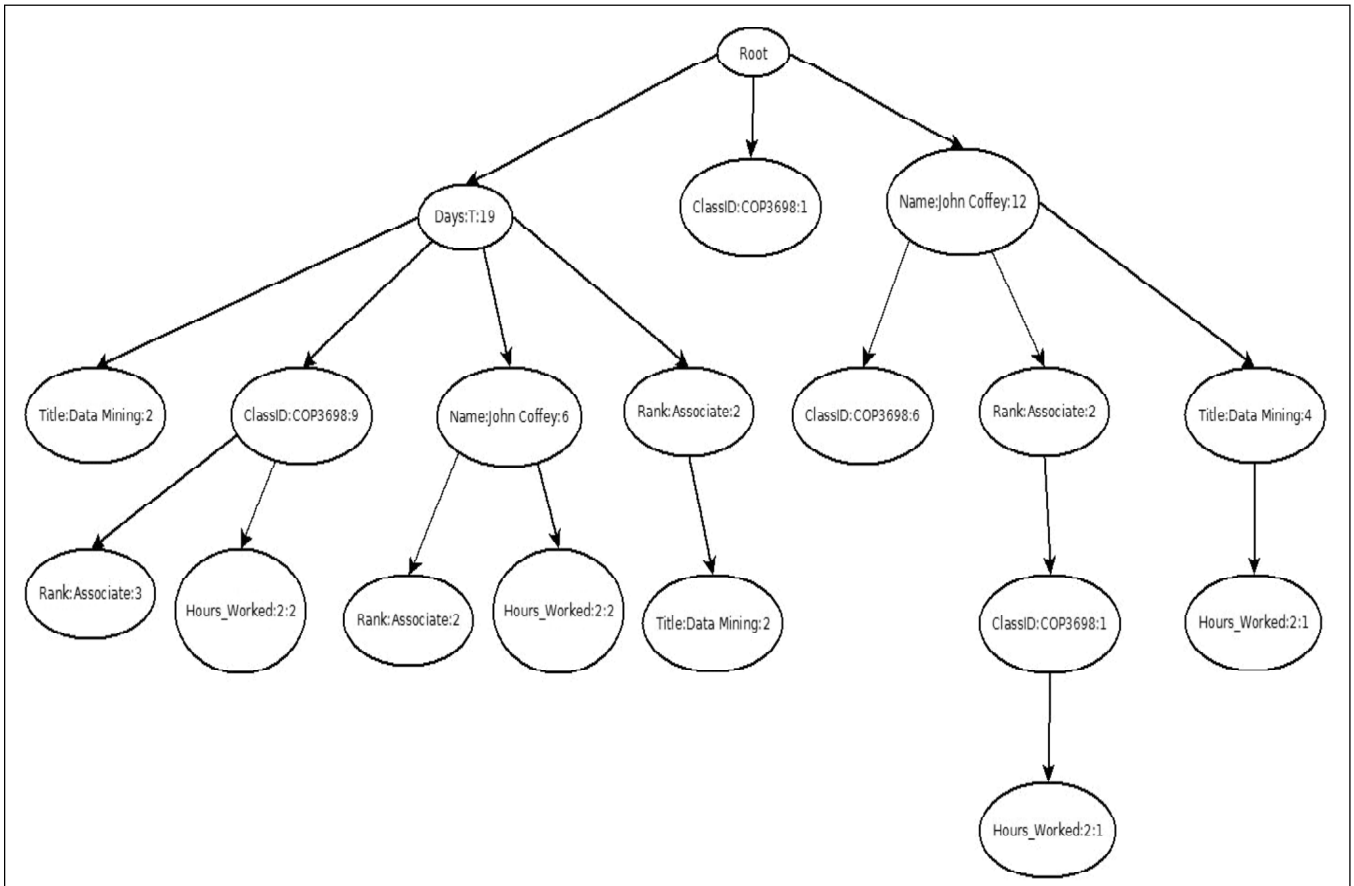


**Figure 5: FP-growth Tree**

The paths generated from the FP-growth tree would be:

```
{T, COP3698, Associate: 3}
{T, COP3698, 2: 2}
{T, John Coffey, 2: 2}
{T, Data Mining: 2}
{T, John Coffey, Associate: 2}
{John Coffey, COP3698: 6}
{T, Associate, Data Mining: 2}
{COP3698: 1}
{John Coffey, Data Mining, 2: 1}
{John Coffey, Associate, COP3698, 2: 1}
```

Keeping the frequent paths with minimum support count >=2 we get:

```
{T, COP3698, Associate: 3}
{T, COP3698, 2: 2}
{T, John Coffey, 2: 2}
{T, Data Mining: 2}
{T, John Coffey, Associate: 2}
{John Coffey, COP3698: 6}
{T, Associate,Data Mining: 2}
```

To illustrate the modified calculations of the support, confidence and lift measures in multi-relational XML data, we use the following rule to show the adjusted support, adjusted confidence and adjusted lift:

{John Coffey, COP3698: 6}or (John Coffey $\Rightarrow$ COP3698)

### *5.2.1. Calculating Adjusted Support*

Support is defined as the percent of occurrences in a dataset that contain both A and B, or the probability of (A $\cup$ B). Strictly using this formula in a multi-relational XML dataset might give us results greater than 1, since in multi-relational XML datasets, data is often replicated for each entity or object. Hence, something may appear to occur more times than the number of entities or objects, which

is not possible. To resolve this issue, in multi-relational XML datasets, rather than finding the percent of occurrences in the dataset, we need to find the number of times A and B occur together within the total number of times A's tag occurs and the total number of times B's tag occurs. This can only be a maximum of the minimum of the tag counts of tag A and tag B. Hence, adjusted support in an XML multi-relational dataset would be calculated as follows:

```
Adjusted Support = P (AB) / (Lower of
the tag count of A ∪ B)
      = P (John Coffey ∪ COP3698) /
(Lower of the tag count of Name ∪ ClassID)
      = 6 / 19
      = 0.315789 or 31.58%
```

This would translate to: About 1/3 of the time (John Coffey ∪ COP3698) occur together given the occurrences of the Name and ClassID tags.

### 5.2.2. *Calculating Adjusted Confidence*

Confidence of an association rule is defined as the percentage of the number of transactions that contain (A ∪ B) given the total number of transactions that contain A. Confidence is a measure of strength of the association rule. Suppose the confidence of the association rule A ⇒ B is 80%, this means that 80% of the transactions that contain A also contain B, when A is present. Hence, the adjusted confidence would be:

```
Adjusted Confidence = Adjusted support(A
∪ B) / Adjusted support (A)
      = (P (John Coffey ∪ COP3698) /
(Lower of the tag count of Name ∪
ClassID)) / (tag_count of (Name))
= 0.315789 / (18/19)
  = 0.3333
```

This would mean that about 1/3 of the time John Coffey and COP3698 would occur together given that the instructor is John Coffey.

### 5.2.3. *Calculating Adjusted Lift*

Lift is defined as the probability of (A ∪ B) occurring together divided by the probability of A multiplied by the probability of B. *P* (A ∪ B) was explained in the section on calculating support.But, in a regular dataset the probability of A would be defined as the support count of A divided by the number of entities or objects. In a multi-relational XML dataset, however, the probability of A would be defined as the support count of A divided by the tag count of A, and likewise the probability of B would be defined as the support count of B divided by the tag count of B. Hence the calculation of the adjusted lift would be:

```
Adjusted Lift = P (AB) / P(A)P(B)
      = P (John Coffey COP3698) /
P(John Coffey) * P(COP3698)
      = (P (John Coffey COP3698) /
(Lower of the tag count of Name
ClassID)) / (support_count(John
Coffey)/tag_count of (Name) *
      (support_count(COP3698)/
tag_count of (ClassID)
      = (0.315789) / (18/19)*(17/19)
      = 0.372549
```

Since this value is less than 1, there is a negative dependency between the two itemsets, John Coffey and COP3698.

Using the above formulas, the adjusted support, confidence, and lift measures for the rest of the rules would be:

**Table 3**
**Adjusted Support, Adjusted Confidence, Adjusted Lift of the Rest of the Rules**

| | *Adjusted support* | *Adjusted confidence (if A is the first item)* | *Adjusted confidence (if A is all but the last item)* | *Adjusted Lift* |
|---|---|---|---|---|
| {T, COP3698, Associate: 3} | 33.33% | 36% | 69.67% | .41 |
| {T, COP3698, 2: 2} | 33.33% | 36% | 69.67% | .41 |
| {T, John Coffey, 2: 2} | 33.33% | 36% | 100% | .389 |
| {T, Data Mining: 2} | 25% | 27.63% | 27.63% | .276 |
| {T, John Coffey, Associate: 2} | 22.22% | 24.56% | 70.37% | .259 |
| {T, Associate, Data Mining: 2} | 25% | 27.63% | 100% | .276 |

## 6. TESTING OUR ALGORITHM

In order to test our algorithm, we ran our algorithm on three different datasets: (i) Northwind [24]; (ii) Mondial [25]; and (iii) a multi-relational synthetic XML dataset, which we will call Our_Student_DB. For each database we tabulated counts of itemsets A, B, C and D, thesupport counts, the unadjusted support, adjusted support, adjusted confidence, unadjusted lift, and adjusted lift.

The unadjusted support is (Han and Kamber, 2012): *support* $(A \Rightarrow B) = P(A \cup B)$. Here we did not take into account the tag counts for XML data. The adjusted support is the support calculated based on our formula presentedin Section 5.2.1, taking into account the XML tag counts.

The adjusted confidence is the confidence calculated based on our formula presented in Section 5.2.2.

The unadjusted lift is (Han andKamber, 2012): *Lift* $(A, B) = P(A \cup B) / P(A)P(B)$. The adjusted lift is calculated based on the formula presented in Section 5.2.3, taking into account the XML tag counts.

In the following sections we present the results for each database.

### 6.1. The Northwind Database

The Northwind database, packaged with Microsoft Office suite, is a real world-like multi-relational database with the names of companies, products, employees, and other data. It contains the sales data for a fictitious company called Northwind Traders, which imports and exports specialty foods from around the world [24]. The database has 18 tables. Each table has several attributes and relationships to other tables. For this project, we joined all tablesthat are directly or indirectly related to the CUSTOMER table, which we then converted to XML using the process explained in Section 2 ofthe paper. In Table 4 we present sample results of running the XML Northwind database on our algorithm. The first seven rows present 2 itemset cases ($A \Rightarrow B$), the next three rows present 3 itemset cases ($A, B \Rightarrow C$), and the last 5 rows present 4 itemset cases ($A, B, C \Rightarrow D$).

From these results we can see that the unadjusted support is greater than 100%, which it cannot be. This is because of the multiple counting of data when multi-relational databases are converted to XML. The adjusted support is 100% or close, which it should be. Where the support is

**Table 4**
**Sample Results of the Northwind Database**

| *A* | *B* | *C* | *D* | *Support Count* | *Unadjusted Support (%)* | *Adjusted Support (%)* | *Adjusted Confidence (%)* | *Unadjusted Lift* | *Adjusted Lift* |
|---|---|---|---|---|---|---|---|---|---|
| 48 | 48 | | | 48 | 165.52 | 100.00 | 100.00 | 60.417 | 1.091 |
| 101 | 48 | | | 48 | 165.52 | 100.00 | 47.50 | 28.713 | 1.091 |
| 104 | 48 | | | 48 | 165.52 | 100.00 | 46.20 | 27.885 | 1.091 |
| 269 | 48 | | | 48 | 165.52 | 100.00 | 17.80 | 10.781 | 1.091 |
| 464 | 48 | | | 48 | 165.52 | 100.00 | 10.30 | 6.250 | 1.091 |
| 607 | 48 | | | 48 | 165.52 | 100.00 | 7.90 | 4.778 | 1.468 |
| 47 | 48 | | | 47 | 162.07 | 97.92 | 100.00 | 60.417 | 1.091 |
| 48 | 48 | 48 | | 48 | 165.52 | 100.00 | 100.00 | 60.417 | 1.091 |
| 87 | 269 | 48 | | 48 | 165.52 | 100.00 | 55.20 | 33.333 | 1.266 |
| 87 | 464 | 48 | | 48 | 165.52 | 100.00 | 55.20 | 33.333 | 1.266 |
| 44 | 48 | 607 | 48 | 44 | 151.72 | 91.67 | 100.00 | 60.417 | 1.091 |
| 87 | 104 | 48 | 48 | 48 | 165.52 | 100.00 | 55.20 | 33.333 | 1.266 |
| 87 | 104 | 464 | 48 | 48 | 165.52 | 100.00 | 55.20 | 33.333 | 1.266 |
| 86 | 104 | 607 | 48 | 48 | 165.52 | 100.00 | 55.80 | 33.721 | 1.281 |
| 44 | 48 | 48 | 48 | 44 | 151.72 | 91.67 | 100.00 | 60.417 | 1.091 |

100%, this means that 100% of the time itemsets A and B occur together.

The confidence of 100% would mean that itemsets A and B occur together 100% of the time given itemset A. Using the adjusted confidence formula, we did not get any confidence numbers above 100%.

The unadjusted lift results are exceptionally high since the tag counts are not taken into account. The adjusted lift takes the tag counts into account, so the results are more reasonable. The positive lift numbers reflect a positive relationship between the itemsets.

### 6.2. The Mondial Database

The Mondialmulti-relational database has been compiled from geographical web data sources. It has information on small island countries, geographical modeling of rivers, lakes, seas, islands, mountains, deserts, sources of estuaries of rivers, etc [25]. It has 16 tables and each table has several attributes. After joining several tables, we ended up with 94211 records, which we then converted into XML format using the process explained in Section 2 of the paper. In Table 5 we present sample results of our algorithm on this XML formatted Mondial database.

**Table 5**
**Sample Results of the Mondial Database**

| *A* | *B* | *Support Count* | *Unadjusted Support (%)* | *Adjusted Support (%)* | *Adjusted Confidence (%)* | *Unadjusted Lift* | *Adjusted Lift* |
|---|---|---|---|---|---|---|---|
| 844 | 79 | 79 | 34.20 | 2.72 | 9.40 | 27.37 | 3.731 |
| 421 | 80 | 80 | 34.63 | 5.50 | 19.00 | 54.869 | 6.910 |
| 421 | 80 | 80 | 34.63 | 5.50 | 19.00 | 54.869 | 6.910 |
| 421 | 82 | 82 | 35.50 | 2.82 | 19.50 | 54.869 | 7.480 |
| 421 | 87 | 87 | 37.66 | 2.99 | 20.70 | 54.869 | 7.480 |
| 421 | 100 | 100 | 43.29 | 3.44 | 23.80 | 54.869 | 7.480 |
| 421 | 176 | 176 | 76.19 | 6.05 | 41.80 | 54.869 | 7.480 |
| 421 | 177 | 177 | 76.62 | 6.08 | 42.00 | 54.869 | 7.480 |
| 421 | 140 | 140 | 60.61 | 4.81 | 33.33 | 60.61 | 6.190 |
| 442 | 243 | 243 | 105.91 | 8.35 | 55.00 | 52.262 | 7.124 |
| 844 | 243 | 243 | 105.91 | 8.35 | 28.88 | 27.370 | 3.731 |
| 844 | 311 | 311 | 134.63 | 10.69 | 36.88 | 27.370 | 3.731 |

The unadjusted support of the last three rows is above 100%. This shows that the unadjusted support calculations cannot be used to calculate the support of multi-relational XML databases. In this set of results too, the adjusted lift numbers are again more reasonable than the unadjusted lift numbers, and we did not get any adjusted confidence numbers above 100%.

### 6.3. Our_Student_Db Database

This multi-relational database schema has been explained earlier in Section 2 of this paper. In Table 6 we present the results of running our algorithm on this simulated database. The first nine rows present 2 itemset cases (A $\Rightarrow$ B), the next four rows present 3 itemset cases (A, B $\Rightarrow$ C), and the last 2 rows present 4 itemset cases (A, B, C $\Rightarrow$ D).

From the unadjusted support results of Our_Student_DB, we can see that there is one instance of a support over 100%, which it cannot be. The corresponding adjusted support (using the tag counts) for that rule is 55.79%, so 55.79% of the time itemsets A and B occur together. So, the adjusted support calculations should be used for the support measure rather than the unadjusted support calculations.

Here too, the adjusted lift results are more realistic than the unadjusted lift results. This is because the adjusted lift takes into account the tag counts.

**Table 6**
**Sample Results of Our_Student_DB**

| *A* | *B* | *C* | *D* | *Support Count* | *Unadjusted Support (%)* | *Adjusted Support (%)* | *Adjusted Confidence (%)* | *Unadjusted Lift* | *Adjusted Lift* |
|---|---|---|---|---|---|---|---|---|---|
| 163 | 75 | | | 75 | 43.69 | 19.74 | 46.00 | 104.908 | 2.331 |
| 213 | 75 | | | 75 | 43.86 | 19.74 | 35.20 | 80.282 | 1.784 |
| 163 | 83 | | | 83 | 48.54 | 21.84 | 50.90 | 104.908 | 2.331 |
| 213 | 83 | | | 83 | 48.54 | 21.84 | 39.00 | 80.282 | 1.784 |
| 122 | 83 | | | 83 | 48.54 | 21.84 | 68.00 | 140.164 | 3.115 |
| 118 | 102 | | | 102 | 59.65 | 26.84 | 86.40 | 144.915 | 3.220 |
| 213 | 102 | | | 102 | 59.65 | 26.84 | 47.90 | 80.282 | 1.784 |
| 163 | 102 | | | 102 | 59.65 | 26.84 | 62.60 | 104.908 | 2.331 |
| 213 | 213 | | | 213 | 123.98 | 55.79 | 99.50 | 79.905 | 1.784 |
| 105 | 118 | 102 | | 102 | 59.65 | 26.84 | 97.10 | 162.857 | 3.619 |
| 105 | 118 | 102 | | 102 | 59.65 | 26.84 | 97.10 | 162.857 | 3.619 |
| 105 | 213 | 102 | | 102 | 59.65 | 26.84 | 97.10 | 162.857 | 3.619 |
| 118 | 118 | 102 | | 102 | 59.65 | 26.84 | 86.40 | 144.915 | 3.220 |
| 105 | 102 | 118 | 102 | 102 | 59.65 | 26.84 | 97.10 | 162.857 | 3.619 |
| 105 | 102 | 213 | 102 | 102 | 59.65 | 26.84 | 97.10 | 162.857 | 3.619 |

## 7. CONCLUSION

In this paper, we formalized the problem of calculating interestingness measures on multi-relational XML data. We first introduced the challenges faced in XML data. We then focused on the structure of the data as well as explaining the measures used in our computation. Then, as shown in the results section of our paper, since multi-relational XML data has inherent repetitive data, the support, confidence, as well as lift measures,have to be calculated differently. The support, confidenceand lift measures have to be calculated using the proposed respective tag counts rather than the number of records or entities (or objects). The test runs on the sample databases show that, if the tags counts are not taken into consideration, we get support numbers above 100% (as shown by the unadjusted support columns), which is not possible, and lift numbers are also often unreasonably high. The adjusted confidence also does not calculateabove 100% using the adjusted confidence formula.

In future work, we plan to see if other interestingness measures used in association rulemining, forexample, cosine, the chi-square measure, theKulczynski measure, etc. would be affected in the same way when multi-relational XML data is being analyzed.

### *Acknowledgements*

The authors would like to thank the Editor, Dr. John Wang, and the referees of this paper for their constructive suggestions, which led to the present improved version of this paper.

### *References*

[1] AliMohammadzadeh, R., Soltan, S., Rahgozar, M. (2006), 'Template Guided Association Rule Mining from XML Documents'. *Proceedings of WWW*, 963-964.

[2] Agrawal, R., Imielinski, T. and Swami, A. (1993), 'Mining Association Rules between Sets of Items in Large Databases'. *ACM SIGMOD Conference*, ACM Press, 207-216.

[3] Agrawal, R. and Srikant, R. (1994), 'Fast Algorithms for Mining Association Rules'. *Proceedings of the Twentieth International Conference on Very Large Data Bases*, Santiago, Chile, 487-499.

[4] Bagui, S. Just, J., and Bagui, S. (2009), 'Deriving Association Mining Rules Using a Dependency Criterion, the Lift Measure'. *International Journal of Data Analysis Techniques and Strategies,* 1(3), 297-312.

[5] Bayardo, R. J., and Agrawal, R. (1999), 'Mining the Most Interesting Rules'. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 145-154.

[6] Chen, L., Bhowmick, S. S., and Chia, L-T. (2005), 'Mining Positive and Negative Association Rules from XML Query Patterns for Caching'. *Proceedings of the 10th International Conference on Database Systems for Advanced Applications (DASFAA).*

[6] Ding, Q., and Sundarraj, G. (2006). 'Association Rule Mining from XML Data'.*Conference on Data Mining,* Las Vegas, Nevada, 144-150.

[7] Feng, L., and Dillon, T. (2004), 'Mining XML-Enabled Association Rule with Templates'. *Proceedings of KDID04.*

[8] Geng, L., and Hamilton, H. J. (2006), 'Interestingness Measures for Data Mining: A Survey'. *ACM Computing Surveys*, 38(3), 1-32.

[9] Hahsler, M., Hornick, K., and Reutterer, T. (2005), 'Implications of Probabilistic DataModeling for Rule Mining.Research Report Series'. Retrieved, 5/31/2012, *http://statistik.wu-wien.ac.at/*

[10] Han, J. and Kamber, M. (2012), *Data Mining: Concepts and Techniques.* Morgan Kaufmann Publishers, USA.

[11] Han, J., Pei, J., Yin, Y. and Mao, R. (2004), 'Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach', *Data Mining and Knowledge Discovery,* 8(1), 53-87.

[12] Lallich, S., Teytaud, O., and Prudhomme, E. (2007), 'Association Rule Interestingness: Measure and Statistical Validation'. *Quality Measures in Data Mining*, 43, 251-275.

[13] Messaoud, R. B., Rabaseda, S. L., Boussaid, O., Missaoui, R. (2006), 'Enhanced Mining of Association Rules from Data Cubes'. *DOLAP*, Arlington, VA, 11–18.

[14] Nayak, R., Witt, R., and Tonev, A. (2002), 'Data Mining and XML Documents', *Proceedings of International Conference on Internet Computing,* IC, Las Vegas, Nevada, (3), 660-666.

[15] Nicholas, P. D., Murphy, T. B., and Regan, M. O. (2008), 'Standarding the Lift of an Association Rule'.*Computational Statistics & Data Analysis,* 52(10),4712-4721.

[16] Omiecinski, E. (2003), 'Alternative Interest Measures for Mining Associations'.*IEEE Transactions on Knowledge & Data Engineering,* 15(1), 57–69.

[17] Paik, J., Nam, J., Kim, W. Y., Ryu, J. S., Kim, U. M. (2009), 'Mining Association Rules in Tree Structured XML Data'. *ICIS*, November 24-29, 2009, Seoul, Korea, 807-811.

[18] Park, J. S., Chen, M. –S., and Yu, P. S. (1995), 'An Effective Hash based Algorithm for Mining Association Rules'. *Proceedings of the 1995 ACM SIGMOD Conference*, ACM Press, 175-186.

[19] Roiger, R. and Geatz, M. (2003), *Data Mining: A Tutorial-Based Primer,* Addison Wesley.

[20] Romei, A., and Turini, F. (2010), 'XML Data Mining'. *Software – Practice and Experience*, 40, 101-103.

[21] Tan, P. N. and Kumar, V., Srivastava, J. (2002), 'Selecting the Right Interestingness Measure for Association Patterns'. *The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[22] Wan, J. W. W. and Dobbie, G. (2004), 'Mining Association Rules from XML Data Using XQuery'.*The Australian Workshop on Data Mining and Web Intelligence (DMWI).*

[23] Wu, T., Chen, Y. & Han, J. (2010), 'Re-Examination of Interestingness Measures in Pattern Mining: A Unified Framework'. *Data Mining and Knowledge Discovery*, 21(3), 371-397.

[24] *http://office.microsoft.com/en-us/templates/desktop-northwind-2007-sample-database-TC001228997.aspx*

[25] *http://databases.about.com/od/sampleaccessdatabases/Sample_Microsoft_Access_Databases.htm*